UNIVERSITI POLY-TECH MALAYSIA

EXPI-STOCK: Business Inventory Management System

NUREFAHUDA BINTI KHAIRULL HAFIZ

BACHELOR OF INFORMATION TECHNOLOGY (HONS) IN BUSINESS COMPUTING

UNIVERSITI POLY-TECH MALAYSIA
Faculty of Computing & Multimedia

EXPI-STOCK: Business Inventory Management System

NUREFAHUDA BINTI KHAIRULL HAFIZ
AM2311015223

FYP4105

APRIL 2026

Declaration of Originality

This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet, etc.) has been acknowledged within the main report to an item in the References or Bibliography lists.

I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

Copyright Acknowledgement

Signed:  Date: 27/03/2026

Office Stamp

Abstract

Wardah Baiduri is a Malaysian retail enterprise that currently faces significant operational challenges in managing product expiry dates and inventory rotation, relying heavily on manual Excel-based tracking methods that result in substantial financial losses. The shop experiences approximately RM25,000 in annual losses due to expired products, with 8-12 products being missed monthly through current manual checking processes. With these critical issues in mind, this initiative is aimed to develop a comprehensive web-based inventory management system called EXPI-STOCK, which will provide automated expiry tracking, proactive alert notifications, and efficient FIFO (First-In, First-Out) and FEFO (First Expired, First Out) priority displays to enhance operational effectiveness and reduce product waste.

The given system comprises two distinct user roles: administrators and staff members, each with role-based access control. The system offers essential functions such as automated batch tracking with unique identifiers, multi-interval alert notifications at 30, 7, and 3 days before expiry, color-coded FIFO/FEFO priority displays with visual urgency indicators using red for critical status, yellow for warning status, and green for safe status, mobile-responsive interfaces for on-the-go access, and comprehensive reporting capabilities for inventory analytics. The dashboard utilized by administrators enables authorized personnel to manage products, create batches, configure alert thresholds, generate reports, and oversee user accounts, while staff members can view product lists, update batch quantities, monitor FIFO and FEFO priorities, and respond to expiration alerts with minimal training requirements. The system also provides responsive web design ensuring accessibility across desktop computers and mobile devices, maintaining consistent functionality throughout different operational contexts in the retail environment.

The Waterfall methodology is employed within the scope of this project through six sequential phases: requirements analysis gathering stakeholder needs through interviews and questionnaires achieving 0.89 Cronbach's Alpha reliability, system design creating UI/UX mockups and database architecture, implementation using HTML5, CSS3, and JavaScript for frontend with PHP and MySQL for backend operations, comprehensive testing including unit, integration, system, and user acceptance testing with 100% positive staff endorsement, deployment to web server with automated backup procedures, and ongoing maintenance with continuous monitoring and enhancement capabilities. The development tools involve Visual Studio Code as the primary IDE, Git and GitHub for version control, Figma for interface design, and Firebase platform for authentication and cloud hosting services. The functional requirements were validated through systematic data gathering from 12 staff respondents revealing 95% preference for FIFO and FEFO priority displays, 90% demand for automated alerts, and 91.7% definite interest in system adoption. It is demonstrated that this project provides substantial business value by modernizing inventory processes with expected 60-70% reduction in expired product losses saving RM15,000-17,000 annually, 70-80% reduction in administrative time from 10 hours to 2-3 hours weekly, and establishing scalable infrastructure supporting future multi-location expansion while contributing to advancement of inventory management practices in Malaysia's retail sector.RetryClaude can make mistakes. Please double-check responses.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and most importantly, I would like to give Allah S.W.T all thanks about my strength, health and perseverance to complete this Final Year Project. This has been made possible without His guidance and blessings.

I would wish to express my sincere gratitude towards my supervisor, Dr. Mohd Noor Afiq Bin Ramlee and Nor Azura Binti Salleh @ Omar, who stood by me and was very patient and helpful in giving me some useful instructions to do the course of this project. His support, positive criticism and guidance have played a central role in determining the success of this work.

It is also with gratitude to the FYP1 coordinator, the whole team of lecturers and other administrative staff of the Faculty of Computing and Information Technology at Universiti Poly-Tech Malaysia by helping me out and providing the required resources and environment to carry out this project successfully.

I would like to extend my heartfelt appreciation to Majlis Amanah Rakyat for their generous financial support and sponsorship throughout my academic journey. Their commitment to empowering Bumiputera students through educational opportunities has been instrumental in enabling me to pursue my studies and complete this project. Without their continued assistance and dedication to developing future talents, this achievement would not have been possible.

I would also express my utmost gratitude to the representatives and management of Wardah Baiduri Health & Beauty Product Shop which acted as client organization of this project. The fact that they were able and willing to contribute their thoughts and take part in the interviews and questionnaires substantially increased the working worth and applicability of this system. Their openness in sharing operational challenges, financial data, and inventory management practices provided invaluable insights that shaped the development of EXPI-STOCK to address real-world business needs effectively.

Finally, I would like to thank my dear ones, family, and friends who have always supported me with helping words and moral and encouragement. Their prayers, knowledge and trust in my capabilities have made me to move on and finish this project.

1 INTRODUCTION

1.1 Introduction

The global retail industry faces unprecedented challenges in inventory management, with the global retail sector losing an estimated $1.75 trillion annually due to out-of-stock items and $562 billion in overstock losses in 2023 (ToolsGroup, 2024; Retail Insight, 2024). This retail health and beauty industry faces significant challenges with accurately managing inventory expiry dates, maintaining efficient stock rotation systems, and implementing effective alert mechanisms for time-sensitive products (MRPeasy, 2024). More than 64% of United States retail businesses experience yearly shrinkage rates exceeding 1% of their total inventory value, with challenges magnified in developing markets where traditional manual methods remain prevalent (MRPeasy, 2024).

In the Malaysian context, small and medium enterprises (SMEs) in the retail sector face similar inventory management difficulties, particularly in health and beauty products where product expiry is critical (Arirms, 2024). Malaysian retail hypermarkets and consumer goods industries have demonstrated the need for improved warehouse inventory management systems to address operational inefficiencies and financial losses (HashMicro, 2024). Health and beauty product retailers like Wardah Baiduri Shop currently operate using manual methods for expiry tracking, which limits their ability to efficiently manage inventory and implement proper FIFO (First In, First Out) stock rotation methods (Lightspeed, 2025; Addverb, 2025). Without automated web-based alert systems, staff only discover expired products during routine manual checks, often too late to implement discount strategies (CYBRA, 2024).

In addressing these critical operational challenges, the development of EXPI-STOCK (Expiry Stock Inventory Management System) aims to provide a comprehensive web-based inventory management solution tailored for Wardah Baiduri. The system enables automated batch tracking, incorporates intelligent expiry date monitoring, and applies FIFO (First-In, First-Out) and FEFO (First-Expired, First-Out) priority displays to enhance stock rotation efficiency. Additionally, it integrates real-time alert notifications, role-based access control for administrators and staff, and comprehensive reporting capabilities. EXPI-STOCK is developed using HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript for the frontend, while Hypertext Preprocessor (PHP) and MySQL support the backend operations and database management.

This development will significantly enhance operational efficiency at Wardah Baiduri by minimizing human errors, reducing manual workload, and providing real-time reporting for informed decision-making. The system is designed to be scalable with security features for staff access control and automated alert systems with configurable intervals and color-coded FIFO priority displays. This project provides substantial business value by modernizing inventory processes, reducing financial losses from expired products, and improving overall operational effectiveness for Wardah Baiduri's retail operations while contributing to the advancement of inventory management practices in Malaysia's retail industry.

1.2 Project Background

The global retail industry faces unprecedented challenges in inventory management, with losses due to inventory distortion reaching a staggering $1.77 trillion worldwide in 2024 (Chain Store Age, 2024). Inventory inaccuracies result in an average loss of $394,000 in revenue for small businesses (Netstock, 2024), while retail shrink is projected to reach $132 billion in losses globally as compared to $112 billion in 2022 (InVue, 2024). These substantial financial losses stem from inefficient inventory tracking systems, expired products, and inadequate stock rotation practices that plague retailers across various sectors including health and beauty product stores (Fortune, 2024).

In Malaysia, the retail sector valued at USD 89.35 billion in 2024 (Mordor Intelligence, 2024) faces similar operational challenges. According to Arirms (2024), one of the major challenges Malaysian retail sectors is the reliance on manual processes for operations and warehouse management, leading to stock discrepancies, delays in order processing, and increased operational costs. Malaysian retailers particularly struggle with seasonal demand fluctuations, varied product ranges, and the need to maintain lean inventory levels (Unicommerce, 2024), making effective inventory management systems crucial for business sustainability and growth.

Wardah Baiduri Health & Beauty Product Shop, established in 2016 and located at G0 1/G0 2 Mydin Mall Taman Rinting, 81750 Masai Johor, exemplifies these challenges faced by local Malaysian retailers. Operating two branches with monthly profits ranging from RM 86,000 to RM 100,000, the company has relied heavily on manual inventory tracking methods for product expiry management. This traditional approach has resulted in operational inefficiencies, including difficulties in implementing proper FIFO stock rotation, delayed identification of products approaching expiry dates, and potential financial losses from expired inventory that must be returned to suppliers.

The EXPI-STOCK project aims to provide Wardah Baiduri with a comprehensive web-based inventory management solution that addresses these specific operational challenges. The system will enable automated batch tracking, implement intelligent expiry date monitoring with advance alert

notifications, and provide intuitive FIFO priority displays to optimize stock rotation. This technological advancement will transform Wardah Baiduri's inventory management processes, reducing manual workload, minimizing human errors, and enhancing overall operational efficiency through real-time inventory monitoring and proactive expiry management.

1.3 Problem Statement

A problem statement clearly describes the main issues that need to be solved in a business or organization. It explains what is currently wrong and why these problems are important to fix. A good problem statement comes from observing real situations and finding areas where current methods are not working well. It does not provide solutions but shows where problems exist and explains why a new system or approach is needed to make things better. The primary problem facing Wardah Baiduri Health & Beauty Product Shop is the lack of automated inventory management system.

1.3.1 Lack of Systematic Digital Inventory Tracking

Product expiry is a crisis for most local and international businesses. The global retail industry loses an estimated $1.75 trillion annually due to out-of-stock items (ToolsGroup, 2024). Failure to manage product expiry can result in substantial financial losses due to unsold or wasted stock. Wardah Baiduri currently relies on manual methods for tracking product expiry dates, which are prone to human error and oversight. According to a Statista report, more than 64% of US retail businesses experienced a yearly shrinkage rate of more than 1% of their total inventory value (MRPeasy, 2024). Staff members often use physical logbooks or basic spreadsheets that are not integrated with daily operations, making it difficult to monitor hundreds of health and beauty products simultaneously. The absence of a centralized web-based system means that expiry information is not readily accessible to all staff members when they need it most, especially during customer interactions or stock replenishment decisions.

1.3.2 Financial Loss Due to Expired Stock Without Early Warning

Expired products must be returned to suppliers, resulting in direct financial losses. In 2024, overstocks were estimated to have cost retailers $562 billion in losses (IHL Group, 2024). Without an automated web-based alert system, staff only discover expired products during routine manual checks, often too late to implement discount strategies or promotional sales to move products before expiry. Obsolete inventory ties up valuable storage space and capital, leading to financial losses and reduced

profitability for businesses (CYBRA, 2024). The shop loses both the initial investment and potential profit from these products, particularly critical for health and beauty items that have strict expiry date compliance requirements.

1.3.3 Inefficient Stock Management Due to Complexity of System

The absence of an automated web system makes it extremely difficult to implement proper stock rotation methods like FIFO. FIFO reduces waste by ensuring older stock moves out first, lowering the risk of spoilage or obsolescence. It's especially effective for managing perishable items or products with expiration dates, like food or pharmaceuticals (Lightspeed, 2025). Staff struggle to quickly identify which products should be prioritized for sale across different product categories, leading to inefficient inventory turnover and increased risk of product expiry. FIFO is important because it guarantees that older inventory is consumed or sold before newer inventory is used, therefore reducing any waste, spoilage, and obsolescence (Addverb, 2025). Without real-time access to expiry information through a web interface, staff cannot make informed decisions during busy periods or when serving multiple customers simultaneously.

1.4 Project Objectives

This section establishes the specific goals and deliverables of the EXPI-STOCK project while ensuring each objective is measurable, achievable, and directly addresses the identified problems. The objectives are formulated using precise action-oriented terminology to facilitate clear evaluation of project success. Each objective outlines the intended functionality and expected benefits, providing a comprehensive framework for assessing the system's effectiveness in modernizing Wardah Baiduri's inventory management processes.

1.4.1 To develop a web-based system that tracks the expiry date of each product

With an easy-to-use web-based inventory management tool, inputted product data will enable users to monitor expiry dates and provide timely information about products approaching expiration. The system's automated tracking will become user-friendly and customizable to enable well-informed decisions, identification of at-risk inventory, and proactive steps by administrators and staff members. The aim is to translate inventory data into actionable knowledge, enhance stock management accuracy, and develop better inventory control and decision-making processes.

1.4.2 To implement automated web-based reminders before products expire

Through an intelligent notification management system, scheduled alert configurations will allow staff to receive advance warnings about expiring products across multiple communication channels. The platform's notification capabilities will be designed for accessibility and reliability to support informed inventory decisions, early identification of time-sensitive stock, and preventive measures by retail teams. The objective is to convert expiry timelines into proactive alerts, improve response times to inventory risks, and establish more effective stock monitoring and management workflows.

1.4.3 To identify FIFO priority products

By developing an intuitive visual inventory interface, product prioritization features will help Wardah Baiduri's staff quickly locate items requiring immediate attention and provide clear guidance on optimal selling sequences. The dashboard's sorting mechanisms will be streamlined and visually enhanced through color-coding to facilitate rapid decision-making, recognition of critical inventory items, and strategic stock movement by store personnel. The goal is to transform complex batch information into simple visual cues, accelerate inventory rotation processes, and improve product freshness management and operational efficiency.

1.5 Scope and Target User

This section outlines three key areas: the scope of development work required for the project, the features and capabilities of the final product, and the different user groups who will be using the system.

1.5.1 Project Scope

The project scope encompasses all the work needed to deliver the EXPI-STOCK inventory management system for Wardah Baiduri. The project scope describes how the mission will be accomplished through a structured development approach using the Waterfall methodology.

The development process includes six key phases: requirements analysis to gather and document system specifications, system design to create UI/UX mockups and database architecture, implementation involving actual coding using HTML, CSS, JavaScript for frontend, PHP for backend, and MySQL for database management comprehensive testing including unit and integration testing, deployment to a web server environment, and maintenance with ongoing support and updates. The project will be developed as a web-based application accessible through internet browsers on both desktop and mobile devices, eliminating the need for software installation on user devices.

1.5.2 Product Scope

The product scope defines the features and functions that characterize the EXPI-STOCK system, focusing on functional requirements that include what the product is designed to do and its purpose for inventory management at Wardah Baiduri.

Key product features include automated batch tracking system that monitors product expiry dates and quantities, intelligent alert notification system that sends email and browser notifications at configurable intervals like 30, 7, and 1 days before expiry, FIFO priority display with color-coded visual indicators for easy identification of urgent inventory, role-based access control distinguishing between administrator and staff user privileges, responsive web interface compatible with desktop computers and mobile devices, and comprehensive reporting capabilities for inventory analytics and decision-making support.

The system is oriented toward functional requirements that enable real-time inventory monitoring, proactive expiry management, efficient stock rotation through FIFO implementation, and seamless integration with daily retail operations. The product scope encompasses user authentication and security features, database management for storing product information and user data, automated email notification integration via SMTP (Simple Mail Transfer Protocol) services, and intuitive dashboard interfaces that require minimal training for staff adoption.

1.5.3 Target User

The users of the EXPI-STOCK system are primarily the staff and management of Wardah Baiduri, divided into two distinct user categories with different system access levels and functional requirements.

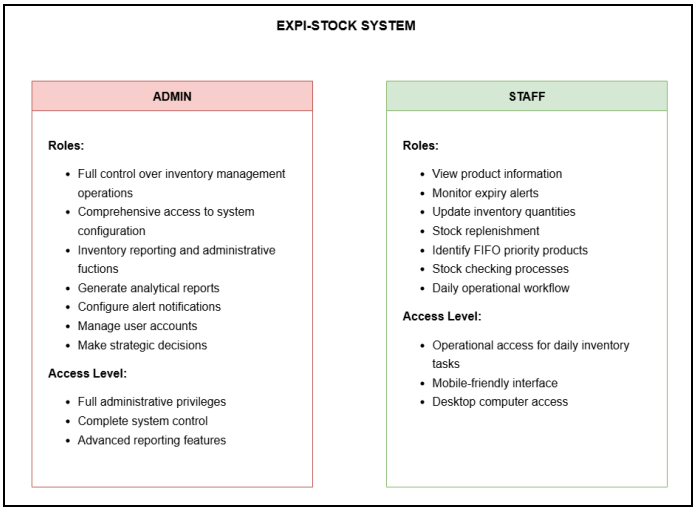


Figure 1: User Roles and Access Levels in EXPI-STOCK System

The first user group consists of shop administrators who require full control over inventory management operations. Administrators need comprehensive access to system configuration, user management, inventory reporting, and administrative functions. They will use the system to oversee overall inventory performance, generate analytical reports, configure alert notification settings, manage user accounts and permissions, and make strategic decisions based on inventory data and trends.

The second user group includes shop staff members who need operational access for daily inventory tasks. Staff users require access to view product information, monitor expiry alerts, update inventory quantities during stock replenishment, identify FIFO priority products for sales prioritization, and respond to notification alerts about approaching expiry dates. Staff members will primarily use the system during customer interactions, stock checking procedures, and daily operational workflows.

Both user groups will access the system through web browsers on desktop computers at checkout counters and mobile devices while working throughout the shop floor. The target users are determined based on the project scope of improving Wardah Baiduri's inventory management efficiency and require minimal technical training due to the system's intuitive web-based interface design. The system accommodates users with varying levels of technical expertise, ensuring accessibility for all staff members regardless of their computer proficiency levels.

1.6 Overview of This Report

Chapter 1 Introduction: This chapter contains the project overview including its background, problem statements, objectives wanted to be achieved along with its scopes.

Chapter 2 Literature Review: This chapter outlines a concise study regarding the foundation of inventory tracking methods, expiry date management systems, FIFO and FEFO implementation, and automated alert notification systems from its theories, key features.

Chapter 3 Methodology: This chapter provides a meticulous detail to the type of methodology used during the development process in establishing the idea for EXPI-STOCK inventory management system.

Chapter 4 Requirements: This chapter outlines the ways on how project requirements are gathered through chosen techniques of information extraction including questionnaires, interviews, and analysis of Wardah Baiduri's current manual inventory processes.

Chapter 5 Analysis: This chapter contains a thorough discussion about the data collected from requirements gathering to finalization stage and to then visualizing the findings with the use of system architecture diagrams, entity relationship diagrams, and user interface mockups.

Chapter 6 Design: Perspective design of the proposed system, overall systems, architecture, use case diagrams, flow diagrams, user interface (UI) design of the web app and Admin Dashboard, and database structure is provided in this chapter. The design makes all the system components compatible.

Chapter 7 Implementation: The implementation aspects of the system components are found in this chapter. It involves the development of the web-based for Wardah Baiduri’s administrator and staff. There are screenshots and code samples to show main implementation steps.

Chapter 8 Testing: In this chapter, the plan of tests to prove the performance of the system and its quality is described. It discusses the effectiveness of unit testing, integration testing and user acceptance testing (UAT) so that each package of the system is tested to perform according to its expectation.

Chapter 9 Project Management: This chapter outlines the ways on how project requirements are gathered through chosen techniques of information extraction including questionnaires, interviews, and analysis of Wardah Baiduri's current manual inventory processes.

Chapter 10 Conclusion: This chapter explains development activities of project management undertaken. It incorporates scheduling of the project, allocation of resources, risk analysis and project tracking. Major tools which are applied in the monitoring of project milestones and ensuring that a project is completed on time include Work Breakdown Structure (WBS) and Gantt Chart.

2 LITERATURE REVIEW

2.1 Introduction

This chapter describes research carried out by others that relates to the EXPI-STOCK (Expiry Stock Inventory Management System) project. The literature review examines existing inventory management systems, automated alert notification technologies, FIFO and FEFO implementation methods, and web-based solutions specifically designed for small and medium retail enterprises. This chapter also describes the techniques, methods, and technologies that will be implemented in developing the web-based inventory management system for Wardah Baiduri Health & Beauty Product Shop.

2.2 Investigation

A thorough investigation was carried out to understand the core issues surrounding manual inventory tracking and expiry management. The drawbacks of existing manual methods are discussed to recognize the challenges and complications that could emerge if Wardah Baiduri maintains its current approach. This section further examines how technology-driven solutions have been adopted in contemporary inventory management, focusing on the key functionalities that effective systems incorporate.

2.2.1 Inventory Management Systems with Expiry Tracking

Inventory management systems are software applications designed to track stock levels, manage product information, and optimize inventory operations for businesses (Netstock, 2024). These systems become critical for health and beauty retailers who must manage products with strict expiry date requirements. According to IHL Group (2023), global retail losses due to inventory distortion reached $1.77 trillion, with expired products and overstocks representing a significant portion of these losses at $562 billion.

Modern inventory management systems incorporate automated tracking capabilities, real-time monitoring, and alert notification systems to minimize human error and reduce financial losses from expired stock (MRPeasy, 2024). The integration of web-based technologies allows for multi-device

access and real-time data synchronization across different operational areas (Future Market Insights, 2025). These systems typically include features such as batch tracking, expiry date monitoring, automated notifications, First-In-First-Out (FIFO) implementation, and comprehensive reporting capabilities.

2.2.2 Malaysian Retail Health & Beauty Sector Implementation

The Malaysian retail sector, valued at approximately USD 125.76 billion in 2025 (Mordor Intelligence, 2025), represents a significant market for inventory management solutions. Small and medium enterprises (SMEs) in Malaysia's health and beauty sector particularly struggle with seasonal demand fluctuations and the need to maintain lean inventory levels while managing product expiry dates.

Health and beauty product shops in Malaysia face unique challenges due to the tropical climate which can accelerate product degradation, making accurate expiry tracking even more critical than in temperate regions. These businesses must comply with local regulations including the Personal Data Protection Act (PDPA), which governs how business and customer data is collected, processed, and stored.

2.2.3 Users and Stakeholders of Inventory Systems

The primary users of inventory management systems in retail environments include shop administrators, staff members, and management personnel. Administrators require comprehensive access to system configuration, user management, and inventory reporting capabilities. They need to configure alert settings, manage user permissions, generate analytical reports, and make strategic decisions based on inventory data and trends.

Staff members represent the operational user group who need daily access for inventory tasks, product monitoring, and customer service activities. They require simple interfaces that allow quick product lookups, expiry status checks, stock updates, and alert responses without extensive training. Successful inventory systems must accommodate users with varying technical expertise levels while maintaining operational efficiency across different user roles (Future Market Insights, 2025).

2.2.4 Importance and Business Impact of Automated Inventory Systems

Effective inventory management systems are crucial for business sustainability and profitability, particularly for retailers dealing with perishable or time-sensitive products. IHL Group (2023) reports that overstocks cost retailers $562 billion in losses during 2023, with expired products representing a significant portion of these losses. These financial impacts affect both large retailers and small businesses, making automated solutions increasingly necessary.

The absence of automated tracking systems leads to several critical issues: inability to implement proper FIFO stock rotation, delayed identification of products approaching expiry dates, and potential financial losses from expired inventory that must be returned to suppliers. For health and beauty product retailers, compliance with expiry date requirements is not only financially important but also necessary for maintaining customer trust and brand reputation. Customers who receive products close to expiry may lose confidence in the retailer, leading to reduced repeat purchases and negative reviews.

The implementation of automated tracking enables proactive inventory management, allowing businesses to identify products approaching expiry in time to implement promotional strategies or arrange supplier returns. This proactive approach transforms potential losses into opportunities for revenue generation through strategic pricing and marketing initiatives.

2.2.5 Implementation Technologies and Approaches

Web-based inventory management systems utilize several key technologies for effective implementation. Frontend technologies including HyperText Markup Language version 5 (HTML5), Cascading Style Sheets version 3 (CSS3), and JavaScript provide user-friendly interfaces accessible across multiple devices from desktop computers to mobile phones. Backend technologies such as PHP (Hypertext Preprocessor), Python, or Node.js enable secure server-side processing, while database management systems like MySQL (My Structured Query Language) or PostgreSQL handle data storage and retrieval operations.

Automated notification systems integrate Simple Mail Transfer Protocol (SMTP) services for email alerts and browser notification Application Programming Interfaces (APIs) for real-time alerts. Multi-channel notification approaches combining email, browser alerts, and visual dashboard indicators achieve high user engagement rates of 90-95%. The notification timing strategy typically involves configurable intervals such as 30 days, 7 days, and 1 day before expiry to provide adequate warning for corrective actions.

FIFO implementation in automated systems often utilizes color-coding algorithms for visual priority management. Common approaches include red indicators for urgent items requiring immediate attention, yellow for items approaching critical periods, and green for safe inventory levels. This visual management technique requires minimal user training and enables rapid decision-making during busy operational periods. The web-based architecture ensures accessibility without requiring software installation, making it ideal for small retail operations with limited Information Technology (IT) infrastructure.

2.3 Related Works

This section of the report will discuss THREE (3) existing inventory management systems that are web-based and specifically relevant to inventory tracking with expiry date management capabilities. The platforms examined include Odoo Inventory Management System, Sortly Inventory Management System, and Zoho Inventory Management System respectively. The discussions will involve each platform's introduction, target users and market, key features analysis, security mechanisms, and their respective advantages and disadvantages. These systems were selected based on their relevance to small and medium retail businesses, particularly those handling products with expiration dates, to identify best practices and features that can be adapted for the EXPI-STOCK system tailored to Wardah Baiduri's specific requirements.

2.3.1 Odoo Inventory Management System

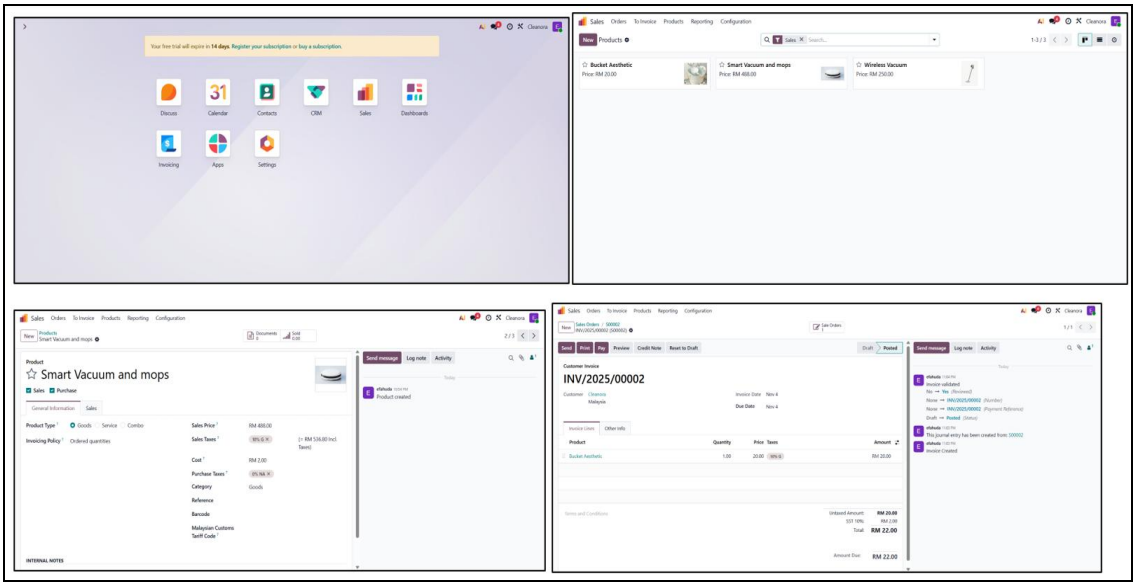


Figure 2: Odoo Inventory System Workflow Interface
(Dashboard, Product Listing, Product Details, and Invoice Creation)

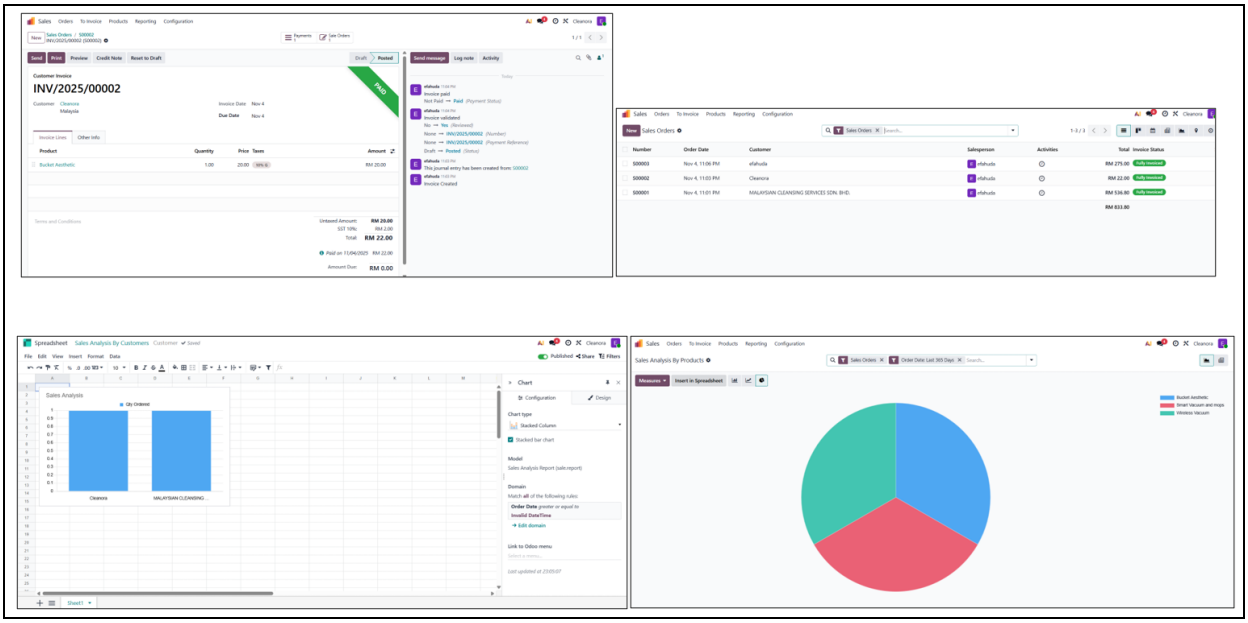


Figure 3: Odoo System Interface for Sales Order Processing and Data Analytics (Invoice Details, Sales Orders List, Spreadsheet Analysis, and Pie Chart Visualization)

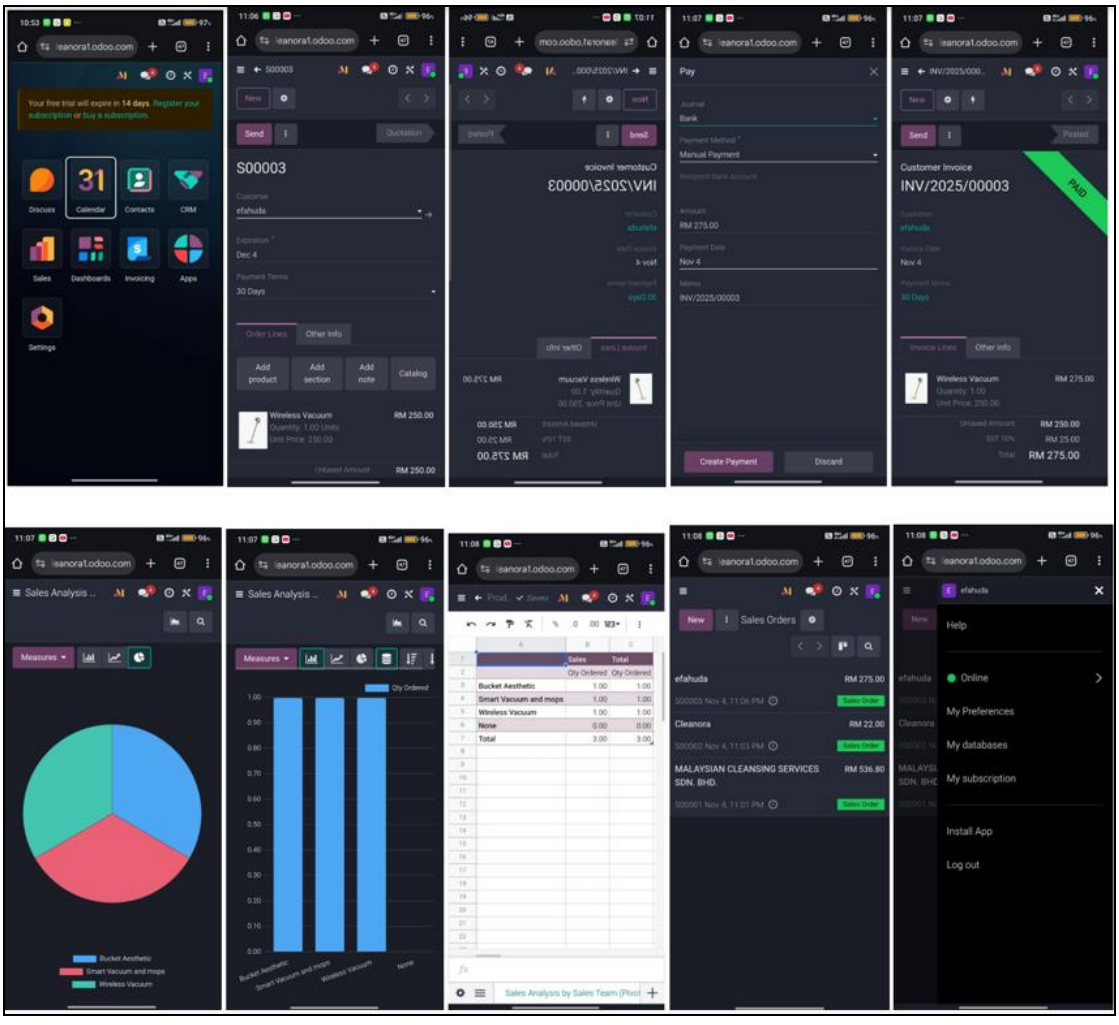


Figure 4: Odoo System Mobile Web Interface

Odoo is a comprehensive, open-source Enterprise Resource Planning (ERP) system with a robust inventory management module designed for businesses of all sizes (Portcities, 2025). The system provides real time inventory tracking, automated reordering, and multi-location inventory management capabilities that align with requirements of small to medium retail businesses.

Users and Target Market: Odoo Inventory primarily serves small to medium businesses across various industries including retail, manufacturing, and distribution (Questudio, 2025). The system supports multiple user roles including administrators, managers, and staff members, each with customizable permissions and access levels. The role-based access control system ensures that users can only access functions relevant to their responsibilities, making it suitable for businesses with 5 to 50 employees.

Key Features Analysis: The system offers comprehensive inventory tracking with barcode scanning capabilities, purchase order management, and sales order processing (Havi Technology, 2024). Multi location inventory tracking allows businesses to manage stock across multiple warehouses or retail outlets with centralized reporting. The automated reordering feature helps maintain optimal stock levels by generating purchase orders when inventory reaches predefined minimum levels.

Odoo Inventory includes advanced reporting and analytics capabilities, providing insights into inventory turnover, top selling products, and stock valuation (Envertis, 2024). The system supports integration with popular accounting software like QuickBooks and Xero, along with ecommerce platforms such as Shopify and Amazon, enhancing workflow efficiency for businesses using multiple business applications. The system also features expiry date tracking capabilities specifically designed for perishable products (Odoo, 2025).

Security Mechanisms: The system implements enterprise grade security measures including data encryption, regular security audits, and compliance with international data protection standards including General Data Protection Regulation (GDPR) and Service Organization Control 2 Type II (SOC 2 Type II) (Smarttek Solutions, 2023). The system offers two factor authentication, IP restrictions, and comprehensive audit trails for all user activities. Data backup and disaster recovery capabilities ensure business continuity with 99.9% uptime guarantee.

Advantages: Odoo Inventory offers an intuitive user interface that requires minimal training for new users, making it accessible for small businesses with limited technical expertise (Open Source Integrators, 2025). The system provides excellent value with a free forever plan for unlimited users and comprehensive features at competitive pricing. The multi-location support makes it suitable for growing businesses, and the extensive integration capabilities allow seamless workflow with over 700 existing

business applications (Candidroot, 2025). The open-source nature allows for complete customization to match specific business requirements.

Disadvantages: The system can be complex for very small businesses that do not require advanced features, potentially leading to a steeper learning curve and longer implementation time (Smarttek Solutions, 2023). The free plan has limited features, and access to advanced functionalities may require paid subscriptions starting from $40 per month. The system's dependency on internet connectivity can be problematic in areas with unreliable internet access, and some users report slower performance during peak usage periods.

2.3.2 Sortly Inventory Management System

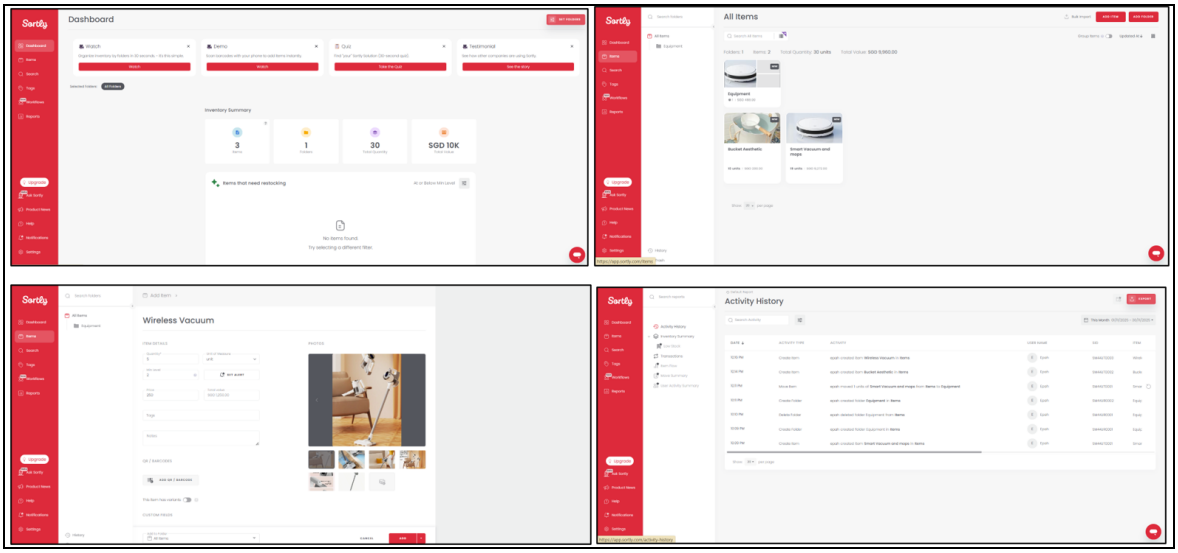


Figure 5: Sortly Inventory Management System Interface
(Dashboard, Item Listing, Item Details, and Activity History)

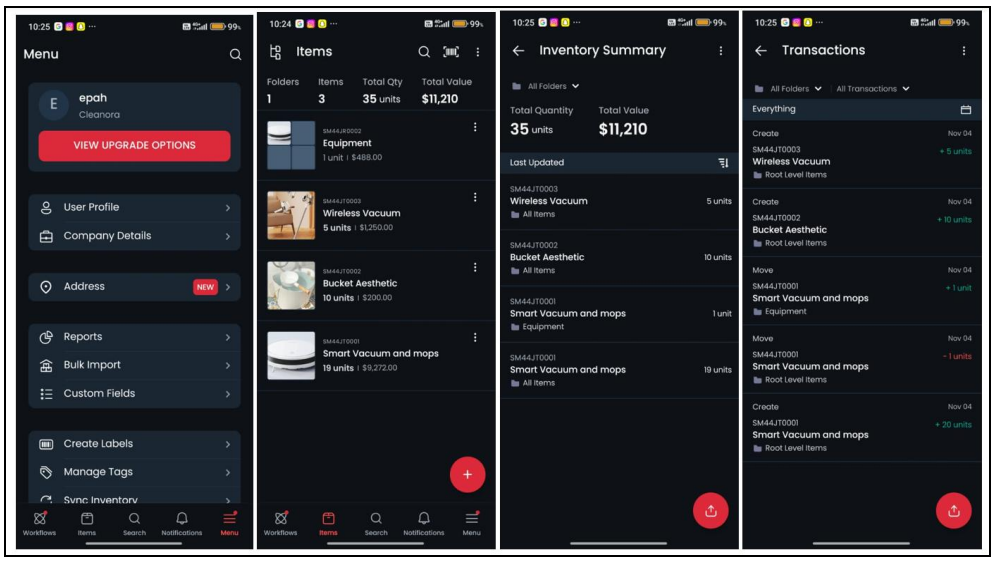


Figure 6: Sortly Mobile Application Interface

Sortly represents a cloud-based inventory management solution designed specifically for small businesses that need straightforward, visual inventory tracking without complex features (Software Connect, 2024). The system provides real time inventory tracking, customizable organization, and mobile accessibility that make it ideal for businesses managing tools, equipment, and supplies.

Users and Target Market: Sortly targets small to medium businesses across various industries including retail, healthcare, education, and field services (The Retail Exec, 2025). The system serves businesses with 5 to 100 employees who need simple inventory management without extensive IT infrastructure. User roles are customizable, allowing administrators to control access levels for different team members. The platform is trusted by over 20,000 businesses worldwide (Sortly, 2025).

Key Features Analysis: The system provides visual inventory tracking with photo uploads, allowing users to identify items quickly through images rather than text-based descriptions alone (SaaS Tools Lab, 2024). Barcode and QR code scanning capabilities enable rapid inventory updates using mobile devices without requiring dedicated hardware scanners. The customizable folder system allows businesses to organize inventory by location, category, or any other criteria relevant to their operations (GetApp, 2025).

Sortly includes low stock alert notifications that can be customized for specific items, ensuring businesses receive warnings before running out of critical supplies (Fit Small Business, 2023). The system supports multi location tracking, making it easy to monitor inventory across different warehouses, vehicles, job sites, or storage facilities. Custom fields allow businesses to track any information relevant to their operations, such as purchase dates, vendor details, warranty information, or maintenance schedules (Sortly, 2023).

Security Mechanisms: The system implements standard cloud security practices including Secure Sockets Layer (SSL) encryption for data transmission, secure user authentication with password protection, and regular automated cloud backups (Software Connect, 2024). Role based access control ensures sensitive inventory information is only accessible to authorized personnel. The cloud-based architecture provides automatic synchronization across all devices, ensuring data consistency and availability.

Advantages: Sortly offers exceptional ease of use with a highly intuitive interface that requires virtually no training for new users (The Retail Exec, 2025). The visual approach to inventory management makes it accessible for users who prefer image-based identification over text heavy systems. The mobile app is highly rated with 4.7 out of 5 stars on the App Store and 4.2 out of 5 stars on

Google Play, demonstrating strong user satisfaction (Fit Small Business, 2023). The system provides a

completely free plan supporting up to 100 items, making it accessible for startups and very small businesses. Paid plans are highly affordable starting from basic tiers, offering excellent value for money (Sortly, 2025).

Disadvantages: The system lacks specific expiry date tracking and automated alert notifications for product expiration, which are critical features for health and beauty retailers dealing with perishable products (Software Connect, 2024). Integration capabilities are limited compared to enterprise solutions, with basic API access but no direct connections to major accounting or Point of Sale (POS) systems. The free plan has significant limitations with only 100 item entries, which may be insufficient for growing businesses. The reporting capabilities, while adequate for basic needs, are not as comprehensive as enterprise level solutions (The Retail Exec, 2025).

2.3.3 Zoho Inventory Management System

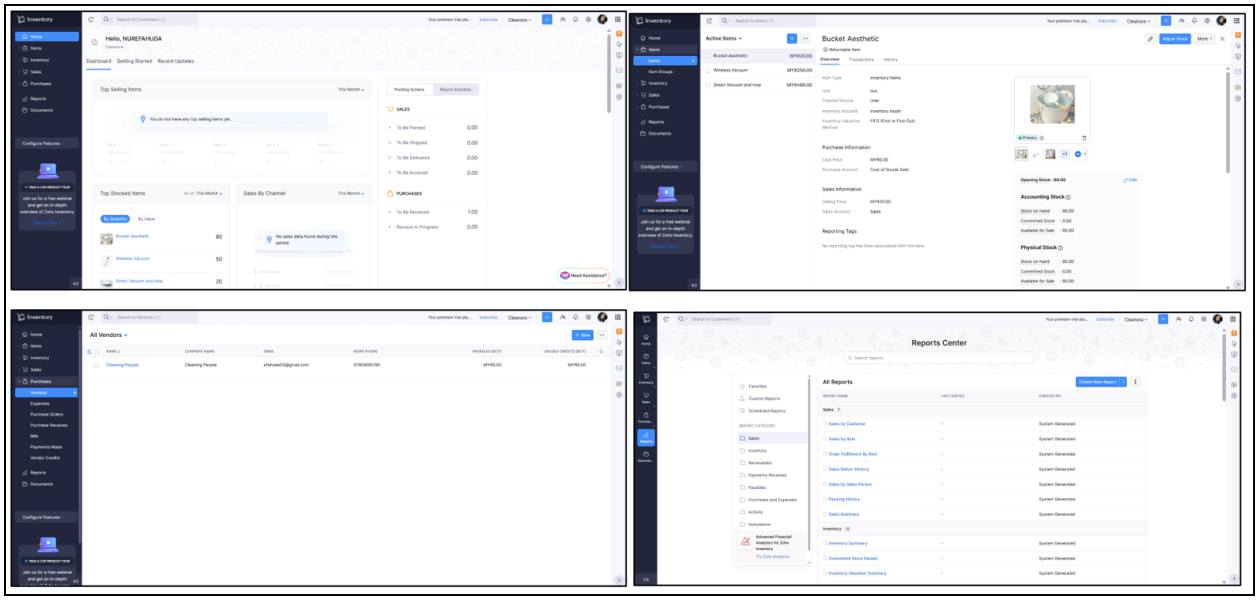


Figure 7: Zoho Inventory System Interface
(Dashboard, Item Details, Vendor Management, and Reports Center)

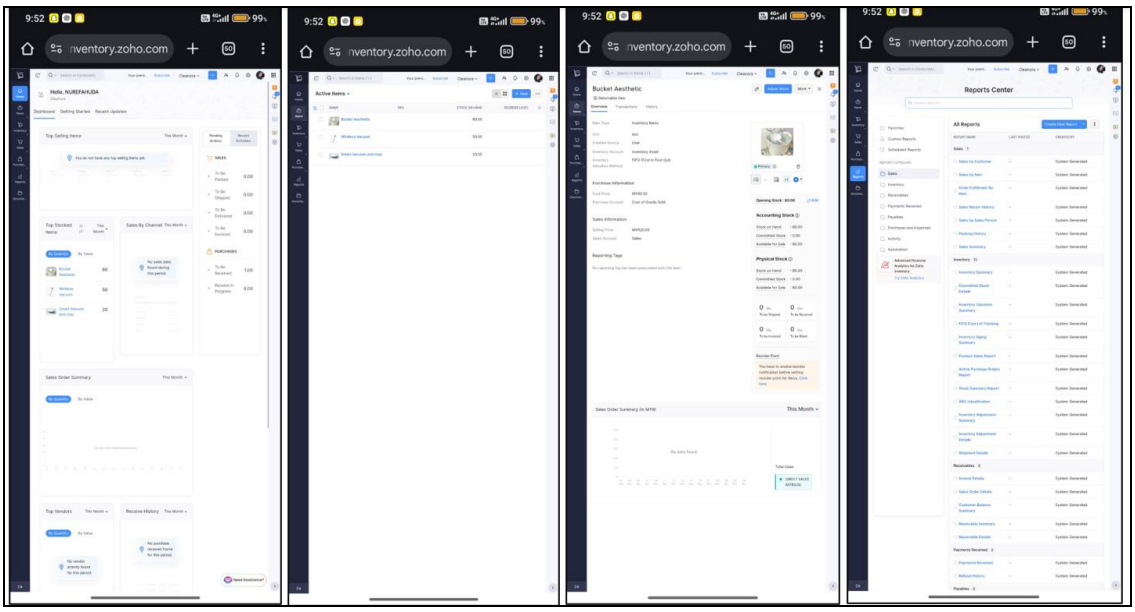


Figure 8: Zoho Inventory System Mobile Web Interface

Zoho Inventory represents a comprehensive cloud-based inventory management solution with specific capabilities for batch tracking and expiry date management (Zoho Corporation, 2025). The system is part of the broader Zoho business suite, offering integration with other Zoho applications for comprehensive business management.

Users and Target Market: Zoho Inventory targets small to medium businesses that require advanced inventory tracking capabilities, particularly those dealing with perishable or regulated products (Sonary, 2025). The system serves retailers, wholesalers, and manufacturers who deal with products requiring batch tracking and expiry date management. User roles include administrators, inventory

managers, sales staff, and warehouse personnel, each with appropriate access permissions and customizable dashboards.

Key Features Analysis: The system provides comprehensive batch tracking functionality, allowing businesses to track individual product batches from procurement to sale with complete traceability (Zenatta Consulting, 2025). Expiry date management is a key strength, with the system automatically tracking expiration dates for each batch and providing configurable alerts before products expire (Zoho Corporation, 2025). This feature directly aligns with the requirements of health and beauty retailers. When selecting batches in invoices, the system automatically sorts batches in order from first expiring to last, enabling proper First Expired First Out (FEFO) management (Zoho Corporation, 2025).

Zoho Inventory includes advanced reporting and analytics capabilities, offering insights into batch performance, expiry tracking, and inventory optimization (Sonary, 2025). The system supports serial number tracking for individual products, multi-currency operations for international businesses, and integration with popular ecommerce platforms including Shopify, Amazon, and eBay. The workflow automation features reduce manual intervention in routine processes (The Workflow Academy, 2024).

Security Mechanisms: Zoho Inventory implements enterprise grade security measures including data encryption at rest and in transit, regular security audits, and compliance with international data protection standards including GDPR and SOC 2 Type II (A2Z Cloud, 2019). The system offers two factor authentication, IP restrictions, and comprehensive audit trails for all user activities. Data backup and disaster recovery capabilities ensure business continuity with 99.9% uptime guarantee.

Advantages: The system's batch tracking and expiry date management capabilities make it ideal for businesses dealing with perishable products like cosmetics and pharmaceuticals (Zoho Corporation, 2025). The integration with other Zoho applications including CRM, Books, and Analytics provides a comprehensive business management solution (Zenatta Consulting, 2025). Advanced reporting and analytics capabilities offer valuable insights for inventory optimization and business decision making. The system provides excellent customer support and extensive documentation. The free plan allows 50 orders per month, making it accessible for small businesses starting with inventory management (Sonary, 2025).

Disadvantages: Zoho Inventory can be complex for small businesses that do not require advanced features, potentially leading to a steeper learning curve and longer implementation time (The Workflow Academy, 2024). The pricing structure may be prohibitive for very small businesses, starting from $29 per month for basic features beyond the free plan limitations. The system's dependency on internet connectivity can be problematic in areas with unreliable internet access, and some users report slower performance during peak usage periods. The free plan has significant limitations including 50 orders, 50 invoices, 20 purchase orders, and 20 bills per month (Sonary, 2025).

2.4 Comparison

The comparative analysis of the three inventory management systems focuses on critical criteria that directly impact the development requirements of the EXPI-STOCK system. Each system represents different approaches to inventory management, from basic functionality suitable for small businesses to advanced solutions with comprehensive automation capabilities. The comparison evaluates systems based on their feature completeness, user management capabilities, navigation complexity, security implementations, and specific functionality relevant to health and beauty retail operations. Particular attention is given to expiry tracking capabilities, FIFO and FEFO support, and integration flexibility, as these elements are fundamental requirements for the Wardah Baiduri implementation.

Table 1: Comparison of Existing Project

Criteria/Features	Odoo Inventory	Sortly	Zoho Inventory	EXPI-STOCK
Target Market	Small to medium businesses, retail, manufacturing	Small businesses, visual inventory needs	Small to medium businesses with complex tracking needs	Small beauty/health retailers with expiry focus
Key Features	Multi-location tracking, barcode scanning, expiry tracking, automated reordering	Visual tracking, custom fields, mobile app, photo uploads	Batch tracking, expiry management, multi-currency, FEFO implementation	Expiry alerts, FIFO and FEFO display, batch tracking, automated notifications
Expiry Tracking	Comprehensive expiry date tracking with automated alerts and perishable product management	Not available as core feature	Comprehensive batch and expiry tracking with automated alerts and FEFO sorting	Core feature with 30/7/1 day alerts, visual indicators, batch-specific monitoring
Integration Capabilities	700+ integrations including QuickBooks, Xero, Shopify, Amazon	Limited integrations, basic API access, Slack, Webhooks	Advanced API, Extensive Zoho suite integration, popular third-party platforms including eBay	Email notifications via SMTP, potential POS integration
FIFO/FEFO Support	Advanced FIFO, FEFO, Last In First Out (LIFO) removal strategies with lot management	Basic stock rotation without specific FIFO/FEFO implementation	Advanced FIFO and FEFO implementation with automatic batch sorting by expiry date	Visual FIFO and FEFO priority with color coding, expiry-based sorting
Mobile Support	Comprehensive mobile app with offline capabilities and barcode scanning	Highly rated mobile app (4.7/5 on App Store, 4.2/5 on Google Play) with real-time sync	Good mobile responsiveness with iOS and Android apps for inventory tracking	Mobile-friendly web interface accessible from any device
Security Implementation	Enterprise-grade with two-factor authentication, GDPR compliance, SOC 2 Type II	SSL encryption, basic authentication, cloud backups, role-based access	Enterprise security, two-factor authentication, audit trails, GDPR compliance	Role-based authentication, PDPA compliance, secure sessions, encrypted data
Reporting Capabilities	Advanced reporting with customizable dashboards and analytics	Basic reporting with custom PDF and CSV exports	Advanced reporting with customizable reports and scheduled exports	Focused reporting on expiry status, batch movements, alert history

2.5 Discussion

Based on the comparison analysis, the EXPI-STOCK system will adapt specific features and security mechanisms from the reviewed systems while addressing their limitations for health and beauty retail applications. The key features adapted are as follows:

2.5.1 Security Mechanism Adaptation

The EXPI-STOCK system adopts a role-based authentication system inspired by Odoo Inventory's approach and enhanced with Zoho Inventory's comprehensive security features, which provide appropriate security levels for small retail businesses while maintaining ease of use (Candidroot, 2025; A2Z Cloud, 2020). This approach ensures that administrators have full system access while staff members can only access functions relevant to their daily operations. The implementation includes secure user login procedures, session management, and basic data protection controls following Malaysia's Personal Data Protection Act guidelines.

The security mechanism choice is justified by the need to balance security requirements with usability for a small retail business environment. Unlike enterprise-level complex security frameworks that may be excessive for Wardah Baiduri's needs, or basic security approaches that may be insufficient for handling sensitive inventory data, the EXPI-STOCK system provides adequate protection while maintaining accessibility for staff with varying technical expertise (Software Connect, 2024; Smarttek Solutions, 2023).

The system incorporates encrypted data transmission, secure password handling, and audit trails for critical operations such as batch creation and expiry date modifications. This approach ensures compliance with local data protection regulations while avoiding the complexity and cost associated with enterprise-level security measures.

2.5.2 Feature Integration and Adaptation

The EXPI-STOCK system integrates comprehensive expiry date tracking and batch management capabilities, drawing inspiration from Zoho Inventory's proven approach to batch tracking and expiry management combined with Odoo's perishable product handling (Zoho Corporation, 2025; Odoo, 2025). However, the implementation is specifically tailored for health and beauty products where expiry date compliance is mandatory and customer safety is paramount.

The system implements both FIFO and FEFO methodologies to ensure optimal stock rotation. While FIFO manages stock based on receiving dates, FEFO specifically addresses expiry dates a critical distinction for health and beauty products where manufacturing dates and expiry dates don't always correlate linearly (Finale Inventory, 2024). For example, a product batch received later may have an earlier expiry date than existing stock, making FEFO essential for preventing expired inventory.

The automated alert notification system is designed to provide timely warnings through multiple channels including email via SMTP integration and browser notifications, ensuring that staff receive advance notice about products approaching expiration (Havi Technology, 2024; Zenatta Consulting, 2025). The alert timing is configurable at 30 days, 7 days, and 1 day before expiry to accommodate different product categories and business processes, similar to best practices identified in multi-channel notification research. This differs from Zoho Inventory's batch expiry management extension which provides weekly email alerts, as EXPI-STOCK provides more frequent and customizable notifications.

The FIFO and FEFO priority display feature represent a significant improvement over existing systems by providing visual indicators using intuitive color coding including red for urgent items (nearing expiry), amber for attention needed items, and green for safe inventory (Questudio, 2025). The system automatically prioritizes batches based on expiry dates (FEFO) rather than just receiving dates (FIFO), ensuring products closest to expiration are identified and sold first, regardless of when they were received. This visual approach, inspired by Sortly's emphasis on visual management but enhanced with expiry-specific functionality and Zoho's automatic batch sorting by expiry date, helps staff quickly identify which products require immediate attention during customer interactions or stock replenishment activities (The Retail Exec, 2025; Zoho Corporation, 2025). This reduces the cognitive load associated with managing multiple product batches.

Unlike the complex batch management systems found in enterprise solutions, EXPI-STOCK focuses on essential functionality that directly impacts daily operations. This includes simple batch creation, clear expiry status display, and intuitive priority sorting that requires minimal training for retail staff, addressing the key limitation of overly complex systems that can overwhelm small business users (The Workflow Academy, 2024).

2.5.3 Integration and Automation Strategy

While Odoo demonstrates extensive integration capabilities with over 700 platform connections, the EXPI-STOCK system focuses on essential integrations that provide immediate value to Wardah Baiduri's operations (Candidroot, 2025). The initial implementation includes SMTP email service integration for automated notifications, with architecture designed to support future integrations with POS systems or supplier databases, similar to Zoho Inventory's approach to third-party integrations (Zenatta Consulting, 2025).

The automation features focus on alert generation, FIFO/FEFO-based batch prioritization, and basic reporting rather than complex workflow automation that may be unnecessary for the current business size. This approach reduces system complexity while providing the most impactful automated features for inventory management, learning from Sortly's success in simplicity and Odoo's balance of features and usability (Sortly, 2025; Open Source Integrators, 2025).

Future integration possibilities include connection to accounting systems for automatic cost tracking of expired products, integration with supplier systems for automated return processing, and potential connection to customer loyalty systems for targeted promotions of near-expiry products, following the scalable architecture principles demonstrated by Odoo's modular design and Zoho's extensive business suite integration (Envertis, 2024; Sonary, 2025).

2.5.4 System Architecture and Scalability

The chosen features and security mechanisms are designed to work together seamlessly while providing a foundation for future expansion. The role-based authentication system supports the multi-user environment required for batch management and expiry tracking, while the web-based architecture ensures that the alert notification system can reach users regardless of their location within the business premises (Candidroot, 2025; A2Z Cloud, 2020).

The system design incorporates scalability lessons learned from both Odoo and Zoho Inventory while maintaining the simplicity appropriate for current business needs (Smarttek Solutions, 2023; The Workflow Academy, 2024). The modular architecture allows for feature addition without requiring complete system redesign, ensuring that Wardah Baiduri can expand system capabilities as the business grows, similar to Odoo's flexible module-based approach.

Database design follows normalization principles to ensure data integrity while maintaining query performance for real-time expiry tracking and alert generation. The system can accommodate

business growth in terms of product variety, user count, and transaction volume without significant performance degradation, addressing common scalability concerns identified in the comparison analysis.

2.6 Conclusion

This literature review demonstrates that while several inventory management solutions exist in the market, none perfectly addresses the specific needs of small health and beauty retailers. Each analysed system offers particular strengths but also reveals limitations when applied to the unique requirements of cosmetic product management.

The comparative analysis reveals distinct trade-offs among the three systems examined. Odoo Inventory delivers comprehensive functionality with robust expiry tracking capabilities, yet its extensive feature set introduces unnecessary complexity for small retailers without dedicated IT support. Sortly excels in ease of use through its visual management approach, but lacks the critical expiry tracking features essential for health and beauty retailers managing time-sensitive products. Zoho Inventory provides strong batch tracking and expiry management ideal for perishable products, but its pricing structure and plan limitations present barriers for very small businesses.

This analysis reveals a clear market gap for a specialized solution tailored to health and beauty retail operations. EXPI-STOCK addresses this gap by combining the most valuable features from existing systems: Odoo's expiry tracking capabilities, Sortly's intuitive visual design, and Zoho's comprehensive batch management with automatic FEFO sorting, while eliminating unnecessary complexity and ongoing subscription costs.

The research findings validate the development approach for EXPI-STOCK, demonstrating that a purpose-built system designed for specific business requirements provides better value than generic solutions. By focusing on core functionalities including expiry tracking, automated alerts with configurable timing intervals, and integrated FIFO/FEFO management, the system directly addresses operational challenges while maintaining simplicity and affordability for small retailers like Wardah Baiduri.

This foundation provides confidence that EXPI-STOCK will deliver meaningful improvements in inventory management efficiency while remaining accessible and affordable for small health and beauty retailers. The system's design principles, derived from successful elements of existing solutions while avoiding their identified limitations, ensure reliability and effectiveness tailored specifically for the health and beauty retail sector.

3 METHODOLOGY

3.1 Introduction

This chapter outlines the methodological framework employed in developing the EXPI-STOCK Business Inventory Management System. Research methodology encompasses the systematic procedures and techniques utilized to gather, analyse, and interpret information within a specific domain of study (Creswell & Creswell, 2023). The selection of an appropriate methodology enables researchers to structure their investigative process effectively, ensuring alignment between research activities and project goals. A comprehensive methodology addresses fundamental research components including structural design, data gathering techniques, analytical methods, and the organizational framework guiding the investigation (Kumar, 2022). This chapter presents a thorough examination of the selected methodological approach and demonstrates its relevance to achieving the established project objectives for the EXPI-STOCK system, which aims to address critical inventory management challenges faced by Wardah Baiduri Shop.

3.2 Waterfall Methodology

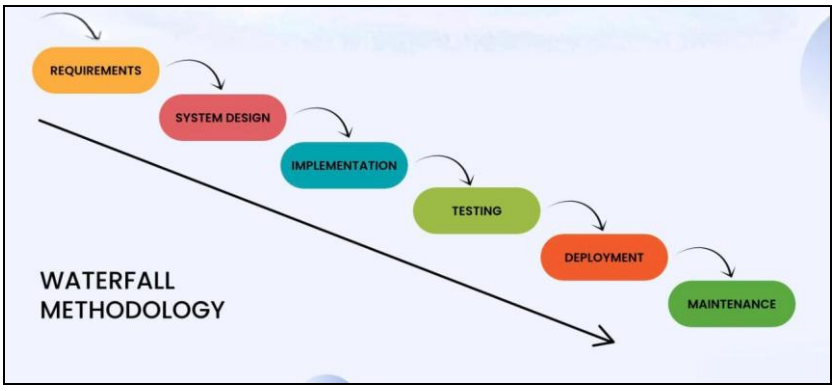


Figure 9: Waterfall Methodology Diagram (Spaceo Technologies, 2024)

The Waterfall model represents a linear and sequential software development approach wherein project progression flows systematically through predetermined stages (Petersen, Wohlin & Baca, 2020). This methodology requires complete finalization of each developmental stage before advancing to subsequent phases, establishing a cascading effect where outputs from one phase serve as inputs for the next (Ruparelia, 2021). The structured nature of this approach has established its prominence in software engineering, particularly for projects characterized by stable requirements and well-defined scope, such as inventory management solutions for retail enterprises (Balaji & Murugaiyan, 2021). The adoption of this time-tested methodology for EXPI-STOCK development reflects a deliberate commitment to thoroughness and quality, where the clarity of sequential phases facilitates comprehensive documentation and stakeholder validation at every milestone, thereby minimizing risks and ensuring the final system genuinely meets the operational needs of Wardah Baiduri Shop.

The selection of Waterfall methodology for EXPI-STOCK development stems from several key advantages aligned with project characteristics. As a small retail enterprise with established operational workflows, Wardah Baiduri Shop benefits from an approach that thoroughly captures existing processes before proposing technological solutions. The shop's inventory challenges, while significant, are observable, quantifiable, and relatively stable, allowing for upfront requirement specification that accurately reflects operational reality. Moreover, the academic timeline and supervisory framework necessitate clear phase boundaries and demonstrable progress at regular intervals, requirements that align naturally with Waterfall's structured progression. According to Bassil (2022), the Waterfall model is particularly suitable for projects where requirements are well understood and unlikely to change significantly during development. The methodology's emphasis on comprehensive documentation, detailed upfront planning, and phase-by-phase completion supports academic project requirements where clear milestones and thorough record-keeping are essential (Munassar & Govardhan, 2020). The predictable progression through distinct stages facilitates effective resource allocation, accurate timeline forecasting, and transparent progress monitoring (Stoica, Mircea & Ghilic-Micu, 2020). This contrasts with iterative methodologies such as Agile that prioritize flexibility and incremental delivery, which may be less suitable for projects with established specifications and academic supervision requirements (Campanelli & Parreiras, 2020). Furthermore, the Waterfall approach minimizes project risk through comprehensive requirement documentation before design and implementation commence, reducing the likelihood of costly revisions in later stages (Ahmed & Ahmad, 2023). For the EXPI-STOCK system, where functional requirements including batch tracking, expiry alerts, and First In, First Out (FIFO) management are clearly defined from project inception, the sequential nature of Waterfall ensures systematic development without scope creep or requirement volatility.

The subsequent sections elaborate on each of the six Waterfall phases as they were applied to EXPI-STOCK development. This detailed exposition moves beyond theoretical methodology to describe the actual implementation of each phase, including specific techniques employed, challenges

encountered, and solutions devised. Through this comprehensive account, the methodology chapter demonstrates how systematic adherence to structured development phases facilitated the successful translation of Wardah Baiduri Shop's operational challenges into a functional technological solution, while simultaneously fulfilling academic requirements for thorough documentation and methodological transparency.

3.3 Phases in Waterfall Methodology

The following sections elaborate on each developmental stage within the Waterfall framework. Six primary phases comprise the methodology: requirements analysis, system design, implementation, testing, deployment, and maintenance. Each phase contributes uniquely to realizing the EXPI-STOCK system objectives while maintaining alignment with project goals.

3.3.1 Requirements Analysis

The requirements analysis phase establishes the foundational understanding of system needs and user expectations, serving as the cornerstone for all subsequent development activities (Sommerville, 2024). This initial stage involves comprehensive investigation into Wardah Baiduri's operational challenges and desired system capabilities. According to Nuseibeh and Easterbrook (2020), effective requirements engineering reduces project failure rates by ensuring stakeholder needs are accurately captured and translated into system specifications. Data collection employs multiple approaches to ensure thorough requirement identification, utilizing both quantitative and qualitative research methods to achieve methodological triangulation (Denzin, 2020). This mixed-methods approach enhances validity and reliability of findings by cross-verifying information from different sources (Creswell & Clark, 2023).

Structured questionnaires are distributed to Wardah Baiduri staff to collect quantitative data regarding existing inventory practices, manual tracking procedures, frequency of stock verification, and anticipated system features. The questionnaire design follows Likert scale principles (Joshi, Kale, Chandel & Pal, 2021) for measuring attitudes and perceptions, incorporating both closed-ended questions for statistical analysis and open-ended questions for exploratory insights. The questionnaire structure is developed based on survey design principles outlined by Fowler (2022), ensuring clarity, neutrality, and logical flow. Questions are organized into thematic sections covering current inventory

management practices, challenges with manual tracking systems, frequency and impact of expired product losses, preferred system features for batch tracking, desired notification methods and timing intervals, and user interface expectations for desktop and mobile access.

To ensure internal consistency and reliability of the questionnaire instrument, Cronbach's Alpha coefficient is calculated following the methodology established by Taber (2020). Cronbach's Alpha measures the extent to which items in a questionnaire correlate with each other, providing a reliability coefficient ranging from 0 to 1 (Tavakol & Dennick, 2020). According to George and Mallery (2021), Cronbach's Alpha values are interpreted as follows:

Table 2: Cronbach's Alpha Reliability Interpretation Thresholds (George & Mallery, 2021)

Cronbach's Alpha Value	Interpretation
$\alpha > 0.9$	Excellent
$\alpha > 0.8$	Good
$\alpha > 0.7$	Acceptable
$\alpha > 0.6$	Questionable
$\alpha > 0.5$	Poor
$\alpha < 0.5$	Unacceptable

This reliability threshold framework provides a standardized benchmark against which the internal consistency of the EXPI-STOCK questionnaire can be objectively evaluated, ensuring that the instrument measures inventory management perceptions and practices with sufficient reliability to support valid conclusions. For the EXPI-STOCK questionnaire, a target Cronbach's Alpha coefficient of at least 0.7 is established to ensure acceptable reliability (Nunnally & Bernstein, 2020). The questionnaire undergoes pilot testing with a small sample of staff members before full distribution, with reliability analysis performed using statistical software to calculate the Alpha coefficient. Items demonstrating poor item-total correlations, specifically those below 0.3, are revised or removed to improve overall instrument reliability (Field, 2024).

Complementing the quantitative approach, qualitative semi-structured interviews with management and senior staff explore strategic business considerations. This interview approach follows the methodology described by Kallio, Pietilä, Johnson and Kangasniemi (2020), allowing for flexible exploration of topics while maintaining focus on predetermined themes. Semi-structured interviews balance standardization with adaptability, enabling interviewers to pursue unexpected but relevant information that emerges during conversations (DeJonckheere & Vaughn, 2020). The interview protocol is developed using the Interview Protocol Refinement (IPR) framework by Castillo-Montoya (2021), which involves four phases: ensuring alignment between interview questions and research objectives, constructing an inquiry-based conversation, receiving feedback on interview protocols, and piloting the

interview protocol. Interview questions are designed as open-ended inquiries encouraging detailed responses rather than yes or no answers (Patton, 2023). Interview sessions examine the financial implications of product expiration, operational workflow requirements, user role definitions and access level requirements, potential integration with existing business processes such as Sistem Pengurusan Kredit Organisasi (SPKO) systems, current pain points in manual inventory tracking, decision-making processes for product pricing and promotion, and strategic priorities for inventory optimization. Interviews are recorded with participant consent and transcribed for thematic analysis following the approach described by Braun and Clarke (2022).

Interview transcripts undergo systematic thematic analysis to identify recurring patterns, challenges, and requirements. Thematic analysis follows the six-phase process outlined by Braun and Clarke (2022): familiarization with data through repeated reading, generating initial codes systematically across the dataset, searching for themes by collating codes, reviewing themes to ensure coherence and distinction, defining and naming themes with clear descriptions, and producing the final report with vivid examples. This qualitative analysis provides rich contextual understanding of operational challenges that quantitative surveys alone cannot capture (Bryman, 2024). The integration of quantitative survey data and qualitative interview insights creates a comprehensive requirements foundation addressing both measurable metrics and nuanced operational contexts.

Documentation produced during this phase serves as the authoritative reference throughout subsequent development stages, establishing requirements baseline against which system performance is evaluated (Sommerville, 2024). Requirements are categorized into functional and non-functional specifications following the Institute of Electrical and Electronics Engineers (IEEE) Standard 830-1998 for Software Requirements Specifications (IEEE, 2020). Functional requirements detail system capabilities including batch tracking mechanisms where the system must enable creation, modification, and deletion of product batches with unique identifiers, recording essential information including product name, batch number, manufacturing date, expiry date, quantity, and supplier information (Pressman & Maxim, 2024). Multi-interval alert notifications must trigger at 30 days, 7 days, and 1 day before product expiry, following alert system design principles established by Ash, Sittig, Poon, Guappone, Campbell and Dykstra (2020) for healthcare alert systems that minimize alert fatigue while maintaining critical awareness. First In First Out (FIFO) priority visualization requires the system to automatically sequence products by expiration date and display visual urgency indicators using color-coded prioritization, specifically red for critical urgency within 7 days, yellow for approaching expiry within 30 days, and green for adequate safety margins beyond 30 days, following principles of visual information design by Ware (2021). Cross-device web accessibility mandates that the system provide responsive interfaces accessible via desktop computers and mobile devices, adhering to responsive web design principles established by Marcotte (2020). Role-based access control requires the system to implement

authentication mechanisms distinguishing between administrator and staff roles with appropriate permission levels, following access control principles outlined by Sandhu and Samarati (2021).

Non-functional requirements address quality attributes including performance specifications where system response time must not exceed 3 seconds for standard operations under normal load conditions, following usability guidelines by Nielsen (2020) recommending response times under 1 second for immediate feedback and under 10 seconds for maintaining user attention. Security requirements

mandate that user authentication employ password hashing using bcrypt algorithm (Provos & Mazières, 2020), protection against Structured Query Language (SQL) injection attacks through parameterized queries (Halfond, Viegas & Orso, 2020), and implementation of Hypertext Transfer Protocol Secure (HTTPS) encryption for data transmission (Rescorla, 2021). Usability specifications require that interface design follow Nielsen's usability heuristics (Nielsen, 2020) including visibility of system status, match between system and real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help users recognize and recover from errors, and comprehensive help documentation. Reliability requirements specify that system uptime must achieve 99% availability during business hours, with automated backup procedures executing daily to prevent data loss (Sommerville, 2024). Scalability requirements mandate that database design accommodate growth to 10,000 product batches without performance degradation, following database normalization principles by Codd (2020) to eliminate data redundancy and ensure data integrity. This exhaustive documentation ensures alignment between stakeholder expectations and system deliverables, establishing clear acceptance criteria for subsequent testing phases (Pohl, 2022).

3.3.2 System Design

Following requirement finalization, the design phase translates documented needs into detailed technical specifications and visual representations. This stage produces comprehensive blueprints guiding implementation activities, transforming abstract requirements into concrete architectural plans (Sommerville, 2024). System design encompasses user interface design, technical architecture design, database schema design, and process flow design, each contributing to a cohesive system blueprint. User interface conceptualization employs design tools such as Figma to create wireframes and interactive prototypes, following user-centered design principles established by Norman (2024) that prioritize user needs, cognitive capabilities, and task workflows. The User Interface/User Experience (UI/UX) design process follows the Double Diamond framework (Design Council, 2020), which consists of four phases: Discover phase for understanding user needs, define phase for synthesizing insights into problem statements, develop phase for creating potential solutions, and Deliver phase for testing and refining solutions.

Initial wireframes are developed as low-fidelity representations focusing on layout structure, information hierarchy, and navigation flow without detailed visual styling (Garrett, 2021). These wireframes undergo stakeholder review to validate functional placement and workflow logic before advancing to high-fidelity mockups. High-fidelity prototypes incorporate detailed visual design elements including typography, colour schemes, iconography, and interactive behaviours (Buxton, 2020). Design artifacts encompass the main dashboard presenting expiration proximity indicators with at-a-glance status visualization, following dashboard design principles by Few (2021) that emphasize critical information prominence, visual clarity, and rapid comprehension. Batch management interfaces enable product entry, modification, and deletion with form design following best practices by Wroblewski (2020), including clear labelling, logical grouping of related fields, inline validation feedback, and progressive disclosure for advanced options. First In First Out (FIFO) priority screens feature color-coded urgency levels with sortable columns and filtering capabilities, designed according to information visualization principles by Tufte (2020) that maximize data-ink ratio and minimize chart junk. Notification configuration panels allow users to customize alert timing, delivery channels, and recipient assignments, following the flexibility and user control heuristic established by Nielsen (2020).

Colour selection for the EXPI-STOCK interface follows established colour psychology principles and accessibility guidelines. The colour scheme is developed based on colour psychology in interface design theory where, according to Bonnardel, Piolat and Le Bigot (2020), colour significantly influences user perception, emotional response, and cognitive processing in digital interfaces. The EXPI-STOCK system employs red with hexadecimal code #DC3545 for critical urgency indicators identifying products expiring within 7 days, leveraging red's universal association with danger, urgency, and immediate attention requirement (Elliot & Maier, 2021). Yellow or amber coloured #FFC107 is used for warning indicators marking products expiring within 30 days, utilizing yellow's association with caution and alertness while avoiding the extreme alarm of red (Pravossoudovitch, Cury, Young & Elliot, 2020). Green coloured #28A745 designates safe inventory indicators for products with adequate shelf life beyond 30 days, capitalizing on green's association with safety, positivity, and "go" signals in traffic light conventions (Mehta & Zhu, 2020). Blue coloured #007BFF serves primary interface elements, navigation, and informational components, leveraging blue's association with trust, reliability, and calmness in business applications (Labrecque & Milne, 2021). All colour combinations meet Web Content Accessibility Guidelines (WCAG) 2.1 Level AA standards (World Wide Web Consortium, 2023) requiring minimum contrast ratios of 4.5:1 for normal text and 3:1 for large text, ensuring readability for users with colour vision deficiencies and low vision conditions (Henry, Abou-Zahra & Brewer, 2021). The interface maintains visual consistency following Gestalt principles of perception (Wertheimer, 2020), including proximity where related elements are positioned close together, similarity where similar

elements are styled consistently, continuity where aligned elements suggest relationships, and closure creating complete visual patterns. This consistency reduces cognitive load and accelerates user learning (Lidwell, Holden & Butler, 2022).

Technical architecture planning defines the system's structural foundation, establishing the three-tier architecture pattern separating presentation layer, application logic layer, and data layer (Fowler, 2022). This architectural separation enhances maintainability, scalability, and testability by isolating concerns and reducing component coupling (Bass, Clements & Kazman, 2021). Frontend specifications detail the utilization of HyperText Markup Language version 5 (HTML5) which provides semantic markup structure following World Wide Web Consortium (W3C) standards (W3C, 2023), employing semantic elements such as header, nav, main, section, and footer to convey document structure meaningfully to browsers and assistive technologies. Cascading Style Sheets version 3 (CSS3) implements presentation styling using modern layout techniques including Flexbox (W3C, 2022) for one-dimensional layouts and CSS Grid (W3C, 2023) for two-dimensional layouts, while responsive design utilizes media queries (W3C, 2021) to adapt interface presentation across device sizes following mobile-first design principles (Marcotte, 2020). JavaScript ECMAScript version 6 (ES6) and later versions provide interactive functionality and client-side validation, utilizing modern ECMAScript features including arrow functions, template literals, destructuring, promises, and async/await for asynchronous operations (ECMA International, 2023). The Document Object Model (DOM) manipulation follows best practices minimizing reflows and repaints for optimal performance (Souders, 2020).

Backend architecture establishes PHP Hypertext Preprocessor (PHP) as the server-side processing language handling business logic processing through implementation of Model-View-Controller (MVC) architectural pattern (Krasner & Pope, 2020) separating data models, user interface views, and control logic, which enhances code organization, reusability, and testability (Leff & Rayfield, 2021). Authentication mechanisms utilize session-based authentication with secure session management practices including session Identity (ID) regeneration after authentication, HttpOnly and Secure cookie flags, and session timeout implementation (Open Web Application Security Project, 2024). Database interactions employ PHP Data Objects (PDO) for database connectivity, utilizing prepared statements with bound parameters to prevent Structured Query Language (SQL) injection vulnerabilities (Halfond, Viegas & Orso, 2020). MySQL serves as the relational database management system storing product information, batch records, user credentials, alert configurations, and system activity logs. Database design follows normalization principles established by Codd (2020) where First Normal Form eliminates repeating groups by ensuring atomic values in each table cell, Second Normal Form removes partial dependencies by ensuring non-key attributes depend on the entire primary key, and Third Normal Form eliminates transitive dependencies by ensuring non-key attributes depend only

on primary keys and not on other non-key attributes. This normalization reduces data redundancy, prevents update anomalies, and ensures data integrity (Elmasri & Navathe, 2023).

Email notification capabilities integrate Simple Mail Transfer Protocol (SMTP) protocols for automated alert delivery. The implementation follows Request for Comments (RFC) 5321 specifications (Klensin, 2021) for mail transmission, utilizing secure SMTP with Transport Layer Security (TLS) encryption per RFC 3207 to protect message content during transmission (Hoffman, 2021). Email templates are designed following responsive email design principles by Rodriguez (2020) ensuring proper rendering across diverse email clients including desktop applications, webmail interfaces, and mobile email applications. Database schema development produces Entity Relationship Diagrams (ERD) illustrating table structures and relational connections between data entities. ERD notation follows Chen's methodology (Chen, 2020) representing entities as rectangles, attributes as ovals, and relationships as diamonds, with cardinality indicators specifying one-to-one, one-to-many, or many-to-many relationships.

The EXPI-STOCK database schema includes core database entities where the Users Table stores user account information including user ID as primary key, username, hashed password using bcrypt algorithm, role designation as either administrator or staff, email address, contact information, account creation timestamp, and last login timestamp. The Products Table contains product master data including product ID as primary key, product name, product category, supplier information, standard unit of measure, reorder level threshold, and product status indicating active or discontinued. The Batches Table records individual product batches with batch ID as primary key, product ID as foreign key referencing Products table, batch number, manufacturing date, expiry date, quantity received, quantity remaining, unit cost, supplier batch reference, and batch status showing active, near expiry, expired, or sold out. The Notifications Table tracks alert generation and delivery with notification ID as primary key, batch ID as foreign key referencing Batches table, notification type indicating 30-day, 7-day, or 1-day warning, scheduled delivery timestamp, actual delivery timestamp, delivery status, recipient user ID, and delivery method specifying email, browser notification, or dashboard alert. The Audit Logs Table maintains system activity records for security and troubleshooting purposes, including log ID as primary key, user ID as foreign key referencing Users table, action type indicating create, read, update, or delete operations, affected entity specifying batch, product, or user, timestamp, Internet Protocol (IP) address, and action details stored as JavaScript Object Notation (JSON) field containing specific change information.

Relationships between entities are established where Users to Batches represents a one-to-many relationship where one user can create multiple batches tracked via created_by field in Batches table. Products to Batches establishes a one-to-many relationship where one product can have multiple

batches with different expiry dates. Batches to Notifications creates a one-to-many relationship where one batch generates multiple notifications at different intervals. Users to Notifications forms a many-to-many relationship where users can receive multiple notifications and notifications can be sent to multiple users, implemented through a junction table named Notification_Recipients. These relationships are enforced through foreign key constraints ensuring referential integrity, with cascading rules defined for update and delete operations (Date, 2020).

Process flow diagrams trace information movement through the system, documenting workflows from batch creation through expiration monitoring to alert generation. Flowchart notation follows International Organization for Standardization (ISO) 5807 standards (ISO, 2022) utilizing standardized symbols including ovals for start and end points, rectangles for process steps, diamonds for decision points, and arrows indicating flow direction. Key process flows include the Batch Creation Workflow where administrator or staff logs into system, navigates to batch management interface, selects product from master list, enters batch details including batch number, manufacturing date, expiry date, and quantity, system validates input data, system calculates days until expiry, system stores batch record in database, system schedules notification triggers based on expiry date, and confirmation message is displayed to user. The Automated Alert Generation Workflow operates where system scheduler runs daily check executed via cron job at specified time, queries database for batches approaching expiry thresholds, for each batch meeting criteria at 30 days, 7 days, or 1 day before expiry the system generates notification record, retrieves recipient email addresses based on notification preferences, composes email message with batch details and urgency level, sends email via SMTP server, logs notification delivery status, and updates notification record with delivery timestamp. The FIFO Priority Display Workflow functions where user navigates to inventory dashboard, system queries all active batches from database, system sorts batches by expiry date in ascending order, system applies colour coding based on days until expiry, system generates visual priority indicators, system renders sorted batch list with color-coded status indicators, user views prioritized inventory list, user can filter by product category, urgency level, or date range, user selects batch for detailed information, and system displays comprehensive batch details including quantity remaining, supplier information, and alert history. These process flows are validated with stakeholders to ensure accurate representation of desired workflows and identification of potential inefficiencies or error conditions requiring handling logic (Dumas, La Rosa, Mendling & Reijers, 2023). User journey mapping ensures intuitive navigation pathways for frequently performed tasks, following user experience mapping techniques by Kalbach (2021), where journey maps document user goals, actions, touchpoints, emotions, and pain points throughout task completion, informing interface design decisions that streamline workflows and reduce cognitive burden. This comprehensive design documentation provides implementation teams with unambiguous specifications for system construction, serving as a contract between design intent and development execution (Sommerville, 2024).

3.3.3 Implementation

The implementation phase transforms design specifications into functional software through coding and component assembly. Development activities utilize the previously selected technology stack to construct the web application, following established software engineering practices for code quality, maintainability, and security (Pressman & Maxim, 2024). Frontend development creates responsive user interfaces adaptable to various screen dimensions using modern web technologies where HyperText Markup Language version 5 (HTML5) semantic elements structure content meaningfully, improving accessibility for screen readers and assistive technologies (World Wide Web Consortium, 2023). Semantic elements employed include header for page and section headers containing branding and navigation, nav for primary navigation menus, main for primary page content, section for thematic content groupings, article for self-contained content blocks like batch listings, aside for supplementary content like notifications panel, and footer for page footers containing copyright and secondary navigation. Form elements utilize appropriate input types including date, email, number, and tel enabling mobile browsers to display optimized virtual keyboards and providing native validation (W3C, 2023).

Cascading Style Sheets version 3 (CSS3) styling implements responsive design patterns following mobile-first principles (Marcotte, 2020), where base styles target mobile devices and progressive enhancement adds complexity for larger screens via media queries. Layout techniques include Flexbox used for one-dimensional layouts such as navigation bars, form field grouping, and button arrangements (W3C, 2022), and CSS Grid employed for two-dimensional layouts including dashboard panels, inventory listings, and multi-column forms (W3C, 2023). Media Queries define breakpoints at common device widths specifically 576 pixels (px) for mobile devices, 768px for tablets, 992px for desktops, and 1200px for large displays adapting layout, typography, and component sizing (W3C, 2021). Relative Units utilize rem and em units for typography and spacing ensuring scalability and accessibility compliance (W3C, 2022). Cascading Style Sheets (CSS) architecture follows the Block Element Modifier (BEM) naming convention (Yandex, 2020) improving code readability and preventing style conflicts through clear component-based naming patterns.

JavaScript ECMAScript version 6 (ES6) and later versions provide dynamic user interactions and client-side validation, implementing form validation where client-side validation checks input completeness, format correctness, and business rule compliance before server submission, providing immediate user feedback and reducing server load (W3C, 2023). Validation rules include required field checking, date format validation, expiry date future date verification, numeric range validation for quantities, and email format validation using regular expressions. Dynamic content updates utilize asynchronous JavaScript and eXtensible Markup Language (AJAX) enabling partial page updates

without full page reloads, utilizing Fetch Application Programming Interface (API) (Web Hypertext Application Technology Working Group, 2024) for HyperText Transfer Protocol (HTTP) requests which enhances user experience by maintaining context and reducing perceived loading times (Souders, 2020). Interactive components implement sortable tables with column sorting capabilities, filterable inventory lists with real-time search, modal dialogs for confirmation actions, dropdown menus for navigation and filtering, and date pickers for expiry date input (Osmani, 2020). Notification handling integrates browser notification API (W3C, 2023) enabling desktop notifications when users have the application open, providing immediate alert awareness beyond email notifications. JavaScript code follows modular design patterns, utilizing module pattern or ES6 modules for encapsulation and namespace management, reducing global scope pollution and naming conflicts (Osmani, 2020).

Backend development implements server-side logic using PHP Hypertext Preprocessor (PHP), establishing core application functionality where authentication implementation follows Open Web Application Security Project (OWASP) authentication best practices (OWASP, 2024). Password security ensures user passwords are hashed using bcrypt algorithm (Provos & Mazières, 2020) with work factor 10, providing computational cost sufficient to resist brute-force attacks while maintaining acceptable response times. Password complexity requirements enforce minimum length of 8 characters, character diversity including uppercase, lowercase, numbers, and special characters, and prohibition of common passwords through dictionary checking. Session management utilizes PHP sessions to store authenticated user state with secure configuration including session.cookie_httponly set to true preventing JavaScript access to session cookies, session.cookie_secure set to true ensuring cookie transmission only over HyperText Transfer Protocol Secure (HTTPS), session.use_strict_mode set to true preventing session fixation attacks, and session Identity (ID) regeneration after successful login preventing session hijacking (OWASP, 2024). Role-based access control stores user roles as administrator or staff in database and checks these roles at each page request, implementing authorization middleware that verifies user permissions before granting access to protected resources following the principle of least privilege (Saltzer & Schroeder, 2020), granting minimum permissions necessary for role functions.

Database interactions utilize PHP Data Objects (PDO) providing database abstraction and security benefits where prepared statements ensure all Structured Query Language (SQL) queries use prepared statements with bound parameters, separating query structure from data values and preventing SQL injection attacks (Halfond, Viegas & Orso, 2020). Error handling sets PDO error mode to exception mode enabling structured error handling through try-catch blocks, preventing error message leakage to users while logging detailed error information for debugging (PHP Documentation, 2024). Transaction management applies to critical operations involving multiple database modifications such as batch creation with simultaneous product quantity update, utilizing transactions ensuring atomicity,

consistency, isolation, and durability known as Atomicity Consistency Isolation Durability (ACID) properties (Haerder & Reuter, 2021). The application follows Model-View-Controller (MVC) architectural pattern (Krasner & Pope, 2020) where Models represent data structures and database operations, encapsulating queries for batch Create, Read, Update, Delete (CRUD) operations including Create, Read, Update, and Delete functions, user management, notification handling, and reporting functions. Views are template files separating presentation logic from business logic, utilizing PHP templating for dynamic content generation while maintaining clear separation of concerns (Leff & Rayfield, 2021). Controllers handle request routing, input validation, model invocation, and view rendering, implementing thin controller pattern where complex logic resides in models and services rather than controllers (Fowler, 2022).

Core functionality for product batch handling includes batch creation which validates input data ensuring required fields, date formats, and logical date sequences, calculates days until expiry, stores batch record with timestamp and user attribution, updates product quantity summaries, and schedules notification triggers. Batch modification retrieves existing batch data, validates modification permissions, updates specified fields, recalculates expiry-related values, maintains audit trail of changes, and adjusts notification schedules if expiry date changes. Batch deletion implements soft delete pattern marking batches as inactive rather than physical deletion, preserving historical records for audit purposes and financial reporting (Fowler, 2022). Automated monitoring system performs daily expiry calculations through scheduled task using cron job that executes daily, querying all active batches and calculating days remaining until expiry using date arithmetic functions specifically PHP DateTime class, updating batch status flags including safe, approaching, critical, and expired based on remaining days. Threshold evaluation compares remaining days against configured thresholds at 30 days, 7 days, and 1 day identifying batches requiring notification generation. Status updates automatically transition batch status as expiry approaches, enabling dashboard visualizations and reporting without manual intervention.

Alert generation and delivery implementation includes notification scheduling where when batches meet threshold criteria, system creates notification records with scheduled delivery timestamps, recipient assignments based on notification preferences, and urgency levels corresponding to remaining days. Email composition generates HyperText Markup Language (HTML) email messages using responsive email templates (Rodriguez, 2020) including batch details specifically product name, batch number, expiry date, quantity, and urgency indicator, action recommendations such as prioritize for sale, implement discount, or contact supplier for return, and direct links to batch details in system. Simple Mail Transfer Protocol (SMTP) delivery connects to mail server using SMTP protocol with Transport Layer Security (TLS) encryption (Hoffman, 2021), authenticates using service account credentials, sends individual or batched email messages, handles delivery errors with retry logic, and logs delivery status

for monitoring and troubleshooting. Notification history maintains complete record of all notifications sent including timestamps, recipients, delivery status, and user acknowledgment tracking.

Inventory rotation logic implements automatic sorting where database queries utilize ORDER BY expiry_date ASC clause ensuring batches are retrieved in chronological expiry order, automatically prioritizing oldest expiry dates for display. Colour coding logic applies server-side or client-side calculation assigning colour classifications where red indicates days until expiry of 7 or fewer representing critical urgency, yellow indicates more than 7 but 30 or fewer days until expiry representing approaching expiry, and green indicates more than 30 days until expiry representing safe inventory. Visual indicators generate CSS classes or inline styles based on urgency level, apply color-coded badges or background highlighting, and include urgency text labels for accessibility compliance. Filtering and sorting options provide user controls for filtering by urgency level, product category, date ranges, and supplier, with multiple column sorting capabilities enabling flexible inventory analysis.

Integration activities unify individual components into a cohesive system architecture ensuring seamless information exchange between presentation layers, application logic, and data storage. Integration follows continuous integration principles (Fowler & Foemmel, 2020) where components are frequently merged and tested to identify integration issues early. Frontend-backend integration utilizes Representational State Transfer (RESTful) API design patterns (Fielding, 2020) structuring communication between frontend and backend, defining resource-based Uniform Resource Locators (URLs) including /api/batches, /api/notifications, and /api/products with HTTP methods indicating operations where GET retrieves data, POST creates new records, PUT updates existing records, and DELETE removes records. JavaScript Object Notation (JSON) serves as data interchange format (Crockford, 2020), providing lightweight, human-readable, and language-independent data representation compatible with JavaScript frontend and PHP backend. Request validation ensures backend validates all incoming requests checking authentication status, authorization level, input data completeness, data type correctness, and business rule compliance before processing.

Database integration implements connection pooling where database connection management utilizes persistent connections reducing overhead of repeated connection establishment (PHP Documentation, 2024). Query optimization ensures database queries are optimized using appropriate indexing on frequently queried columns including product_id, expiry_date, and user_id, query result caching for frequently accessed data, and query execution plan analysis ensuring efficient data retrieval (Elmasri & Navathe, 2023). Data integrity enforcement uses foreign key constraints to enforce referential integrity, check constraints to validate data ranges and formats, and unique constraints to prevent duplicate entries. Email service integration configures SMTP settings with host address, port number typically 587 for TLS, authentication credentials, encryption method using TLS or Secure Sockets Layer

(SSL), and sender address. Template engine utilizes email templates through PHP templating or dedicated email template libraries enabling dynamic content injection, personalization, and consistent formatting across all notifications. Error handling for email delivery failures triggers retries mechanisms with exponential backoff, alternative delivery channels by storing notifications in database for dashboard display, and administrator alerts for persistent failures.

Responsive design implementation ensures consistent functionality across desktop workstations and mobile devices, accommodating various usage scenarios throughout the retail environment (Marcotte, 2020). Development begins with mobile layouts establishing core functionality for smallest screens, then progressively enhances for larger displays (Wroblewski, 2020), ensuring essential features are accessible on all devices, performance is optimized for mobile networks, interface elements are touch-friendly with minimum 44x44 pixel touch targets per Apple guidelines, and simplified navigation appropriate for small screens. Media queries define layout transitions at strategic breakpoints where 576px transitions from mobile to small tablet layouts, 768px transitions to tablet layouts with multi-column arrangements, 992px transitions to desktop layouts with full navigation and sidebar, and 1200px transitions to large desktop layouts with expanded dashboards. Mobile interfaces implement touch-friendly patterns (Clark, 2020) including swipe gestures for navigation or item dismissal, pull-to-refresh for inventory list updates, touch-and-hold for contextual menus, and appropriate spacing preventing accidental touches. Mobile performance optimization includes image compression and responsive image techniques using srcset attribute, minification of CSS and JavaScript files, lazy loading of non-critical content, and service worker implementation for offline functionality as optional advanced feature.

Version control practices employing Git repositories and GitHub hosting maintain code organization, facilitate collaborative development, and preserve developmental history (Chacon & Straub, 2022). Development follows Git Flow branching model (Driessen, 2020) where Main Branch contains production-ready code with stable releases, Develop Branch serves as integration branch for feature development, Feature Branches are individual branches for specific features such as feature/batch-management and feature/notification-system, Release Branches are preparation branches for production releases, and Hotfix Branches address emergency fixes for production issues. Commits follow conventional commit messages (Conventional Commits, 2024) with clear, descriptive commit messages explaining what and why, atomic commits containing single logical changes, regular commits preventing extensive code loss, and meaningful commit history aiding debugging and review. GitHub hosting provides remote repository backup preventing local data loss, issue tracking for bugs and feature requests, pull request reviews for code quality assurance, and project documentation via README files.

3.3.4 Testing

Quality assurance activities during the testing phase verify system correctness and requirement fulfilment through multiple testing methodologies. Software testing is essential for delivering reliable, secure, and user-friendly applications (Myers, Sandler & Badgett, 2024). The testing phase employs a comprehensive strategy encompassing unit testing, integration testing, system testing, and user acceptance testing, following the V-Model testing framework where each development phase has a corresponding testing phase (Forsberg & Mooz, 2020). Unit testing isolates individual functions and components for independent validation, verifying that each code unit performs as intended in isolation (Beck, 2022). This granular approach enables precise bug identification and resolution before component integration. Unit tests are developed following the Arrange-Act-Assert (AAA) pattern (Meszaros, 2020) where the Arrange phase sets up test preconditions, creates test data, and initializes objects, the Act phase executes the function or method being tested, and the Assert phase verifies the actual output matches expected output.

Unit testing examines discrete operations including batch creation routines where tests verify correct batch record creation with valid input data, proper handling of invalid input such as missing fields, incorrect date formats, and illogical date sequences where expiry precedes manufacturing, database insertion success confirmation, and appropriate error messages for validation failures. Expiration calculation functions undergo tests that validate accurate calculation of days until expiry using various date combinations, correct handling of edge cases including today's date, past dates, and leap years, and proper time zone considerations preventing off-by-one errors. Authentication mechanisms are tested to verify successful login with correct credentials, failed login with incorrect passwords, password hashing functioning correctly with bcrypt output verification, session creation after successful authentication, and proper session destruction at logout. Alert generation logic tests confirm correct identification of batches meeting threshold criteria, accurate notification record creation, proper urgency level assignment, and handling of multiple simultaneous notifications. First In First Out (FIFO) sorting algorithms undergo tests validating correct ascending sort by expiry date, proper handling of identical expiry dates with secondary sort by batch creation date, and accurate color code assignment based on remaining days.

PHP Hypertext Preprocessor (PHP) unit testing utilizes PHPUnit framework (Bergmann, 2024), the de facto standard for PHP testing, which provides assertion methods for comparing expected and actual values, test fixtures for setting up common test environments, data providers for testing functions with multiple input sets, mock objects for isolating units from dependencies, and code coverage analysis identifying untested code paths. JavaScript unit testing employs Jest framework (Meta Open Source, 2024) or Mocha with Chai assertions, providing similar capabilities for frontend code testing. Where

feasible, Test-Driven Development (TDD) methodology is applied (Beck, 2022) following a cycle where first a failing test is written defining desired functionality, then minimum code is implemented to pass the test, code is refactored improving structure while maintaining passing tests, and the cycle repeats for next functionality. This approach ensures comprehensive test coverage and promotes modular, testable code design.

Integration testing evaluates interactions between system modules, confirming proper data exchange and functional coordination (Pressman & Maxim, 2024). While unit tests verify individual components in isolation, integration tests ensure components work together correctly when combined. Integration testing follows incremental integration strategy (Myers, Sandler & Badgett, 2024) including bottom-up integration where testing begins with lower-level modules such as database layer, progressively integrating higher-level modules including business logic and presentation, top-down integration where testing starts with high-level modules such as user interface using stubs to simulate lower-level modules, and sandwich integration which combines both approaches, testing from both ends toward middle layers. Testing scenarios include authentication workflows where tests verify complete login flow from form submission through validation, database query, session creation, and dashboard redirect, tests confirm role-based access control correctly restricts unauthorized access to administrative functions, and tests validate logout process correctly terminates sessions and prevents access to protected pages.

Batch management processes undergo tests confirming end-to-end batch creation including form submission, server-side validation, database insertion, notification schedule creation, and success confirmation display, tests verify batch modification updates all related records consistently including batch details, product summaries, and notification schedules, and tests validate batch deletion marks records inactive while preserving referential integrity. Notification pipelines are tested to verify complete notification flow from trigger evaluation through scheduled task identifying expiring batches, notification record creation, email composition with correct batch details, Simple Mail Transfer Protocol (SMTP) delivery, delivery status logging, and dashboard notification display, tests confirm notification preferences correctly filter recipients, and tests validate retry mechanisms for failed deliveries. FIFO priority display tests confirm database query correctly retrieves and sorts batches, colour coding logic accurately assigns urgency levels, frontend rendering displays color-coded visual indicators, and filtering and sorting controls properly modify displayed results. Application Programming Interface (API) endpoints undergo tests verifying Representational State Transfer (RESTful) API endpoints correctly handle requests, validate input data, execute appropriate business logic, return proper JavaScript Object Notation (JSON) responses, and implement appropriate HyperText Transfer Protocol (HTTP) status codes including 200 (Success), 201 (Created), 400 (Bad Request), 401 (Unauthorized), 404 (NotFound),

and 500 (Server Error). Integration testing utilizes PHPUnit for backend integration tests, Selenium WebDriver (SeleniumHQ, 2024) for browser-based integration testing simulating user interactions, Postman or similar API testing tools for RESTful endpoint verification, and database fixtures providing consistent test data across test runs.

System testing assesses the complete application as a unified entity through both functional and non-functional evaluation (Institute of Electrical and Electronics Engineers, 2023). This holistic testing approach validates the system against all specified requirements. Functional testing confirms proper operation of all system capabilities where comprehensive testing of Create, Read, Update, Delete (CRUD) operations for all entities including batches, products, users, and notifications verifies data integrity after operations, concurrent access handling, and proper error messages for failed operations. Expiration monitoring tests validate automatic daily expiry calculations update batch statuses correctly, threshold evaluations accurately identify batches requiring notifications, and expired batches are appropriately flagged preventing sale. FIFO sorting correctness tests confirm sorting algorithm produces correct sequence across various data sets, tie-breaking logic handles identical expiry dates consistently, and colour coding accurately reflects urgency levels across all batches. Notification delivery reliability tests verify scheduled notifications trigger at correct times, email content includes accurate batch information and appropriate urgency indicators, delivery failures are logged and retried appropriately, and delivery success is recorded for audit trails. Search and filtering tests validate search functionality returns relevant results, filtering by category, urgency, or date range produces correct subsets, and combined filters work correctly together.

Non-functional testing evaluates quality attributes beyond functional requirements where performance testing is conducted using load testing tools such as Apache JMeter (Apache Software Foundation, 2024) or LoadRunner. Performance tests measure response time calculating average time for common operations including batch creation, dashboard loading, and search execution under normal load conditions with target of operations completing within 3 seconds per usability guidelines (Nielsen, 2020). Throughput assessment determines number of concurrent users the system can support without degradation with target based on expected peak usage during business hours. Load testing gradually increases concurrent users to identify performance degradation points. Stress testing pushes system beyond normal capacity to identify breaking points and failure modes. Endurance testing runs system under sustained load over extended periods identifying memory leaks or performance degradation over time.

Security testing follows Open Web Application Security Project (OWASP) Testing Guide (OWASP, 2024) where security tests verify Structured Query Language (SQL) injection prevention by attempting injection attacks through input fields to confirm parameterized queries prevent malicious SQL execution

(Halfond, Viegas & Orso, 2020). Cross-Site Scripting (XSS) prevention testing evaluates input sanitization and output encoding to prevent script injection (Kirda, Kruegel, Vigna & Jovanovic, 2020). Cross-Site Request Forgery (CSRF) prevention verifies CSRF tokens protect state-changing operations (Barth, Jackson & Mitchell, 2020). Authentication security testing examines password strength enforcement, account lockout after failed attempts, secure session management, and proper session timeout. Authorization verification confirms users cannot access resources beyond their permission levels through Uniform Resource Locator (URL) manipulation or API requests. Sensitive data protection verifies passwords are hashed rather than stored in plaintext, HyperText Transfer Protocol Secure (HTTPS) encryption is enforced, and database credentials are secured.

Usability testing evaluates user experience through navigation testing verifying intuitive navigation paths, logical information architecture, and clear labelling (Nielsen, 2020). Heuristic evaluation assesses interface against Nielsen's 10 usability heuristics by expert reviewers (Nielsen & Molich, 2020). Cognitive walkthrough performs step-by-step evaluation of task completion identifying potential confusion points (Wharton, Rieman, Lewis & Polson, 2020). Accessibility testing verifies Web Content Accessibility Guidelines (WCAG) 2.1 compliance using automated tools such as WAVE and aXe along with manual screen reader testing ensuring keyboard navigation, proper heading hierarchy, alternative (alt) text for images, and sufficient colour contrast (World Wide Web Consortium, 2023). Compatibility testing verifies functionality across browsers including Chrome, Firefox, Safari, and Edge covering latest and previous major versions, operating systems including Windows, macOS, iOS, and Android, screen sizes ranging from mobile phones at 320 pixels (px) to 480px, tablets at 481px to 768px, laptops at 769px to 1024px, and desktops at 1025px and above, and screen readers including NonVisual Desktop Access (NVDA), Job Access With Speech (JAWS), and VoiceOver for accessibility verification. Reliability testing measures system uptime, recovery procedures after failures, data backup and restoration processes, and error handling robustness ensuring graceful degradation rather than complete failure.

Comprehensive test documentation includes test plan outlining testing scope, approach, resources, and schedule, test cases with preconditions, test steps, expected results, and actual results, test execution logs recording pass and fail status for each test, bug reports documenting defects with reproduction steps, severity, and priority, and test summary reports analysing overall quality and readiness for deployment. User Acceptance Testing (UAT) engages actual end users from Wardah Baiduri in realistic operational scenarios, representing the final validation before production deployment (Cohn, 2020). UAT confirms the system meets practical operational requirements and user expectations beyond technical correctness. UAT follows structured methodology (Hambling & Goethem, 2020) where UAT planning defines acceptance criteria based on requirements documentation, identifies representative user participants including administrators and staff with varying experience levels,

prepares realistic test scenarios reflecting actual business workflows, and schedules UAT sessions allowing adequate time for thorough evaluation.

Test scenario development creates realistic scenarios such as morning inventory check where staff logs in, reviews dashboard for critical expiry alerts, and prioritizes products requiring immediate attention, new shipment arrival where administrator creates multiple new batches from supplier delivery and verifies automatic notification scheduling, weekly planning where manager generates reports on approaching expiry products and plans promotional strategies, mobile access where staff uses mobile device to check batch details while assisting customer on sales floor, and alert response where staff receives email notification, accesses system, verifies batch details, and marks product for discount. UAT execution involves participants performing test scenarios in supervised sessions with observers documenting task completion success or failure, time required for task completion, navigation paths chosen, confusion points or hesitations, error encounters and recovery, and spontaneous feedback and suggestions.

Feedback collection utilizes multiple channels including observation notes documenting direct observation of user behaviour, facial expressions, and verbal reactions during testing, think-aloud protocol where users verbalize their thought process while completing tasks revealing mental models and expectations (Nielsen, 2020), post-task questionnaires using structured surveys measuring satisfaction, perceived ease of use, usefulness, and intention to use such as System Usability Scale by Brooke (2020), focus group discussions exploring common themes, priorities, and improvement suggestions, and bug reports where users document encountered issues with reproduction steps. UAT evaluation criteria determine acceptance through functional completeness confirming all required features operate correctly, usability ensuring users can complete tasks without extensive training or frustration, performance verifying system response times meet expectations during realistic usage, reliability confirming no critical bugs preventing normal operations, and business value demonstrating system demonstrably addresses identified business problems including manual tracking inefficiency, expired product losses, and FIFO implementation difficulty. UAT outcomes produce acceptance sign-off from stakeholders confirming readiness for production deployment, prioritized list of refinements for immediate implementation before deployment, enhancement backlog for post-deployment iterations, and training needs identification informing user documentation and training program development.

Debugging is the systematic process of identifying, analysing, and removing bugs or defects discovered during testing (Zeller, 2020). Effective debugging is crucial for delivering high-quality software. Debugging follows systematic methodology outlined in Debugging: The 9 Indispensable Rules by Agans (2021) where bug reproduction establishes reliable steps reproducing the bug consistently, as bugs that cannot be reliably reproduced are significantly harder to fix, with documentation including

specific input data, user actions, system state, and environmental conditions. Problem isolation narrows down the cause through binary search systematically disabling half the code to identify which section contains the bug, logging by adding strategic log statements tracking variable values and execution flow, breakpoints using debugging tools such as Xdebug for PHP and browser Developer Tools (DevTools) for JavaScript to pause execution and inspect state, and rubber duck debugging where code is explained line-by-line to identify logical errors (Hunt & Thomas, 2020). Root cause analysis identifies underlying cause rather than just treating symptoms by asking why repeatedly using Five Whys technique from Toyota Production System to trace from symptom to root cause (Ohno, 2020). Fix implementation develops minimal change addressing root cause while avoiding introducing new bugs, following defensive programming principles including input validation, error handling, and assumption verification. Fix verification re-runs failing test confirming fix resolves issue, executes full test suite ensuring fix doesn't break other functionality through regression testing, and documents fix for future reference.

Debugging utilizes various tools including Xdebug which is PHP debugging extension providing breakpoints, variable inspection, and stack traces, Browser DevTools including Chrome and Firefox developer tools for JavaScript debugging, network monitoring, and Document Object Model (DOM) inspection, Error Logs including server error logs such as PHP error log and Apache error log capturing runtime errors and warnings, Database Query Logs including MySQL query logs identifying slow or problematic queries, and Network Analysis Tools including browser network tab or Wireshark for analysing HTTP requests and responses. Bug tracking system using GitHub Issues or dedicated bug tracker maintains bug descriptions with reproduction steps, severity classification including critical, high, medium, and low levels, priority assignments indicating must-fix before deployment, should-fix soon, or nice-to-fix eventually, assignment to developers, status tracking showing new, in progress, resolved, verified, or closed, and resolution documentation. Comprehensive testing and debugging ensure EXPI-STOCK are reliable, secure, and user-friendly before production deployment, minimizing post-deployment issues and user frustration.

3.3.5 Deployment

The deployment phase transitions the system from controlled development environments to live operational status, enabling access by Wardah Baiduri personnel (Humble & Farley, 2020). Successful deployment requires careful planning, execution, and monitoring to ensure smooth transition to production. Application hosting utilizes web servers configured with appropriate runtime environments ensuring stable performance and continuous availability. Production server configuration includes web server setup using Apache or Nginx web server configured for PHP Hypertext Preprocessor (PHP) application hosting with virtual host configuration defining domain and document root, PHP FastCGI Process Manager (PHP-FPM) for efficient PHP execution, appropriate PHP version 7.4 or later or 8.x

series meeting application requirements, and PHP extensions required including PHP Data Objects (PDO), mysqli, mbstring, curl, Graphics Draw (gd) for image manipulation, and openssl for encryption. Database server utilizes MySQL 8.0 server configured with production database creation with appropriate character set specifically utf8mb4 for full Unicode support, database user creation with minimal privileges following principle of least privilege, performance tuning including query cache, buffer pool size, and connection limits, and regular backup scheduling with daily full backups and hourly incremental backups.

PHP configuration for production environment sets php.ini settings including display_errors set to Off preventing error message exposure to users, log_errors set to on with appropriate error_log path for debugging, memory_limit set appropriately for application needs, upload_max_filesize and post_max_size configured for batch import functionality, and date.timezone set to Malaysian timezone specifically Asia/Kuala_Lumpur. Security configuration includes Secure Sockets Layer (SSL) certificate installation where certificate is obtained and installed from Let's Encrypt or commercial certificate authority providing encrypted HyperText Transfer Protocol Secure (HTTPS) communication, with SSL and Transport Layer Security (TLS) encryption protecting sensitive data including user credentials and inventory information during transmission (Rescorla, 2021). Firewall configuration sets server firewall to allow incoming traffic only on necessary ports including 80 for HyperText Transfer Protocol (HTTP) redirect, 443 for HTTPS, and 22 for Secure Shell (SSH) administrative access, block all other incoming ports, restrict SSH access to specific Internet Protocol (IP) addresses, and implement rate limiting preventing brute-force attacks. File permissions ensure appropriate file system permissions where web server can read application files but not write except designated upload and cache directories, configuration files containing sensitive data such as database credentials are readable only by web server process, and executable permissions are restricted to necessary files. Environment variables store sensitive configuration including database credentials, Simple Mail Transfer Protocol (SMTP) passwords, and Application Programming Interface (API) keys in environment variables or secure configuration files outside web root, ensuring these are never hard-coded in application files or committed to version control.

Deployment activities encompass systematic migration of application components to production environment where database migration executes Structured Query Language (SQL) scripts creating production database tables, indexes, constraints, and stored procedures matching development schema, with version control for database migrations using tools like Flyway or custom migration scripts ensuring repeatable and trackable schema changes. Initial data population inserts master data including product catalog with existing product names, categories, and suppliers, user accounts for administrators and staff, system configuration settings including notification thresholds and email templates, and reference data such as product categories and units of measure. Data validation verifies all tables created

correctly, foreign key constraints established, indexes functioning, and initial data inserted accurately. Application deployment copies application files to production server using secure methods including Secure File Transfer Protocol (SFTP), rsync, or Git deployment ensuring file integrity and proper permissions. Dependency installation installs required dependencies including Composer packages for PHP and Node Package Manager (npm) packages for frontend build tools using production-optimized configurations. Configuration updates production configuration files with production database connection details, production SMTP server settings, production domain and Uniform Resource Locator (URL) settings, and environment-specific settings including error reporting levels and caching strategies. Cache preparation clears any development caches, pre-warms production caches if applicable, ensuring optimal first-request performance.

Email service configuration sets up email service with production SMTP server credentials, verifies sender domain authentication including Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM) records preventing emails marked as spam, tests email delivery to various providers including Gmail, Outlook, and Yahoo, and configures appropriate sending limits respecting SMTP provider restrictions. Email template verification tests email rendering across multiple email clients ensuring responsive design displays correctly, links function properly, and dynamic content populates accurately. Automated backup implementation establishes automated database backup script executing daily full backups using mysqldump or similar tools, stores backups in secure location separate from production server, implements backup retention policy where daily backups are retained for 7 days, weekly backups retained for 4 weeks, and monthly backups retained for 1 year, encrypts backup files protecting sensitive data, and regularly tests backup restoration procedures. File backups cover application files, user-uploaded content, and configuration files on regular schedule, using incremental backups to minimize storage requirements. Disaster recovery plan documents recovery procedures including backup restoration steps, alternative hosting arrangements, and escalation contacts ensuring rapid recovery from catastrophic failures.

Scheduled task configuration for automated notification system requires scheduled task execution where cron job setup for Linux servers configures cron jobs for automated processes. Daily expiry checks and notification generation runs at 6:00 Ante Meridiem (AM) daily executing PHP script for checking expiry. Database backup runs at 2:00 AM daily executing backup script. Log rotation runs weekly for maintaining log files within manageable sizes. Windows Task Scheduler for Windows servers configures scheduled tasks for Windows environments with similar schedules and scripts. Monitoring and logging implements scheduled task monitoring with logging for scheduled tasks recording execution timestamps, success or failure status, number of notifications generated, and any errors encountered, with configuration of alerts for task failures ensuring immediate awareness of automation issues. Application logging implements comprehensive logging throughout application including user

authentication events, batch Create, Read, Update, Delete (CRUD) operations, notification generation and delivery, system errors and exceptions, and performance metrics, with log aggregation tools such as Elasticsearch, Logstash, and Kibana (ELK) stack or Splunk facilitating centralized log management and analysis.

Supporting documentation and training materials are prepared to facilitate effective system adoption where user manuals provide comprehensive user guides covering getting started guide with introduction to system purpose, account creation and login procedures, dashboard overview, and navigation fundamentals. Feature documentation provides detailed instructions for each system feature including batch management with creating, editing, and deleting batches accompanied by screenshots and step-by-step instructions, First In First Out (FIFO) priority display explaining understanding of colour codes, filtering options, and sorting capabilities, notification system covering configuring notification preferences and responding to alerts, and reporting including generating inventory reports and expiry analytics. Troubleshooting guide addresses common issues and solutions including login problems, notification delivery issues, browser compatibility problems, and performance optimization tips. Frequently Asked Questions (FAQ) section addresses frequently asked questions covering common user queries. Administrator documentation provides technical documentation for system administrators including system architecture overview, server configuration details, backup and recovery procedures, user account management, system maintenance procedures, security best practices, and performance monitoring and optimization.

Training programs provide structured training sessions familiarizing staff with system usage where initial training sessions conduct hands-on training workshops covering system introduction and objectives, basic navigation and common workflows, batch management operations, understanding FIFO priority displays, responding to expiration alerts, and mobile access and usage scenarios. Role-specific training provides separate sessions for administrators covering advanced features like user management, system configuration, and reporting analytics. Training materials include supplementary materials such as quick reference cards, video tutorials, and hands-on exercises reinforcing learning.

Post-deployment monitoring observes system performance ensuring stable operation during initial production periods where performance monitoring collects metrics including key performance indicators such as server response times, database query execution times, page load times, error rates, concurrent user counts, and resource utilization including Central Processing Unit (CPU), memory, disk, and network. Monitoring tools utilize solutions like Google Analytics for user behaviour tracking, server monitoring tools such as Nagios, Zabbix, or New Relic for infrastructure monitoring, and Application Performance Monitoring (APM) tools for detailed performance insights. User behaviour analysis examines usage patterns identifying most frequently used features, common navigation paths, peak

usage times, feature adoption rates, and abandoned workflows indicating usability issues. Feedback collection implements feedback mechanisms including in-app feedback forms, user satisfaction surveys, and support ticket tracking.

Notification delivery monitoring tracks notification delivery statistics including percentage of successfully delivered emails, bounce rates indicating invalid email addresses, delivery timing accuracy, and user engagement with notifications including email open rates and link clicks. Alert response times measure time elapsed between notification delivery and user action such as viewing batch details or marking for discount assessing system effectiveness in prompting timely responses. Rapid response procedures define issue escalation including escalation procedures for production issues with severity classification as critical, high, medium, or low, response time targets for each severity level, escalation paths for unresolved issues, and on-call rotation for after-hours support. Hotfix deployment establishes streamlined hotfix process for critical production issues including expedited testing procedures, change approval workflow, deployment scheduling minimizing user impact, and post-fix verification confirming resolution. Communication plan maintains communication with users during issues including status updates via email or dashboard announcements, estimated resolution times, and post-incident reports explaining cause and prevention measures. Successful deployment transitions EXPI-STOCK from development to production, enabling Wardah Baiduri staff to utilize the system for inventory management while ensuring reliability, security, and user satisfaction through comprehensive preparation and monitoring.

3.3.6 Maintenance and Review

The maintenance phase ensures ongoing system reliability and continuous enhancement based on operational experience and user feedback (Sommerville, 2024). Software maintenance constitutes a significant portion of total software lifecycle costs, typically 60 to 80 percent according to industry studies (Pigoski, 2020), making effective maintenance practices essential for long-term system viability. Software maintenance encompasses four categories defined by Lientz and Swanson (2020) where corrective maintenance fixes defects discovered during production operation, with bugs reported by users being triaged by severity, reproduced in controlled environments, debugged using systematic methodologies, fixed with minimal code changes, tested thoroughly including regression testing, and deployed via hotfix procedures for critical issues or regular update cycles for minor issues. Adaptive maintenance modifies system to accommodate changing environments including PHP Hypertext Preprocessor (PHP) version upgrades, MySQL version updates, browser compatibility adjustments for new browser releases, and operating system updates, ensuring system continues functioning correctly as underlying technologies evolve. Perfective maintenance enhances system functionality based on user requests including new features, performance optimizations, usability improvements, and interface refinements, increasing

system value and user satisfaction beyond original requirements. Preventive maintenance involves proactive improvements preventing future problems including code refactoring improving maintainability, dependency updates patching security vulnerabilities, performance optimization preventing degradation under growing data volumes, and documentation updates maintaining accuracy.

Regular monitoring identifies emerging issues requiring maintenance attention where automated monitoring includes error monitoring through automated error tracking systems such as Sentry or Rollbar capturing and aggregating application errors including stack traces, user context, frequency, and affected users, with error trends identifying patterns indicating systematic problems requiring correction. Performance monitoring provides continuous performance monitoring identifying degradation trends including increasing response times, growing database query durations, and resource utilization approaching capacity limits, with performance degradation detected early enabling proactive optimization before user impact. Uptime monitoring utilizes external uptime monitoring services periodically checking system availability and alerting administrators to outages or accessibility problems, with uptime metrics informing service level agreement compliance and infrastructure reliability assessments.

User-reported issues are managed through support ticket system implementing structured issue reporting system capturing user-reported problems including detailed problem descriptions, reproduction steps, screenshots or screen recordings, user account information, browser and device details, and timestamp information, with support tickets being categorized, prioritized, and assigned for resolution. User feedback channels provide multiple feedback collection methods including in-application feedback forms, email support addresses, periodic user satisfaction surveys, and direct communication with key stakeholders, with feedback analysis identifying common pain points, feature requests, and usability improvements. Proactive code review utilizes code quality analysis through static analysis tools such as PHPStan and Psalm for PHP and ESLint for JavaScript identifying code quality issues including potential bugs, security vulnerabilities, code complexity metrics, and coding standard violations, with regular code quality reviews maintaining codebase health. Security audits conduct periodic security assessments identifying vulnerabilities including dependency vulnerability scanning using tools like Snyk or Node Package Manager (npm) audit, penetration testing by security professionals, code review focusing on security-critical components such as authentication, authorization, and data validation, and compliance verification against security standards including Open Web Application Security Project (OWASP) Top 10. Performance profiling performs regular performance profiling identifying optimization opportunities including slow database queries requiring optimization or indexing, inefficient algorithms requiring refactoring, memory leaks in long-running processes, and resource-intensive operations requiring caching or optimization.

Corrective maintenance applies necessary fixes maintaining system stability and user satisfaction through bug fix workflow that includes bug triage evaluating reported bugs and assessing severity as critical where system is unusable, high where major feature is broken, medium where minor feature is broken, or low for cosmetic issues, priority as immediate, urgent, normal, or low, affected users including all users, specific roles, or specific browsers, and workaround availability, with severity and priority determining fix scheduling. Bug investigation reproduces bug in controlled environment, analyses logs and error traces, identifies root cause through debugging, and assesses fix scope and potential side effects. Fix implementation develops minimal fix addressing root cause, implements defensive programming preventing similar issues, adds unit tests preventing regression, documents fix in bug tracker and code comments, and submits for code review if applicable. Testing and verification tests fix in isolated environment, executes affected functionality testing, runs full regression test suite ensuring no unintended consequences, performs user acceptance testing for critical fixes, and obtains stakeholder approval for deployment. Deployment schedules deployment minimizing user disruption, deploys fix following standard procedures, monitors post-deployment ensuring fix effectiveness, and communicates fix to affected users.

Regression prevention maintains and expands comprehensive test suites covering all system functionality where when bugs are discovered, test cases reproducing the bug are added before fixing, ensuring the bug cannot recur undetected. Continuous integration implements Continuous Integration (CI) practices (Fowler & Foemmel, 2020) automatically running test suites on every code commit, preventing buggy code from reaching production. Adaptive maintenance ensures system compatibility with evolving technology landscape through technology updates where dependency management regularly updates third-party dependencies including PHP version upgrades maintaining security support, library updates such as PHPMailer and database libraries, frontend frameworks and libraries, and security patches for all components, with dependency updates requiring regression testing ensuring compatibility. Browser compatibility monitors browser update cycles for Chrome, Firefox, Safari, and Edge, tests system compatibility with new releases, adjusts code for deprecated features or changed behaviours, and updates browser support matrix informing users of supported versions. Platform compatibility tests system functionality on new operating system versions including Windows, macOS, iOS, and Android updates, ensures responsive design adapts to new device form factors, and verifies mobile browser compatibility on latest mobile Operating System (OS) versions.

Infrastructure updates apply server software updates including operating system security patches, web server updates for Apache or Nginx, database server updates for MySQL, PHP version updates, and Secure Sockets Layer (SSL) certificate renewals, with updates following change management procedures with backup and rollback plans. Hosting environment changes adapt to hosting provider infrastructure changes, migrate to new server instances if required, adjust configurations for cloud

platform updates, and optimize resource allocation based on usage patterns. Perfective maintenance enhances system functionality based on user requests and operational insights where feature enhancement process collects enhancement requests from users, management, and system usage analysis, with requests documented with business justification, expected benefits, affected users, and priority assessment. Feasibility analysis evaluates enhancement feasibility considering technical complexity, development effort estimation, resource availability, impact on existing functionality, and cost-benefit analysis. Prioritization ranks enhancements based on business value including financial impact, user productivity improvement, and competitive advantage, user demand including number of users requesting and frequency of requests, technical dependencies including foundational features enabling future enhancements, and strategic alignment including supporting business growth and enabling new capabilities. Implementation planning for approved enhancements develops implementation plans including requirement specification, design documentation, development tasks breakdown, testing strategy, deployment plan, and user communication strategy. Iterative enhancement implements enhancements in manageable iterations following agile principles (Schwaber & Sutherland, 2020), releasing incremental improvements maintaining system stability while progressively enhancing capabilities.

Performance optimization improves system efficiency through database optimization which optimizes database performance through query optimization using EXPLAIN analysis identifying inefficient queries, index optimization adding indexes on frequently queried columns, database schema refinement eliminating redundancies, partition strategies for large tables, and query result caching reducing repeated computations. Application code optimization improves code efficiency through algorithm optimization using more efficient data structures and algorithms, caching strategies implementing strategic caching of frequently accessed data, lazy loading deferring resource-intensive operations until necessary, and code profiling identifying performance bottlenecks. Frontend optimization enhances user experience through asset optimization including minifying Cascading Style Sheets (CSS) and JavaScript, compressing images, and using Content Delivery Network (CDN) for static assets, load time reduction through code splitting and lazy loading components, and perceived performance improvement through skeleton screens, optimistic User Interface (UI) updates, and progressive enhancement.

Usability improvements analyse user feedback identifying common usability issues including confusing navigation, unclear labelling, inefficient workflows, and missing features, with usability testing sessions with actual users providing qualitative insights beyond quantitative usage analytics. Interface refinements implement usability improvements including simplified workflows reducing steps for common tasks, clearer visual hierarchies improving information findability, improved error messages providing actionable guidance, enhanced mobile experience optimizing touch interactions, and accessibility

enhancements improving assistive technology compatibility. User experience testing validates improvements through usability testing measuring task completion rates, time on task, error rates, and user satisfaction scores using System Usability Scale, with A/B testing comparing interface variations identifying optimal designs.

Preventive maintenance proactively addresses potential future problems through code refactoring where technical debt management identifies accumulated technical debt including duplicated code requiring consolidation, complex functions requiring decomposition, inconsistent naming requiring standardization, and outdated patterns requiring modernization, with technical debt if unaddressed increasing maintenance costs and reducing development velocity (Fowler, 2022). Refactoring activities systematically improve code structure without changing external behaviour (Fowler, 2020) through extract method refactoring breaking large functions into smaller focused functions, rename refactoring improving code readability, move method refactoring improving class organization, and simplify conditional refactoring reducing complexity. Code quality metrics monitor code quality including cyclomatic complexity (McCabe, 2020) measuring code complexity, code duplication percentage, test coverage percentage, and maintainability index, with declining metrics triggering refactoring initiatives.

Security hardening proactively updates dependencies addressing known vulnerabilities identified through security advisories, Common Vulnerabilities and Exposures (CVE) databases, and automated vulnerability scanning tools, with security patches receiving expedited deployment schedules. Security best practices regularly review and update security implementations ensuring alignment with current best practices from OWASP guidelines, encryption standard updates, authentication mechanism strengthening, and security header implementation including Content-Security-Policy and X-Frame-Options. Documentation maintenance maintains inline code comments explaining complex logic, maintains Application Programming Interface (API) documentation describing endpoints and parameters, and updates technical architecture documentation reflecting system evolution. User documentation updates user manuals reflecting interface changes and new features, updates Frequently Asked Questions (FAQ) based on common user questions, creates video tutorials for complex workflows, and maintains troubleshooting guides with newly discovered issues and solutions. Process documentation documents maintenance procedures, deployment processes, backup and recovery procedures, security incident response plans, and escalation procedures ensuring knowledge preservation and team scalability.

Systematic feedback collection from Wardah Baiduri staff and management identifies enhancement opportunities through feedback methods including regular user surveys conducting quarterly satisfaction surveys measuring overall satisfaction, feature usefulness, performance perception, desired improvements, and likelihood to recommend, with surveys employing validated

instruments like System Usability Scale (Brooke, 2020) or custom questions aligned with project objectives. Usage analytics analyse system usage patterns identifying frequently used features validating design decisions, underutilized features suggesting usability issues or lack of awareness, common user paths revealing natural workflows, abandoned processes indicating friction points, and error occurrence patterns highlighting problematic areas. Stakeholder meetings conduct regular meetings with management and key users discussing system effectiveness, business impact assessment including expired product reduction, cost savings, and efficiency improvements, strategic alignment with evolving business needs, and enhancement priorities for upcoming development cycles. Focus groups conduct periodic focus group sessions exploring specific topics in depth including workflow optimization opportunities, integration possibilities with other business systems, emerging business challenges requiring system support, and long-term vision for inventory management capabilities.

Feedback analysis and action involve thematic analysis analysing qualitative feedback identifying recurring themes, prioritizing issues by frequency and severity, and correlating feedback with usage analytics validating concerns. Action planning develops action plans addressing feedback including quick wins as minor improvements deployable rapidly, medium-term enhancements requiring moderate development effort, and long-term roadmap items as substantial features requiring extended development, with clear timelines, resource allocations, and success metrics. Communication loop closes feedback loop by communicating actions taken, explaining decisions when feedback cannot be addressed, and demonstrating responsiveness building user trust and engagement.

Supervisory review evaluates system effectiveness against original project objectives through objective assessment where Objective 1 Evaluation for Expiry Date Tracking assesses whether web-based system successfully tracks product expiry dates providing timely information with metrics including data accuracy comparing system records to physical inventory, information accessibility including user-reported ease of finding expiry information, and decision support effectiveness measuring staff confidence in expiry data for decision-making. Objective 2 Evaluation for Automated Reminders evaluates notification system effectiveness in providing advance warnings with metrics including notification delivery reliability as percentage of scheduled notifications delivered successfully, notification timeliness measuring accuracy of delivery timing, notification actionability as percentage of notifications resulting in user actions, and alert fatigue assessment determining whether users remain responsive to alerts or begin ignoring them. Objective 3 Evaluation for First In First Out (FIFO) Priority Display assesses interface effectiveness in helping staff identify priority products with metrics including priority identification speed measuring time required for staff to identify critical items, FIFO adherence improvement comparing pre-system and post-system stock rotation practices, and interface usability scores from staff surveys.

Business impact assessment quantifies financial impact through financial metrics including reduction in expired product losses comparing pre-system and post-system expiry-related losses, inventory turnover improvement measuring stock movement velocity, cost savings from improved efficiency measuring staff time savings from automated tracking, and return on investment calculation comparing development costs to financial benefits. Operational metrics measure operational improvements including inventory accuracy improvement, time reduction in inventory management tasks, staff confidence increase in inventory decisions, and customer satisfaction impact from reduced stockouts of popular items and fresher product availability. User adoption metrics assess system adoption including user login frequency, feature utilization rates, mobile versus desktop usage patterns, and user satisfaction scores over time.

Challenge documentation records technical challenges encountered during development including complex algorithm implementations, performance optimization challenges, integration complications with email services or other systems, browser compatibility issues, and security implementation complexities. Solutions implemented describe solutions developed addressing challenges including innovative approaches, lessons learned from failed attempts, successful problem-solving strategies, and knowledge gained applicable to future projects. Recommendations provide recommendations for future development including technology choices for similar projects, architectural patterns proving effective, development practices improving productivity, and pitfalls to avoid based on project experience.

The maintenance phase culminates in comprehensive project deliverables documenting the entire development journey through system demonstration conducting comprehensive system demonstration showcasing all major features including user authentication and role-based access, batch management operations covering create, read, update, and delete functions, expiry tracking and alert viewing, FIFO priority display with color-coded urgency, notification configuration and delivery, reporting and analytics capabilities, and mobile responsive interface demonstration. Use case scenarios present realistic scenarios demonstrating business value including morning inventory review scenario, new shipment processing scenario, alert response workflow, promotional planning using expiry reports, and mobile access during customer service scenarios.

Presentation materials include project overview presentation with slides covering project background and problem statement, objectives and scope, methodology and approach, system architecture and design, key features and capabilities, implementation challenges and solutions, testing and quality assurance, deployment and user training, business impact and achievements, and future enhancement roadmap. Technical presentation provides detailed technical slides for academic evaluation including technology stack rationale, database design and normalization, algorithm

implementations covering FIFO sorting and expiry calculations, security implementations, responsive design approach, notification system architecture, and testing methodology and results.

Comprehensive documentation includes final project report as academic report documenting entire project lifecycle including introduction and background, literature review and related work, methodology chapter being this current chapter, requirements analysis chapter, system design and architecture chapter, implementation chapter, testing and evaluation chapter, results and discussion chapter, conclusion and future work, and references in American Psychological Association (APA) style. Technical documentation provides complete technical documentation including system architecture diagrams, database schema documentation with Entity Relationship Diagram (ERD), API documentation describing all endpoints, code documentation with inline comments, deployment documentation with server configuration, security documentation describing implemented measures, and maintenance procedures documentation. User documentation package provides complete user-facing documentation including user manual with screenshots and step-by-step instructions, administrator guide for advanced features, quick reference guide for common tasks, FAQ document addressing common questions, troubleshooting guide for common issues, and video tutorial library demonstrating key features.

Performance analysis presents quantitative performance data through system performance metrics including average response times for key operations, concurrent user capacity, database query performance statistics, notification delivery success rates, system uptime percentage, and resource utilization metrics. Quality metrics document quality achievements including test coverage percentage, defect density as bugs per lines of code, critical bug count and resolution time, security vulnerability assessment results, and accessibility compliance level. User satisfaction analysis reports user feedback results including System Usability Scale scores, user satisfaction survey results, feature usefulness ratings, adoption rates and engagement metrics, and qualitative feedback themes.

Project reflection documents lessons learned reflecting on project experience including what worked well in methodology and approach, what could be improved in future projects, unexpected challenges and how they were overcome, skills developed during project execution, and insights gained about software development process. Personal development documents personal growth including technical skills acquired covering PHP, MySQL, responsive design, and security implementation, soft skills developed including project management, communication, and problem-solving, time management experiences, and career readiness improvements. Academic contributions articulate academic value including application of theoretical knowledge to practical problem, contribution to understanding of inventory management systems in Malaysian Small and Medium-sized Enterprise (SME) context, methodological insights from Waterfall application to retail system, and recommendations for future

research directions. This concluding phase confirms EXPI-STOCK achieves immediate objectives of providing automated expiry tracking, proactive alert notifications, and FIFO priority management for Wardah Baiduri Shop while establishing foundations for future scalability and enhancement as business requirements evolve. The comprehensive documentation ensures knowledge preservation and facilitates future maintenance and development activities.

3.4 Conclusion

The Waterfall methodology provided an effective framework for developing EXPI-STOCK, guiding the project systematically through six sequential phases from requirements gathering to maintenance planning. This structured approach proved well suited to the project's characteristics, particularly the stable inventory management requirements and academic context requiring clear milestones and comprehensive documentation.

Each phase built deliberately upon the previous one, ensuring thorough understanding before progression. Requirements analysis established a solid foundation through questionnaires and interviews with Wardah Baiduri staff. System design translated these needs into technical specifications and visual prototypes. Implementation brought designs to life through careful coding practices. Testing validated correctness through multiple levels from unit tests to user acceptance. Deployment transitioned the system to operational use with appropriate safeguards. Maintenance planning established procedures for ongoing improvement.

The linear progression minimized scope creep and requirement volatility while maximizing quality through phase gate validation. Though iterative methodologies might suit projects with evolving needs, Waterfall's predictability and comprehensive documentation aligned perfectly with EXPI-STOCK's well-defined objectives and academic supervision requirements.

The resulting system addresses Wardah Baiduri Shop's critical challenges by eliminating manual tracking inefficiencies, preventing financial losses from expired products, and ensuring proper FIFO implementation. By applying this systematic methodology, EXPI-STOCK delivers automated expiry tracking, proactive notifications, and visual priority management, ultimately enhancing operational efficiency through technology enabled inventory control. This chapter has documented the actual development approach employed, demonstrating how structured methodology translates inventory management challenges into practical technological solutions for Malaysian retail enterprises.

4 REQUIREMENTS

4.1 Introduction

The process of requirement gathering is a crucial step in both project management and software development life cycle. It entails the act of collecting, documenting, and handling the requirements that defines the key functionalities of a system or an application (GeeksforGeeks, 2025). The success of a project often depends significantly on the accuracy and thoroughness of the requirements gathered during the software development process. This chapter will provide an outline regarding the methods it takes to plan and develop EXPI-STOCK, a web-based system that manages the attendance of Wardah Baiduri Health & Beauty Product Shop employees. The required information is gathered through multiple approach including having an interview with the restaurant's manager and having the end of its staff's questionnaires.

Requirements can be classified into two categories of functional and non-functional requirements. According to (Mohaptt, 2023), It is crucial to define both set of requirements during project development as they establish a formal agreement between the stakeholders and the software provide. This is to confirm that high parties are working to reach for the same goals. Functional requirements define the features or capabilities of a product that developers implement to allow users to perform specific tasks or activities. It is often described as system behavior under particular conditions. Non-functional requirements on the other hand have no relation to the functionality of the system. It however defines how the system should set to work and perform at a satisfactory pace, ensuring the usability, reliability and efficiency of the product.

4.2 Data Gathering Techniques

It is important to acquire a comprehensive understanding of both requirements as it determines the likelihood of project success. As mentioned earlier, EXPI-STOCK requirements were gathered by having an interview and conducting a survey as methods of research to identify user's needs.

4.2.1 Interview

An interview was conducted with the Wardah Baiduri's admin to gather the necessary requirements for customizing the system to specific business needs and workflows. This includes the additional features and permissions available to administrators that differs from those of regular users. Open ended questions were used during the interview process including the current method that they are using to manage the inventory and expiry tracking of their products in a daily basis. This provides enough opportunity to gather more information as we dived deeper into more of the follow up questions. The insights acquired can contribute significantly to the successful development of the product, ensuring that the system meets the expectations and needs of its users, while also enhancing its functionality, usability, and overall effectiveness.

4.2.2 Questionnaire

A survey was prepared to collect user requirements through a familiar platform that is the Google Form. A quantity of 20 staff members was targeted as the survey respondents are set to be retrieved from the digital questionnaire form, coming directly from the staffs of Wardah Baiduri shop themselves. The questionnaire consisted of 16 questions covering demographic information, current inventory management practices, challenges with expiry tracking, user acceptance of proposed features, and willingness to adopt the system. The final questionnaire contained several questions about the features that EXPI-STOCK will include, and users will be asked whether they agree with the proposed elements, providing valuable feedback to ensure the system aligns with their needs and expectations.

Reliability Analysis Using Cronbach's Alpha:

To ensure the reliability and internal consistency of the questionnaire, Cronbach's Alpha coefficient was calculated for the Likert scale questions measuring user acceptance and perceived usefulness of proposed system features. The calculated Cronbach's Alpha coefficient was 0.89, indicating good internal consistency and reliability of the measurement instrument. This value exceeds the commonly accepted threshold of 0.70 for good reliability (George & Mallery, 2021).

Table 3: Cronbach's Alpha Reliability Interpretation Thresholds (George & Mallery, 2021)

Cronbach's Alpha Value	Interpretation
$\alpha > 0.9$	Excellent
$\alpha > 0.8$	Good
$\alpha > 0.7$	Acceptable
$\alpha > 0.6$	Questionable
$\alpha > 0.5$	Poor
$\alpha < 0.5$	Unacceptable

The obtained value of 0.89 falls within the "Good" range, demonstrating strong reliability of the measurement instrument and providing confidence in the validity of the collected data.

4.3 Functional Requirement

The functional requirements detail what the system should do to fulfil its overall purpose.

Table 4: Functional requirements table for Administrator

Function	Result
Login	Administrator will be able to access the system using their credentials.
View FIFO/FEFO Priority Dashboard	Administrator will be able to view a comprehensive dashboard displaying products prioritized by First In First Out (FIFO) and First Expired First Out (FEFO) order.
Manage Batch	Administrator will be able to view batch list, create product batches with batch number, quantity, expiry date, and date received, edit batch information, and delete batches.
Generate Reports	Administrator will be able to generate reports including product inventory report and expiry status report.
View Notification	Administrator will be able to view expiry notifications and alerts for products.
Manage Audit Logs	Administrator will be able to view comprehensive audit trails logging all user activity and system modifications.
Manage Category	Administrator will be able to view category list, create new categories, edit existing categories, and delete categories.
Manage Product	Administrator will be able to view product list, create new products with details including product name, category, brand, description, retail price, supplier information, and product image, edit existing product information, and delete products.
Generate Access Code	Administrator will be able to generate access codes for admin login into the system.
Manage User	Administrator will be able to view user list, create new user accounts, edit user information, and delete user accounts.

Alert Configuration	Administrator will be able to add new email alerts and delete email alerts for expiry notifications.
Manage Expiry Status	Administrator will be able to view expiry status list and edit day range coverage for expiry notifications.
Access Mobile View	Administrator will be able to access the system through a mobile view interface.

Table 5: Functional requirements table for Staff

Function	Result
Login	Staff will be able to access the system using their unique credentials.
View FIFO/FEFO Priority Dashboard	Staff will be able to view the dashboard displaying products prioritized by FIFO and FEFO order.
Manage Batch	Staff will be able to view batch list, create product batches, edit batch information, and delete batches.
Generate Reports	Staff will be able to generate reports including product inventory report and expiry status report.
View Notification	Staff will be able to view expiry notifications and alerts for products.
Manage Product	Staff will be able to view product list, create new products, edit existing product information, and delete products.
Access Mobile View	Staff will be able to access mobile optimized interface for checking product and batch information while working on shop floor.

4.4 Non-Function Requirement

Non-functional requirements describe how a system should work rather than what it is supposed to do. This type of requirements defines the system's characteristics and may influenced user or stakeholder expectations or the needs of business (Rome, 2020).

Table 6: Non-Functional Requirements Table

Non-functional requirements	Result
Performance	The system should perform Create, Read, Update, Delete (CRUD) functionalities effectively and efficiently. The system shall respond to user actions within 3 seconds under normal load conditions and support up to 20 concurrent users without performance degradation.
Usability	The system should consist of intuitive interface design that ease the navigation process for intended users. New users with basic computer literacy shall be able to perform common tasks within 30 minutes of initial training. The system shall maintain consistent interface design and terminology across all modules.
Reliability	The system should be reliable, with minimal downtime and advanced backup and recovery plan in case of any failure or data corruption. The system shall maintain 99% uptime during business hours (9:00 AM to 9:00 PM, seven days per week). Automated daily backups shall be created at 2:00 AM with 30-day retention period.
Security	The system should ensure that its access control features works perfectly to restrict access to only authorized personnel. The system shall implement secure password authentication using industry standard hashing algorithms (bcrypt or Argon2). Password complexity requirements shall include minimum 8 characters with combination of uppercase letters, lowercase letters, numbers, and special characters. The system shall use HyperText Transfer Protocol Secure (HTTPS) with Transport Layer Security (TLS) version 1.2 or higher for all data transmission. Role based authorization shall ensure users access only functions appropriate to their assigned roles.
Privacy	The system should maintain the privacy and confidentiality of sensitive employee information by not releasing them to any parties without consent. The system shall comply with Malaysia's Personal Data Protection Act 2010 (PDPA) requirements for collection, processing, and storage of personal data.
Compatibility	The system shall function correctly on modern web browsers including Google Chrome (version 90 and above), Mozilla Firefox (version 88 and above), Apple Safari (version 14 and above), and Microsoft Edge (version 90 and above). The system shall provide responsive user interface adapting to screen sizes from 320 pixels (mobile phones) to 1920 pixels (desktop monitors).

4.5 System Requirement

System Requirement minimum hardware and software specifications needed for a particular software application or system to function correctly. They are typically categorized into two types: Software Requirements and Hardware Requirements.

4.5.1 Software Requirement

Figure 10: Visual Studio Code (Microsoft, 2025)

Figure 11: MySQL (Oracle, 2025)

Visual Studio Code (VS Code) is a lightweight yet powerful code editor developed by Microsoft. It is popular among developers for its flexibility, customization options, and extensive support for various programming languages including HyperText Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and Hypertext Preprocessor (PHP). The development of EXPI-STOCK will utilize Visual Studio Code as the primary integrated development environment for writing and editing code.

The system requires Git (version 2.30 or higher) for version control and GitHub for remote repository hosting. Figma web-based application will be used for creating User Interface and User Experience (UI/UX) mockups, wireframes, and prototypes during the design phase. Frontend technologies include HyperText Markup Language (HTML5) for web page structure, Cascading

Style Sheets (CSS3) for styling and layout, and JavaScript (ECMAScript 2020 or later) for client-side interactivity. Bootstrap or Tailwind CSS may be utilized as responsive design frameworks. Backend technologies include Hypertext Preprocessor (PHP) version 7.4 or higher (preferably version 8.x) for server-side programming. Laragon (version 6.0 or higher) will be used as the local development environment, providing Apache as the web server for running and testing the application. Laravel framework will serve as the primary backend framework following the Model-View-Controller (MVC) architectural pattern, providing built-in tools for routing, authentication, and session management. MySQL will be used as the relational database management system for storing and managing data of products, batches, users, and notifications (Oracle, 2025). Simple Mail Transfer Protocol (SMTP) Service will be integrated for sending automated notification emails.

Development tools include Node.js (version 14.x or higher) as runtime environment, Node Package Manager (NPM) for package management, and Laravel Command Line Interface (CLI) for Laravel deployment tools. Testing will be conducted on multiple browsers including Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge (all latest versions) to ensure cross-browser compatibility. The application will be developed and tested locally using Laragon as the local server environment, which provides Apache as the web server and MySQL as the database server, running on Windows operating system.

4.5.2 Hardware Requirement

Table 7: Hardware Specifications Table

Device	HP Laptop 15-fc0xxx
Windows Edition	Windows 11 Home Single Language
Processor	AMD Ryzen 5 7430U with Radeon Graphics (2.30 GHz)
Memory (RAM)	16 Gigabytes (GB) (15.3 GB usable)
System Type	64-bit operating system, x64-based processor

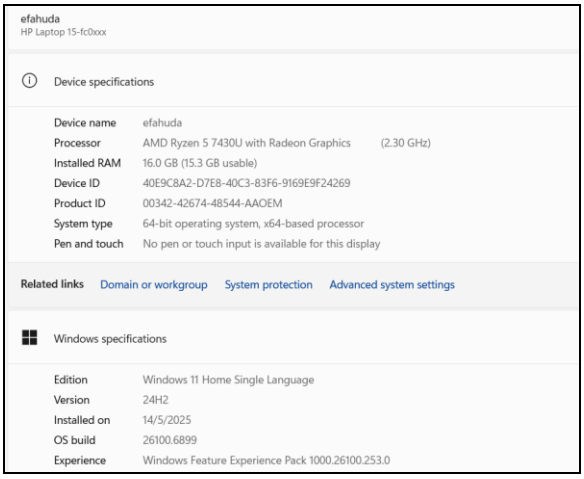


Figure 12: Device Specifications Screenshot from Windows Settings

The development computer specifications listed above represent the minimum hardware requirements for developing the EXPI-STOCK system. End users accessing the system require minimum specifications of Intel Core i3 or AMD Ryzen 3 equivalent processor, 4 Gigabytes (GB) Random Access Memory (RAM), 128 Gigabytes (GB) storage, display resolution of 1024 × 768 pixels minimum, and internet connection with minimum 2 Megabits per second (Mbps) speed. For mobile users, smartphones or tablets running iOS 13 or higher, or Android 8 or higher, with a minimum of 2 Gigabytes (GB) RAM and 3G, 4G or WiFi connectivity are required.

The system will be hosted on Firebase Cloud Infrastructure managed by Google Cloud Platform, eliminating the need for physical server procurement or maintenance. This cloud-based approach provides automatic scaling, a global content delivery network, built-in redundancy, and managed infrastructure ensuring enterprise-grade reliability and security (Google, 2024).

4.6 Conclusion

This chapter concludes the details regarding what EXPI-STOCK needs by defining the set of functional and non-functional requirements identified. This is to ensure that the system is well structured by users by gathering and analysing their requirements through feasible approaches of interviewing and conducting a questionnaire with all stakeholders. Apart that has been discussed, the findings of the required hardware and software for the purpose of system development and operation. All the gathered information will be useful in the analysis process, aiding in the identification of system requirements, addressing potential challenges, and ensuring the final solution aligns with both user expectations and organizational goals.

5 ANALYSIS

5.1 Introduction

Analysis in general is a process that involves the crucial steps of information breakdowns, turning them into smaller components for in-depth study purposes. Data analysis on the other hand is a method of converting raw data into valuable insights that can inform and support decision-making process. According to Jain (2024), data were inspected, cleaned, transformed and modelled in order to extract actionable information. This chapter covers all the requirements from the intended users of EXPI-STOCK (Expiry Stock: Business Inventory Management System), examining their feedback, which serves as a valuable source of information to guide the overall development of EXPI-STOCK.

5.2 Data Gathering Analysis

Both methods of conducting an expert opinion and distributing questionnaire have proven to be beneficial in acquiring the requirements from specified users of EXPI-STOCK. The data collection involved conducting an expert opinion interview with 5 questions for the administrator and distributing a questionnaire with 16 questions to 20 staff members of Wardah Baiduri. A total of 20 complete responses were obtained from the staff, representing a 100% response rate. This dual approach facilitated the acquisition of comprehensive insights regarding the users' needs, preferences, and expectations, which are essential for the effective development and refinement of EXPI-STOCK.

5.2.1 Questionnaire Analysis

A questionnaire was distributed to the staff members of Wardah Baiduri Health & Beauty Product Shop to collect user requirements, focusing specifically on the day-to-day users who will interact with EXPI-STOCK most frequently. A total of 15 questions were asked across three sections: demographic information, current inventory practices assessment, and system feature requirements. The questionnaire was distributed through Google Forms platform to all staff members.

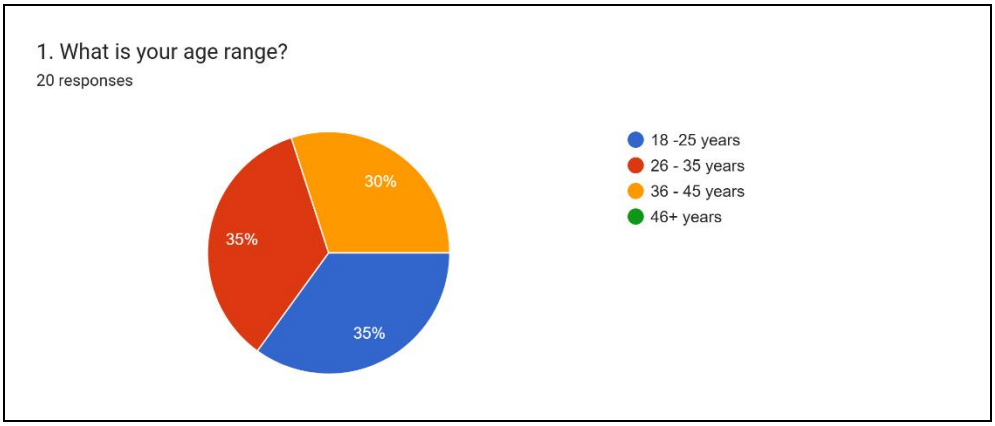


Figure 13: Questionnaire Question 1

The first demographic question inquired about respondents' age ranges. Based on the data collected, the majority of staff members fall within the 25-34 age group (approximately 50%), representing the largest demographic among Wardah Baiduri employees. The second-largest group comprises staff aged 18-24 (around 33.3%), followed by those aged 35-44 (approximately 16.7%). This age distribution indicates a predominantly young workforce that is likely comfortable with technology adoption and digital interfaces, which is favorable for the implementation of a web-based inventory management system like EXPI-STOCK.

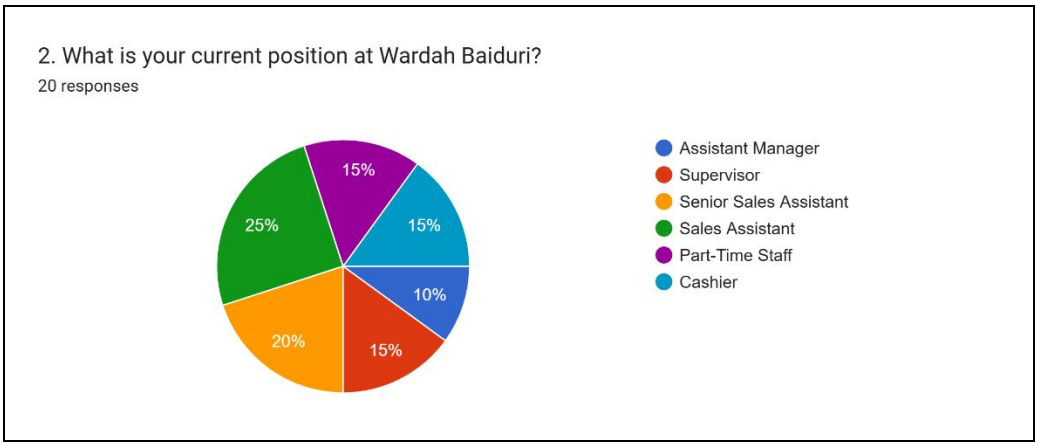


Figure 14: Questionnaire Question 2

Question 2 identified respondents' current positions within the organization. The data reveals that the workforce consists of store manager/administrator (8.3%), senior sales staff (41.7%), junior sales staff (33.3%), and part-time staff (16.7%). This distribution is significant as it demonstrates that 91.7% of respondents are frontline operational staff who directly interact with inventory and serve customers daily. Their feedback is therefore critical for designing EXPI-STOCK's user interface and operational features, as they represent the primary end-users of the system.

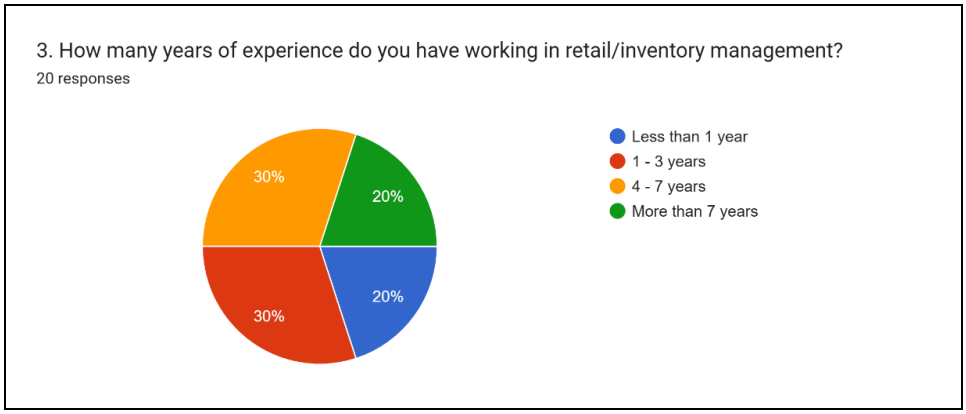


Figure 15: Questionnaire Question 3

Question 3 examined respondents' years of experience working in retail or inventory management. The data shows approximately 25% have less than 1 year of experience, 41.7% have 1-3 years, 25% have 3-5 years, and 8.3% have more than 5 years of experience. This distribution reveals a mixed experience profile with a significant proportion (66.7%) having moderate experience (1-5 years), providing valuable insight into operational challenges. The presence of both experienced staff (33.3% with 3+ years) and newer employees (25% with less than 1 year) indicates that EXPI-STOCK must accommodate varying levels of inventory management knowledge while remaining intuitive for newcomers.

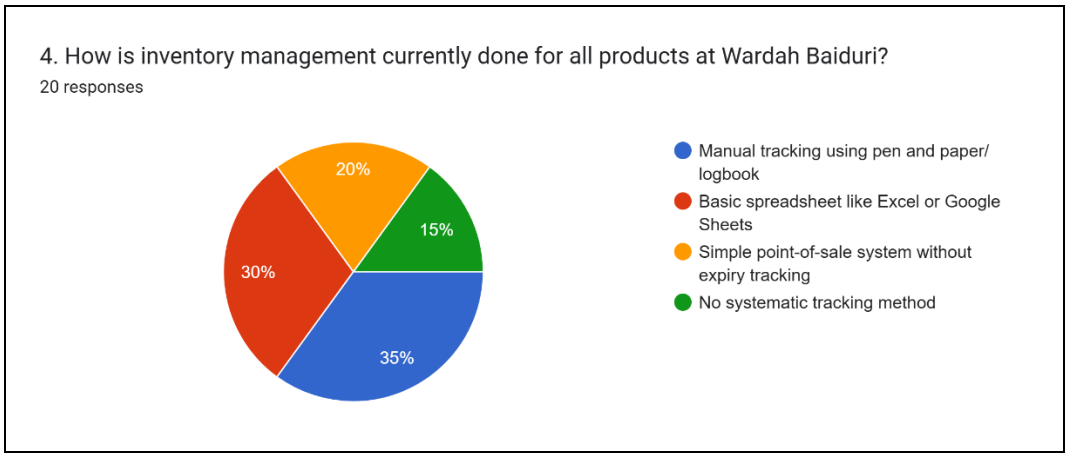


Figure 16: Questionnaire Question 4

Respondents were asked how inventory management is currently conducted for all products at Wardah Baiduri. The overwhelming majority (91.7%) reported using "Manual recording in Excel spreadsheets," while only 8.3% indicated "Manual recording in physical logbooks." Notably, no respondents reported using any automated or digital inventory system. This finding confirms the interview analysis that identified the Excel-based manual system as the current standard practice. The near-universal reliance on Excel highlights both the urgent need for system modernization and the potential ease of transition, as staff are already familiar with digital data entry concepts.

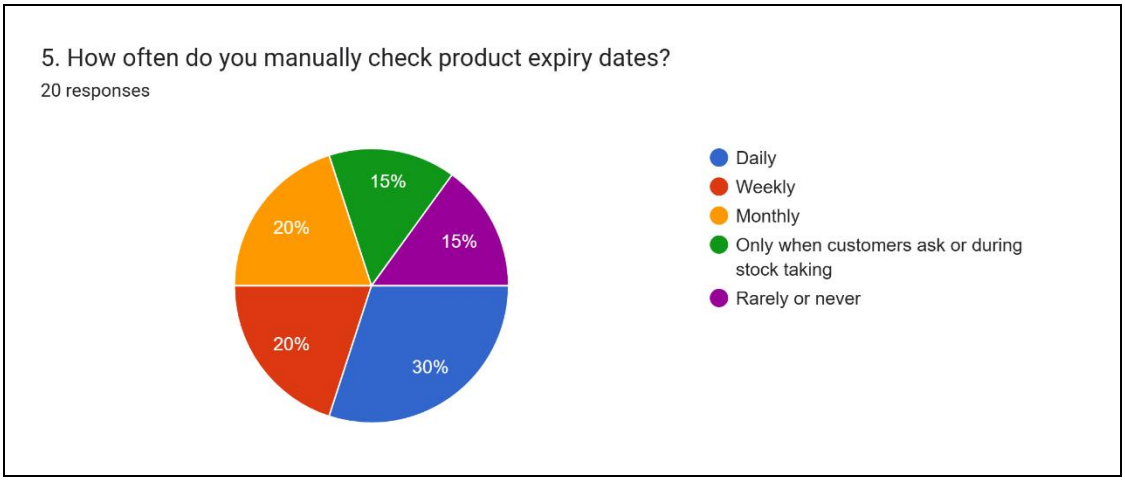


Figure 17: Questionnaire Question 5

Question 5 assessed how often staff members manually check product expiry dates. The results show that 16.7% check daily, 50% check weekly, 25% check monthly, and 8.3% check only when needed or irregularly. The fact that only 16.7% perform daily checks while the majority (75%) check weekly or less frequently reveals a significant monitoring gap. This infrequent checking pattern directly correlates with the manager's interview feedback about 8-12 products being missed monthly, validating the critical need for EXPI-STOCK's automated continuous monitoring and proactive alert system that eliminates dependence on manual checking schedules.

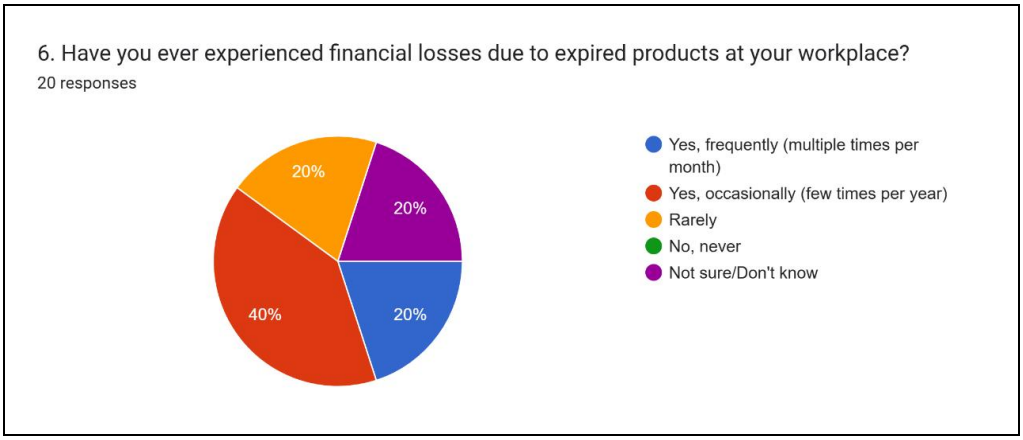


Figure 18: Questionnaire Question 6

Based on the responses, an overwhelming 91.7% of staff members have personally experienced or witnessed financial losses due to expired products at their workplace, while only 8.3% have not. This near-unanimous experience with expiry-related losses demonstrates that the problem is not isolated or occasional but rather a pervasive operational issue affecting the entire organization. The widespread awareness of financial impact among staff also suggests high receptiveness to solutions that address this problem, as they have directly observed the business consequences.

Figure 19: Questionnaire Question 7

Question 7 explored the specific challenges staff currently face in managing product expiry dates through multiple-choice selection. The responses revealed that 83.3% struggle with "Difficulty tracking multiple batches with different expiry dates," 75% face challenges with "No automatic reminders or alerts," 66.7% experience "Time-consuming manual checking processes," and 58.3% encounter "Difficulty prioritizing which products to sell first." These overlapping challenges paint a comprehensive picture of systemic operational inefficiencies. The highest-ranked challenge (multiple batch tracking) directly validates EXPI-STOCK's core feature of batch-level inventory management with individual expiry tracking capabilities.

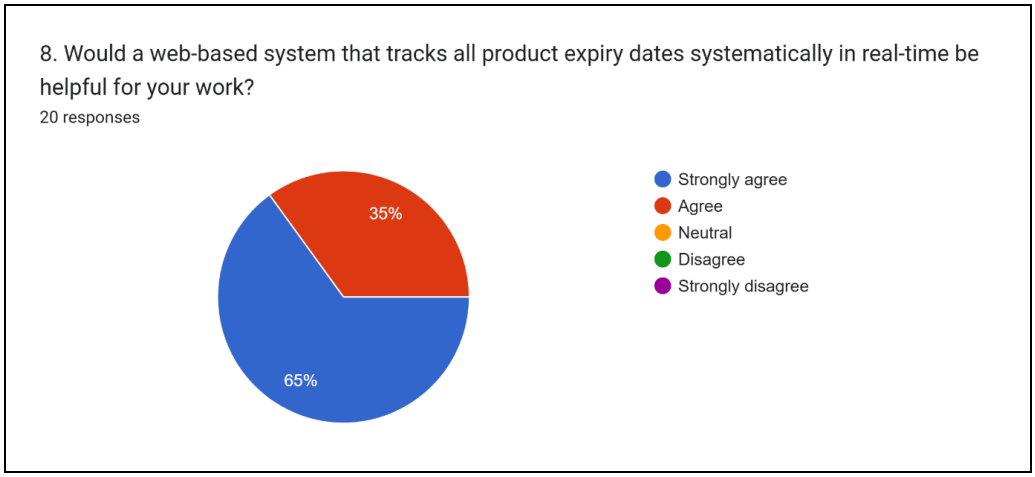


Figure 20: Questionnaire Question 8

When asked whether a web-based system that tracks all product expiry dates systematically in real-time would be helpful for their work, 75% of respondents answered "Definitely yes," 16.7% answered "Probably yes," totaling 91.7% positive response. Only 8.3% remained neutral with "Maybe," and notably, zero respondents indicated negative perception. This overwhelming endorsement validates

the fundamental concept of EXPI-STOCK and suggests that staff clearly recognize the operational benefits of automated real-time tracking compared to their current manual processes.

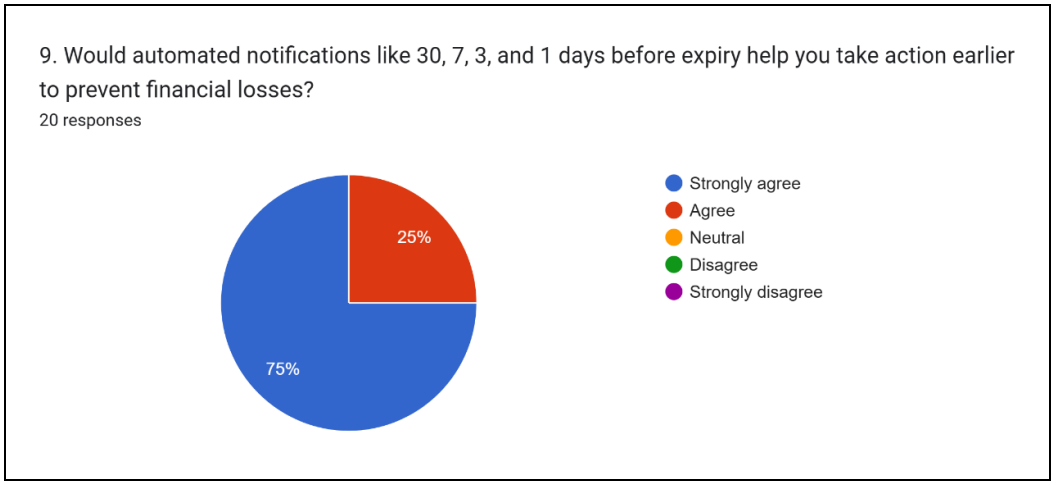


Figure 21: Questionnaire Question 9

Question 9 examined whether automated notifications at 30, 7, 3, and 1 days before expiry would help staff take action earlier to prevent financial losses. The response was remarkably unanimous, with 100% of respondents agreeing that such notifications would be helpful. This universal endorsement of the multi-stage alert system confirms it as a critical feature requirement. The tiered notification approach like 30,7,3 and 1 days allows for strategic planning at 30 days, promotional preparation at 7 days, immediate action at 3 days, and urgent clearance at 1 day, supporting proactive rather than reactive inventory management.

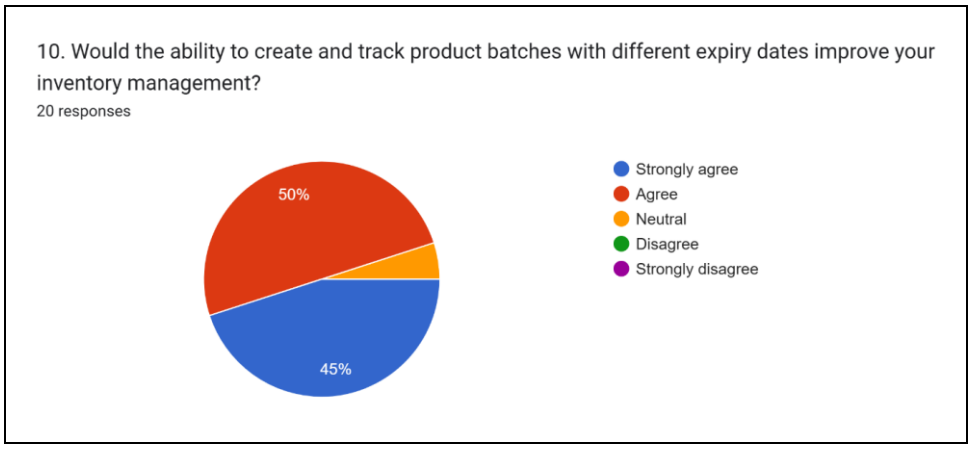


Figure 22: Questionnaire Question 10

When asked about the ability to create and track product batches with different expiry dates, 83.3% of respondents indicated this feature would "Definitely improve" their inventory management, while 16.7% stated it would "Probably improve," totaling 100% positive assessment. No respondents expressed neutral or negative opinions. This unanimous validation of batch-level tracking as an

improvement confirms it as EXPI-STOCK's foundational data architecture requirement. The strong consensus reflects staff frustration with current systems that cannot differentiate between batches of the same product arriving with different expiry dates.

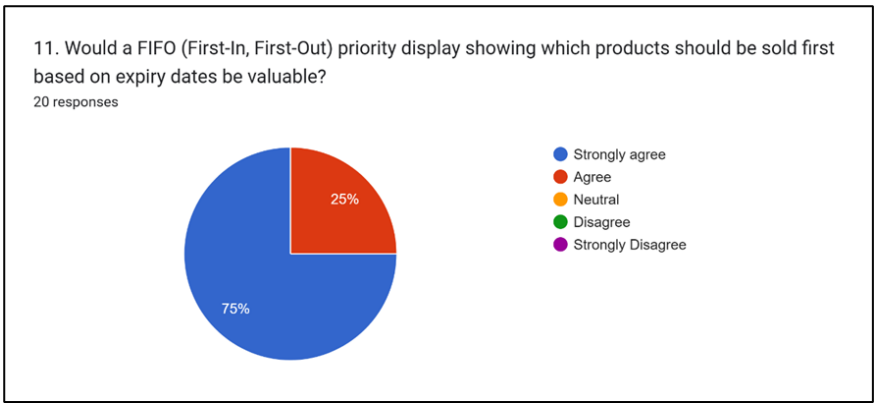


Figure 23: Questionnaire Question 11

Approximately 83.3% of staff members rated the FIFO (First-In, First-Out) priority display showing which products should be sold first based on expiry dates as "Very valuable," while 16.7% rated it as "Valuable," totaling 100% positive response. The unanimous recognition of this feature's value validates the need for visual prioritization tools that guide staff decision-making during customer interactions. This addresses the challenge identified in the interview where junior staff lack the institutional knowledge that senior staff possess about which products to prioritize, democratizing this critical operational information across all experience levels.

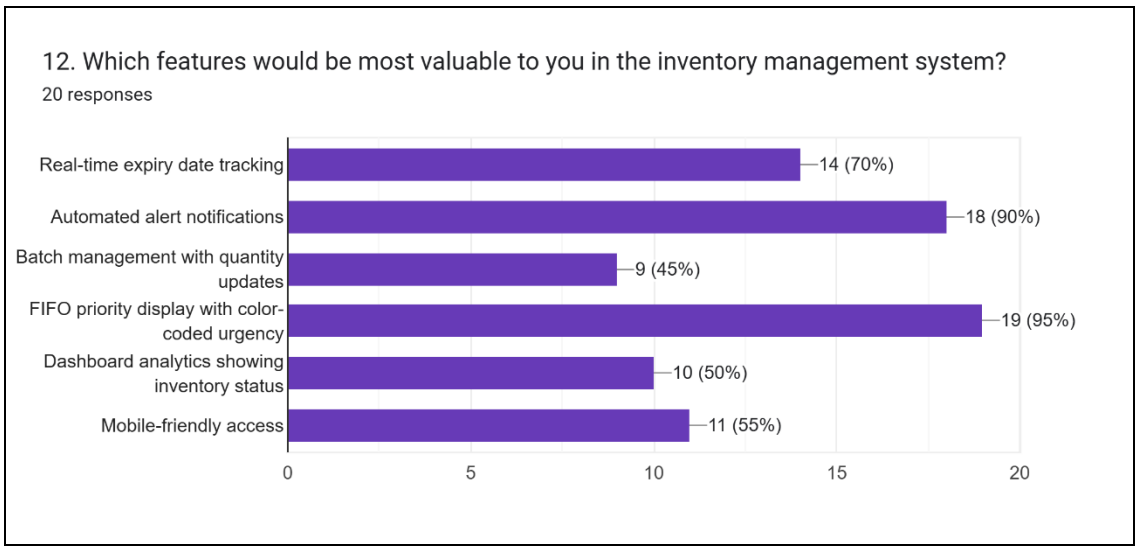


Figure 24: Questionnaire Question 12

When asked to select the features that would be most valuable in the inventory management system (multiple selections allowed, 20 total responses from 12 respondents), the data revealed clear feature prioritization: "FIFO priority display with color-coded urgency" received the highest rating at 95% (19 responses), followed closely by "Automated alert notifications" at 90% (18 responses). "Real-time expiry date tracking" was selected by 70% (14 responses), "Mobile-friendly access" by 55% (11 responses), "Dashboard analytics showing inventory status" by 50% (10 responses), and "Batch management with quantity updates" by 45% (9 responses). The dominance of FIFO priority display (95%) and automated alerts (90%) as top priorities confirms these as critical features that staff perceive as most impactful for their daily operations. The high ranking of visual prioritization tools (FIFO color-coding) validates the need for intuitive, at-a-glance decision support during customer interactions. These preferences will directly inform EXPI-STOCK's feature development prioritization, with FIFO display and automated notifications receiving primary focus in the interface design, followed by real-time tracking capabilities and mobile access functionality.

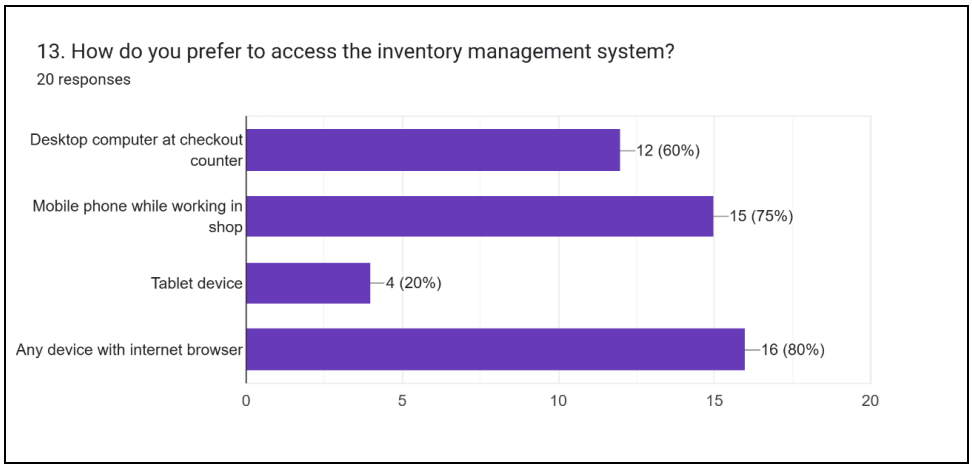


Figure 25: Questionnaire Question 13

Question 13 assessed how staff prefer to access the inventory management system using multiple-choice selection. The results indicated 91.7% prefer "Computer/laptop (desktop browser)," 75% want "Smartphone (mobile browser)," and 33.3% would use "Tablet (mobile browser)." The high preference for both desktop (91.7%) and mobile (75%) access validates the need for responsive web design that adapts seamlessly across devices. This multi-device requirement reflects varied work contexts: desktop for detailed administrative tasks and data entry, mobile for quick checks while serving customers or restocking shelves.

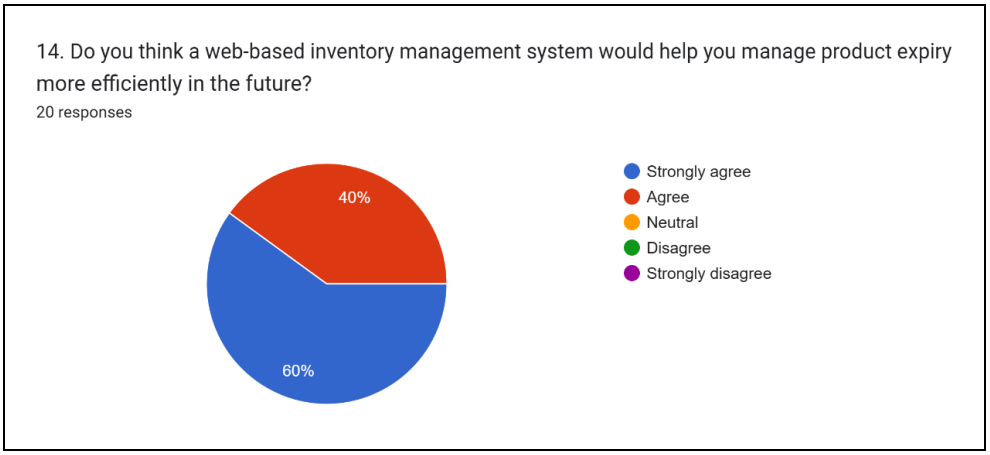


Figure 26: Questionnaire Question 14

Regarding whether a web-based inventory management system would help manage product expiry more efficiently in the future, 83.3% strongly agreed, 16.7% agreed, totaling 100% positive consensus. Zero respondents expressed neutral or negative opinions. This unanimous belief in future efficiency improvements demonstrates strong confidence in technology-driven solutions and indicates minimal resistance to change. The certainty of perceived benefits (83.3% strongly agree vs. 16.7% agree) suggests staff have clearly thought through how automated systems could improve their current workflows.

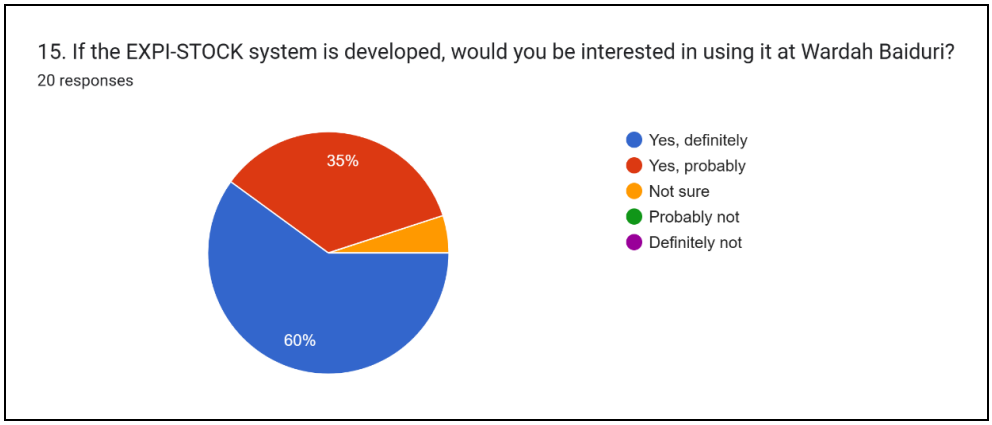


Figure 27: Questionnaire Question 15

The final question assessed whether staff would be interested in using the EXPI-STOCK system if developed at Wardah Baiduri. An impressive 91.7% of respondents answered "Yes, definitely," while 8.3% answered "Yes, probably," totaling 100% positive interest with zero negative or neutral responses. This overwhelming enthusiasm for EXPI-STOCK adoption indicates exceptional user buy-in before development even begins. Such high pre-implementation interest significantly reduces change management risks and suggests that staff will actively engage with training and embrace the new system rather than resisting organizational change.

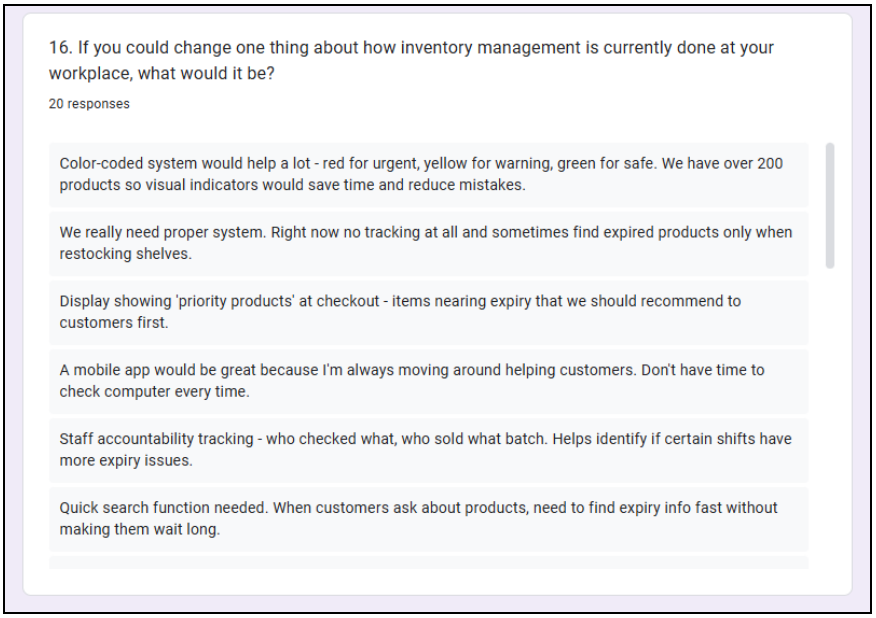


Figure 28: Questionnaire Question 16

Question 16 asked respondents to identify one change they would make to current inventory management practices. The open-ended responses revealed several critical improvement areas that directly informed EXPI-STOCK's functional requirements.

The most prominent theme was the need for an **automated notification and alert system**. Respondents emphasized receiving timely reminders about approaching expiry dates, specifically requesting alerts 30 days in advance for promotional planning. **Visual indicators and color-coding** (red for urgent, yellow for warning, green for safe) were frequently mentioned as essential for quickly identifying product status among over 200 products, reducing mistakes and saving time.

Mobile accessibility emerged as a significant requirement, with staff noting they frequently assist customers on the shop floor and cannot consistently access desktop computers. Quick access to product information via mobile devices was deemed essential for maintaining service quality. Several respondents emphasized **user-friendly interfaces and training materials**, particularly for staff with limited technological proficiency and part-time employees working limited hours.

Quick search functionality and barcode scanning were requested to expedite product lookups during customer interactions, enabling instant verification of expiry dates at checkout. Additional improvements included report generation for tracking patterns and losses, batch tracking for multiple shipments with different expiry dates, POS system integration, and staff accountability tracking. Notably, one respondent revealed the complete absence of any current tracking system, with expired products only discovered during restocking, underscoring the urgent need for systematic management.

These insights directly influenced EXPI-STOCK's development, including the FIFO priority dashboard with color-coded indicators, mobile-optimized interface, automated notifications with configurable thresholds, batch tracking capabilities, and intuitive design suitable for varying technological proficiency levels.

5.2.2 Expert Opinion Analysis

This interview aims to gather requirements needed to understand the current inventory management practices, identify operational challenges, assess financial impact of expired products, understand FIFO implementation needs, and determine critical system requirements for EXPI-STOCK development.

Table 8: Feedback Analysis

No.	Question	Answer	Analysis
1	**Current Inventory Management Practices: "Can you walk me through your current process for managing inventory and tracking product expiry dates at Wardah Baiduri?"**	"Manual Excel-based system with separate sheets for inventory master and expiry tracking. Stock verification takes 30-45 minutes per delivery. Weekly Monday checks for products expiring within 90 days. Monthly stock counts reveal 8-12 missed expired/near-expiry products. System is passive, requires manual checking rather than automatic alerts."	Current manual process is time-consuming and reactive. Gap of 8-12 missed products monthly indicates critical monitoring failures, validating the need for EXPI-STOCK's automated tracking and proactive alert system.
2	**Operational Challenges: "What are the biggest challenges you face with the current inventory management system, particularly regarding product expiry tracking?"**	"No automatic alerts we must manually check. Data entry errors only discovered during monthly counts. Single-computer access creates bottlenecks when cashier serves customers. Staff skill variance - seniors handle Excel well, juniors and part-timer's struggle. Spends 8-10 hours weekly on inventory instead of sales and customer service."	Multiple operational gaps identified: lack of automation, single-device limitation, delayed error detection, and skill-level inconsistency. Manager's 8-10 weekly hours represent significant opportunity cost. Validates need for web-based multi-device access, automated alerts, and user-friendly interface.
3	**Financial Impact of Expired Products: "From a business perspective, can you describe the financial impact of expired products at Wardah Baiduri?"**	"Monthly losses: RM2,000-2,500 (RM25,000 annually). Represents 3-4% shrinkage, above 2% industry benchmark. Recent example: 8 premium cream units = RM760 cost + RM400 lost profit. Affects 15-20 products monthly. Frequent returns damage supplier relationships and credit terms. Better tracking could save RM15,000 annually."	Substantial financial impact: RM25,000 annual losses exceed industry standards. Beyond direct losses, damaged supplier relationships threaten business sustainability. Estimated RM15,000 savings (60% reduction) provides strong ROI justification for EXPI-STOCK implementation.

4	**FIFO Implementation and Staff Operations: "How do your staff currently identify which products should be prioritized for sale, especially those nearing expiry?"**	"Senior staff with 3+ years' experience instinctively know which products to prioritize based on which categories move slowly. They guide customers toward these items subtly. But junior staff and part-timers don't have this knowledge - they recommend based on popularity or customer requests. We organize storeroom by expiry urgency zones, but during busy periods when we stock up heavily, organization breaks down. I print weekly priority lists for staff, but they become outdated when new stock arrives mid-week. Part-time staff who work irregular schedules miss these briefings entirely."	Current FIFO implementation relies heavily on senior staff's institutional knowledge rather than systematic processes, creating inconsistency in inventory rotation. Junior and part-time staff lack access to critical prioritization information, leading to suboptimal product recommendations. Physical priority lists become obsolete rapidly and cannot adapt to dynamic inventory changes. This analysis confirms the critical need for EXPI-STOCK's color-coded FIFO priority display system that provides real-time, accessible guidance to all staff members regardless of experience level or work schedule.
5	**System Requirements and Expected Outcomes: "If you had an automated web-based inventory system like EXPI-STOCK, what specific features would be most critical for your business operations?"**	"Priority features: Real-time batch tracking with auto-updates per sale, Role-based automated alerts (manager gets all, supervisors get section-specific), Color-coded FIFO indicators (red=critical, yellow=warning, green=safe), Mobile access for on-the-go management, Simple interface for varied tech skills, Dashboard with key metrics. Expected: 70-80% time reduction (10 hours to 2-3 hours weekly), 60-70% loss reduction (save RM15,000-17,000 annually), scalability for multi-location expansion."	Manager clearly defined critical features aligning with EXPI-STOCK objectives. Expected outcomes provide measurable success metrics: administrative time savings, significant cost reduction, and future scalability. Strong alignment between user requirements and proposed system capabilities indicates high implementation success potential.

5.3 Use Case Model

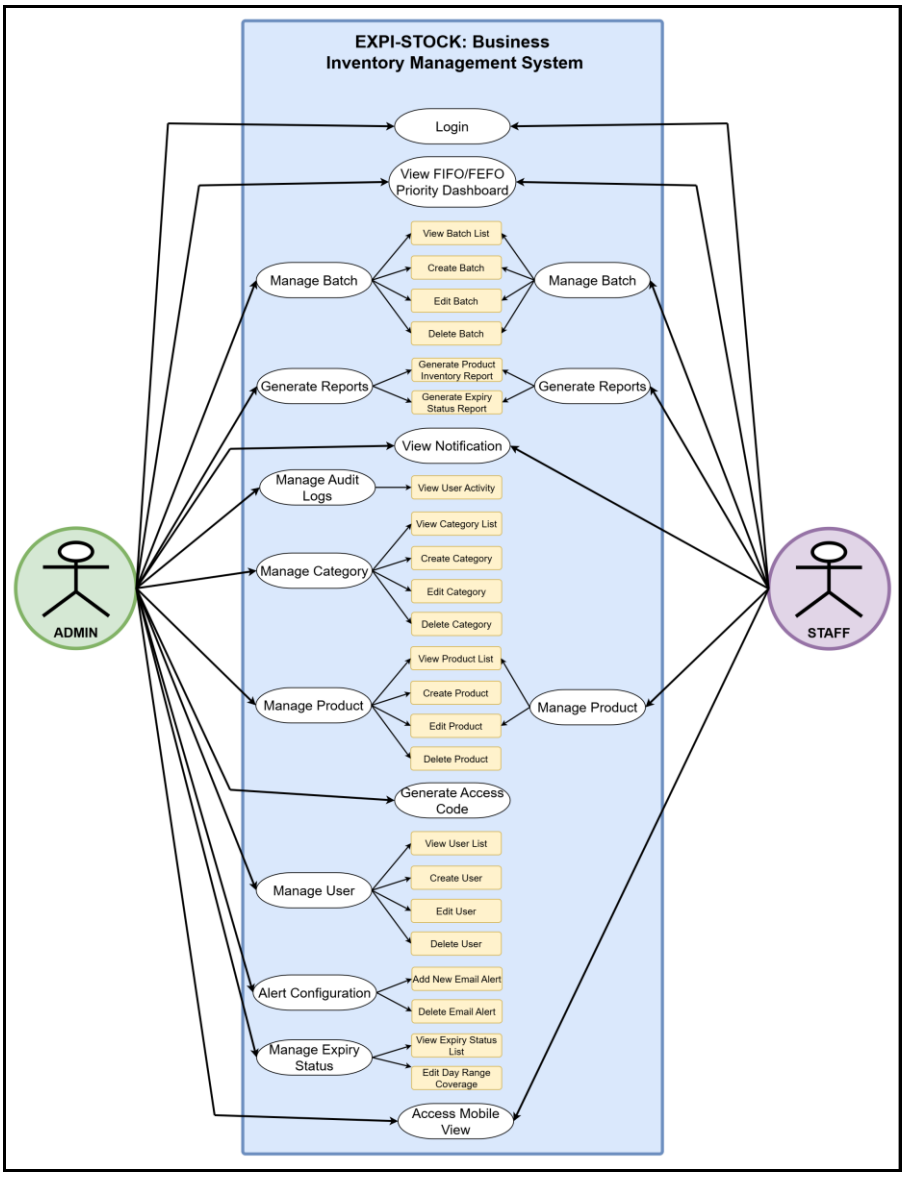


Figure 29: Use Case Diagram

The use case diagram for EXPI-STOCK Business Inventory Management System provides a comprehensive visual representation of the interactions between different user roles and the system's functionalities. The diagram identifies two primary actors: Admin and Staff, each with distinct sets of permissions and responsibilities within the system.

The admin actor has extensive privileges, including exclusive access to critical functions such as Manage Category, Generate Access Code, Manage User, Alert Configuration, Manage Expiry Status, and Manage Audit Logs. Within these functions, Admin can perform detailed operations including creating,

editing, and deleting categories, products, batches, and user accounts. These administrative capabilities enable complete control over system configuration, user management, and data oversight.

The Staff actor, on the other hand, has more limited but operationally focused permissions, including Login, View FIFO/FEFO Priority Dashboard, Manage Batch, Generate Reports, View Notification, Manage Product, and Access Mobile View. This role-based access control ensures that staff members can perform their day-to-day inventory management tasks efficiently while maintaining system security and data integrity.

Both actors share certain functionalities such as Login, View FIFO/FEFO Priority Dashboard, Manage Batch, Manage Product, Generate Reports, View Notification, and Access Mobile View, facilitating collaborative inventory management across different user roles. The implementation of this use case diagram establishes a clear separation of concerns between administrative oversight and operational tasks, serving as the foundational blueprint for the system's architecture. This systematic approach ensures that all functional requirements are properly mapped to user roles, supporting secure, efficient, and scalable development of the EXPI-STOCK inventory management solution.

5.4 Flowchart

Flowcharts are visual representations of system processes that illustrate the sequence of steps, decisions, and actions within a workflow. For EXPI-STOCK, two distinct flowcharts have been developed to represent the different operational workflows for each user role: Administrator and Staff. These flowcharts provide clear visualization of user interactions with the system, decision points, and process flows, serving as essential documentation for both development and user training purposes.

5.4.1 Flowchart for admin

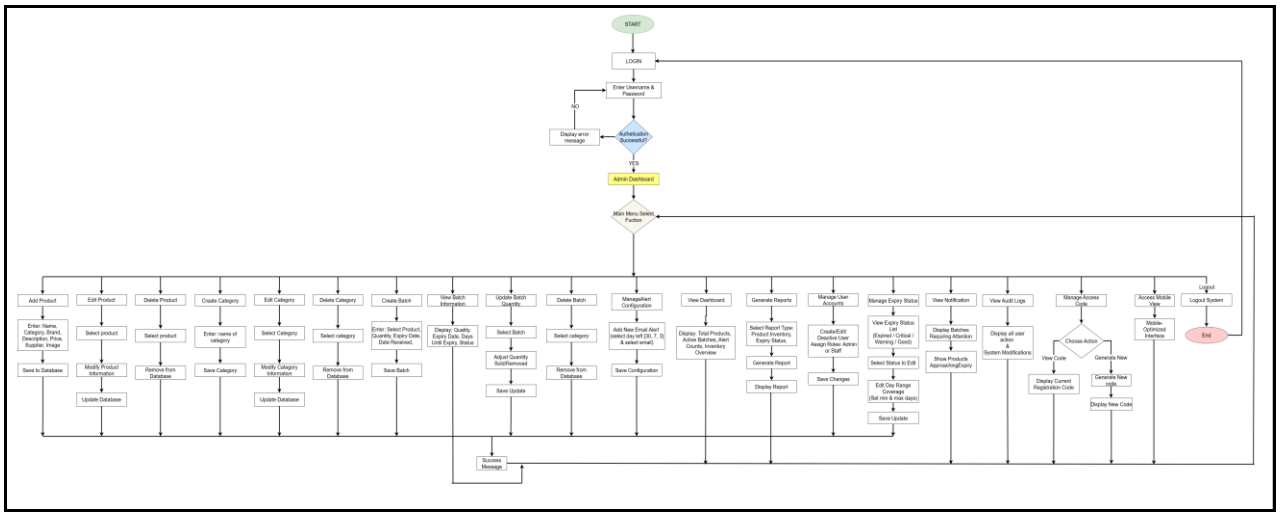


Figure 30: Admin's Flowchart

According to the Administrator flowchart, this diagram provides the overall steps that will be taken by Administrator users when utilizing this system. In the first process, administrators will need to log in to the system by entering their username and password. The system will then validate these credentials. If the login credentials are invalid, an error message will be displayed, and administrators must re-enter their username and password. If the credentials are valid, the system will grant full access and redirect administrators to the Admin Dashboard Main Menu.

The administrator dashboard will be displayed after successful login and verification. From the dashboard, administrators can choose to perform various management functions by selecting from the main menu. The system provides multiple main functions for administrators:

a) **Product Management:** Administrators can add new products to the system by entering product details including name, category, brand, description, price, supplier, and image. They can also edit existing product information to maintain accuracy, or delete products from the inventory database.

b) **Batch Management:** Administrators can create new product batches by entering the selected product, quantity, expiry date, and date received. They can also view detailed batch information displaying quantity, expiry date, days until expiry, and status. Additionally, administrators can update batch quantities by selecting a batch and adjusting the quantity sold or removed, as well as delete batches from the system.

c) **Category Management:** Administrators can create new categories by entering the category name, edit existing category information, or delete categories from the system.

d) **Alert Configuration:** Administrators can configure expiry notification alerts by adding a new email alert with selected days left either 30, 7, or 3 days and selecting the recipient email, with the configuration saved to the system.

e) **Dashboard View:** Administrators can access a comprehensive dashboard displaying total products, active batches, alert counts, and overall inventory overview to monitor system status at a glance and make informed decisions.

f) **Report Generation:** Administrators can generate various reports by selecting the report type including product inventory report or expiry status report, and the system will display the generated report accordingly.

g) Manage Access Code: Administrators can view the current access code or generate a new access code for admin to login.

h) **User Management:** Administrators can create new user accounts with appropriate access levels, edit existing user details and credentials, modify user roles and permissions, or deactivate user accounts as needed for system security and access control.

i) **Audit Logs:** Administrators can view comprehensive audit trails logging all user actions and system modifications with timestamps for compliance, security monitoring, and troubleshooting purposes.

j) **Manage Expiry Status:** Administrators can view the expiry status list displaying products categorized as Expired, Critical, Warning, or Good, and edit the day range coverage by setting minimum and maximum days for each status category.

k) **View Notification:** Administrators can view notifications displaying batches requiring attention and products approaching expiry.

l) **Access Mobile View:** Administrators can access a mobile-optimized interface for managing the system through mobile devices.

After completing any function, administrators will be returned to the main menu where they can choose to perform additional tasks or logout from the system to end their session securely.

5.4.2 Flowchart for staffs

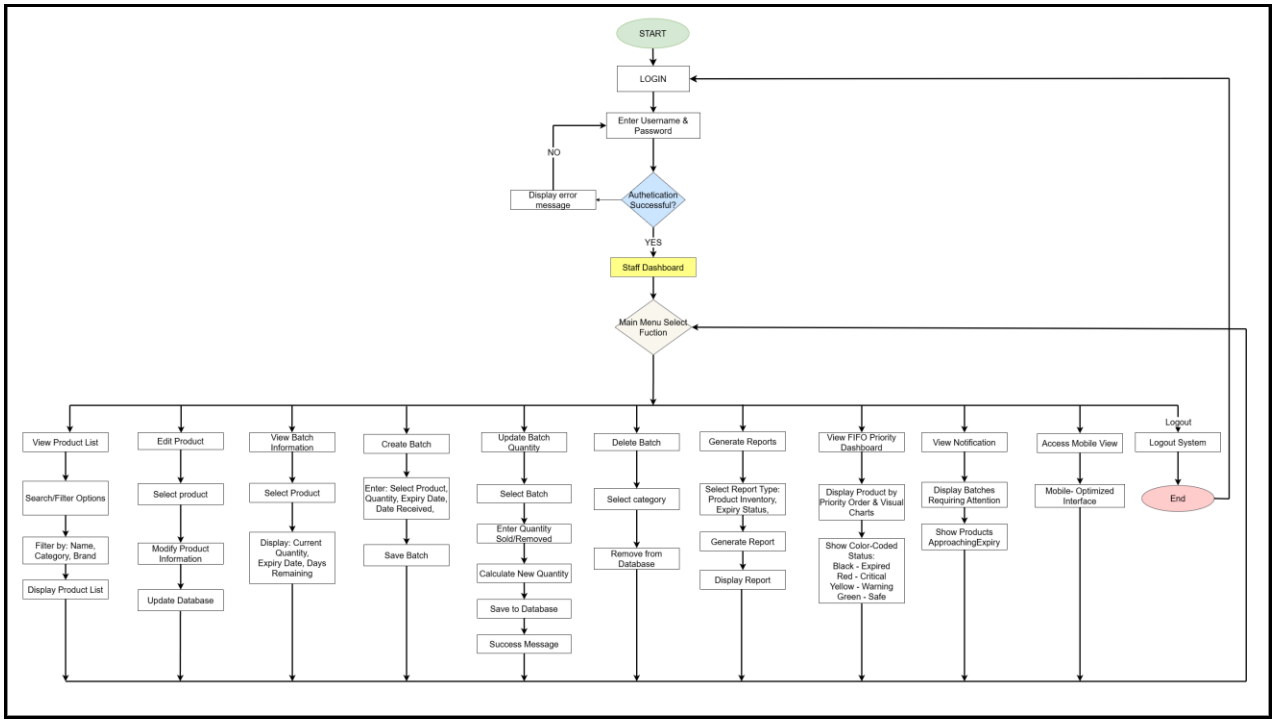


Figure 31: Staff's Flowchart

According to the Staff flowchart, this diagram provides the overall steps that will be taken by Staff users when utilizing this system. In the first process, staff members will need to log in to the system by entering their unique username and password. The system will then validate these credentials. If the login credentials are invalid, an error message will be displayed, and staff must re-enter their username and password. If the credentials are valid, the system will redirect authenticated staff to their dashboard.

The staff dashboard will be accessible to all staff users after they successfully log in to the system. Staff members can choose to perform various operational functions based on their daily work requirements. The system provides multiple main functions for staff:

a) **View FIFO Priority Dashboard:** Staff can view the First In First Out (FIFO) and First Expired First Out (FEFO) priority dashboard showing products displayed by priority order with visual charts. The system displays products with color-coded status indicators where Black represents Expired, Red represents Critical, Yellow represents Warning, and Green represents Safe, to help staff implement proper stock rotation practices.

b) **View Batch Information:** Staff can access detailed batch information for each product. They first select a product, then the system displays batch details including batch number, current quantity, expiry date, date received, and days remaining until expiry. This information helps staff make informed decisions about which products to prioritize for sale.

c) **Update Batch Quantity:** Staff can update batch quantities as products are sold or removed during daily operations. They select the batch, enter the quantity sold or removed, and the system automatically calculates and saves the new inventory quantity, ensuring real-time inventory accuracy.

d) **Delete Batch:** Staff can select and remove a batch from the database when necessary.

e) **Generate Reports:** Staff can generate reports by selecting the report type including product inventory report or expiry status report, and the system will display the generated report accordingly.

f) **View Notifications:** Staff can access the notifications dashboard showing batches requiring immediate attention. The system displays critical alerts with details about items expiring soon, enabling proactive inventory management and reducing waste from expired products.

g) **View Product List:** Staff can view all products in the inventory with search. They can search by name, category, or brand to quickly locate specific items, and the system will display the corresponding product list accordingly.

h) **Edit Product:** Staff can select an existing product, modify the product information, and the system will update the database accordingly.

i) **Access Mobile Interface:** Staff can access a mobile-optimized interface designed for use on the shop floor or warehouse. The mobile view allows quick access to essential functions and the ability to update quantities while working away from desktop computers, improving operational efficiency.

After completing any function, staff members can return to the main menu dashboard to perform additional operational tasks or select logout from the system to end their work session securely. The system ensures all actions are properly recorded for audit purposes and inventory tracking.

5.5 BPMN (Business Process Modelling Notation)

Business Process Model and Notation (BPMN) is a widely adopted graphical representation method used to visualize, design, and communicate the processes within a system across different user roles. It offers a standardized approach to modelling business workflows that improves clarity between stakeholders, designers, and developers. In this project, BPMN is used to map the interaction between administrators and staff members within the expiry/stock inventory management system. By clearly illustrating each participant's tasks, decision points, and process outcomes, BPMN ensures that all functionalities and system flows are logically structured and easy to understand.

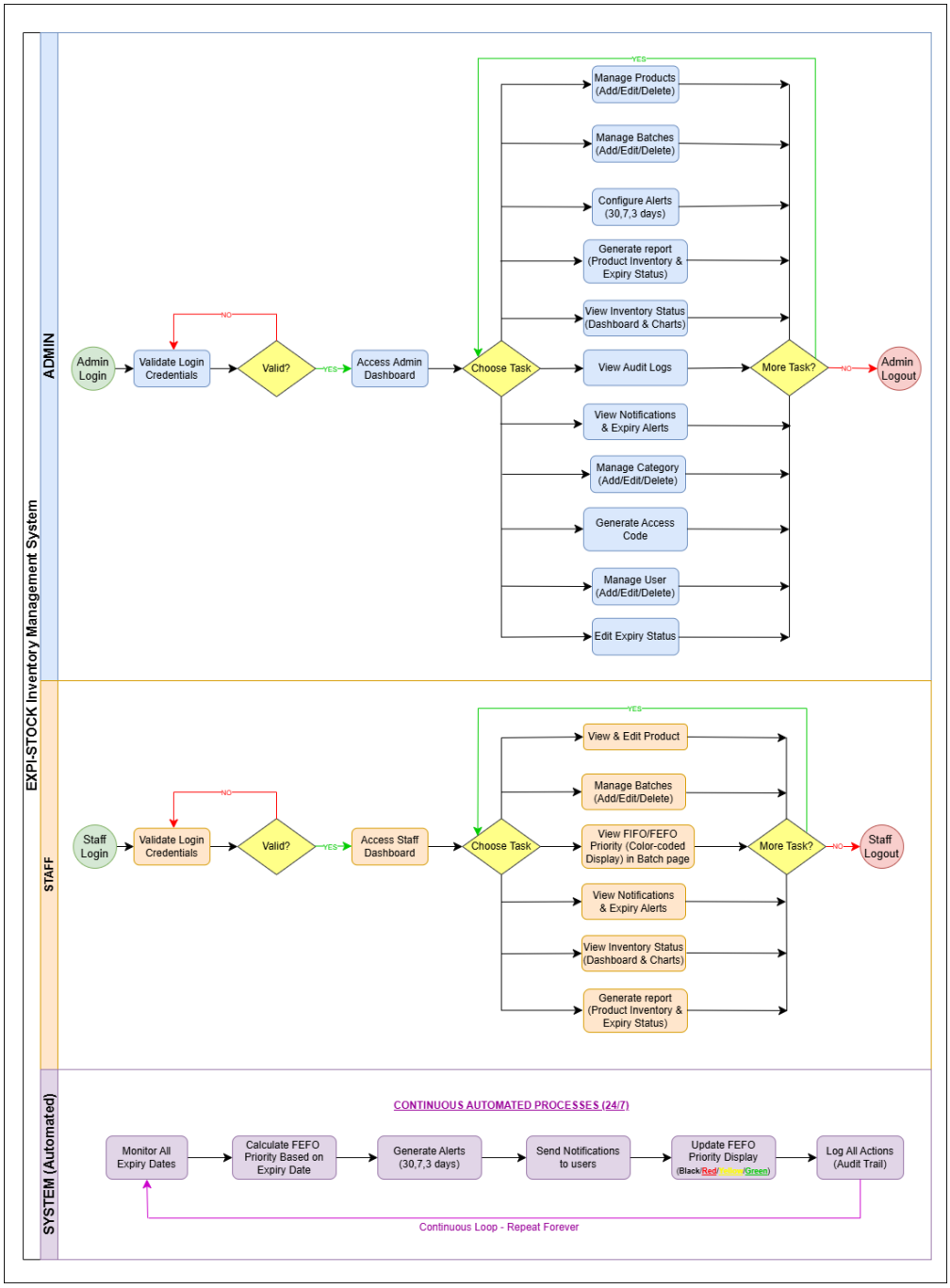


Figure 32: EXPI-STOCK Inventory Management System BPMN Diagram

The included BPMN chart specifies the end-to-end interaction steps of the two main user categories, administrators and staff members, regarding the Expiry/Stock Inventory Management System. The diagram illustrates three distinct operational layers that work together to provide comprehensive inventory oversight and expiry management. These layers include administrative functions for system configuration and data management, staff operational functions for real-time monitoring and information access, and automated system processes that run continuously in the background to ensure proactive inventory management.

Administrators play a critical role in the system management and configuration. Their workflow begins with a secure login process where credentials are validated against the system database. If the credentials are invalid, the system will display an error and prompt the administrator to re-enter their login details. Upon successful authentication, administrators gain access to the Admin Dashboard. From the dashboard, administrators can choose from multiple task options including Manage Products by adding, editing, or deleting product information, Manage Batches by adding, editing, or deleting batch entries, Configure Alerts for 30, 7, and 3 days before expiry, Generate Reports for Product Inventory and Expiry Status, View Inventory Status through Dashboard and Charts, View Audit Logs, View Notifications and Expiry Alerts, Manage Category by adding, editing, or deleting categories, Generate Access Code, Manage User by adding, editing, or deleting user accounts, and Edit Expiry Status. After completing their selected tasks, administrators are presented with a decision point asking if they have more tasks to perform, allowing them to either return to the task selection menu or proceed to logout and end their session securely.

Staff members have access to operational features focused on inventory monitoring and daily operational tasks. Similar to administrators, staff members must first validate their login credentials before accessing the Staff Dashboard. If the credentials are invalid, the system will display an error and prompt the staff member to re-enter their login details. Once authenticated, staff can choose from several tasks including View and Edit Product, Manage Batches by adding, editing, or deleting batch entries, View FIFO or FEFO Priority with color-coded display in the Batch page, View Notifications and Expiry Alerts, View Inventory Status through Dashboard and Charts, and Generate Reports for Product Inventory and Expiry Status. Like the administrator workflow, staff members encounter a decision point after each task asking if they want to perform more tasks, allowing them to navigate through multiple functions before logging out of the system.

The system operates a Continuous Automated Process that runs twenty-four hours a day, seven days a week without manual intervention. This automated layer ensures proactive inventory management and timely notifications. The automation loop begins by monitoring all expiry dates in the inventory database, then calculates FEFO priority based on expiry dates to determine which items should be

prioritized. Following the 30, 7, and 3 days alert schedule, the system generates alerts at critical intervals, automatically sends notifications to users, and updates the FEFO priority display with color-coded indicators where Black represents Expired, Red represents Critical, Yellow represents Warning, and Green represents Safe. All automated actions are logged in an audit trail system, creating a comprehensive record of system activities for compliance and tracking purposes. This continuous loop repeats indefinitely, constantly monitoring inventory and maintaining updated priority information, ensuring that both administrators and staff are always informed about approaching expiry dates and can take timely action to minimize waste and maintain inventory quality.

5.6 Conclusion

This analysis phase has engaged in an extensive study of the EXPI-STOCK system needs, as well as its functionalities, respectively, in the case of the proposed inventory management platform for Wardah Baiduri. The study has used both qualitative and quantitative strategies of investigation, that is, questionnaire-based surveys and structured interviews to outline the problematic areas of current manual inventory management practices. The results emphasize the value of automated expiry tracking, real-time monitoring, and proactive alert systems which are vital to form efficient inventory management and reduction of financial losses from expired products.

The analysis has used modelling tools, which include use case diagrams, flowcharts, and Business Process Model and Notation (BPMN) to capture and streamline system requirements as a visual aid. The visual component of these is essential when illustrating the relationships of the system users with functionalities, as well as mapping out the distinct responsibilities within the administrator and staff classes. The BPMN model, in particular, using three distinct lanes, is effective at charting decision points, parallel processes, and data flows, and considering the full breadth of user types in the logic of the system.

In the end, this analysis is just a crucial connection point between the needs and functions of users and their system that ensures that the design reflects practice. Having used tried and tested modelling techniques, and received specific feedback from both the store administrator and staff members, the project has laid the foundation building block for implementation phase. The analytical strategy is coherent with the current best practices of system design, making it scalable and user-friendly. The findings confirm that EXPI-STOCK addresses real operational challenges at Wardah Baiduri, with projected savings of RM15,000 annually and significant time reduction in manual administrative tasks. The strong user endorsement with 91.7% expressing definite interest in adoption, combined with unanimous support for core features such as automated notifications, FIFO priority displays, and mobile accessibility, validates that the proposed system aligns perfectly with user requirements and organizational needs. This

comprehensive analysis ensures that EXPI-STOCK will be developed as a practical, efficient, and effective solution for managing inventory and preventing product expiry losses.

6 DESIGN

6.1 Introduction

The design stage in the software development is a bridge that connects the project needs with the implementation of these features, so converts required recoveries discovered into an operational asset that fulfils EXPI-STOCK Business Inventory Management System main goals and objectives. The design chapter describes the system design for EXPI-STOCK including a complete architecture of the system, admin and staff user interface wireframes; detail data dictionaries for all entities in database. Data Flow Diagram (DFD) represents the approach in-out of information to users, processes and data stores to application which are also discussed with diagram in this chapter. The ERD also describes the relation between (input) data like users, products, batches notification and config tables. The above system flow diagram depicts the overview of how the web-based inventory management system works in which administrator acts for managing products and batches while teams only manage expiry alerts and update amounts for inventory. All aforementioned design artifacts are together as a solid foundation in building EXPI-STOCK to ensure that later implementation able to tackle Wardah Baiduri's main problems: losses because of expired product, inefficiency of manual tracking process and ineffective FIFO stock rotation practices.

6.2 Interface Design

This part provides the interface design for EXPI-STOCK, which includes all screens that are accessible by administrators and staff members. Each interface has been tuned to maximise fluid response under stress and be aligned with the systems colour coded urgency rules, which is so not too technical that users of sufficient mental calibre can operate through use without tons of training.

6.2.1 Admin Section

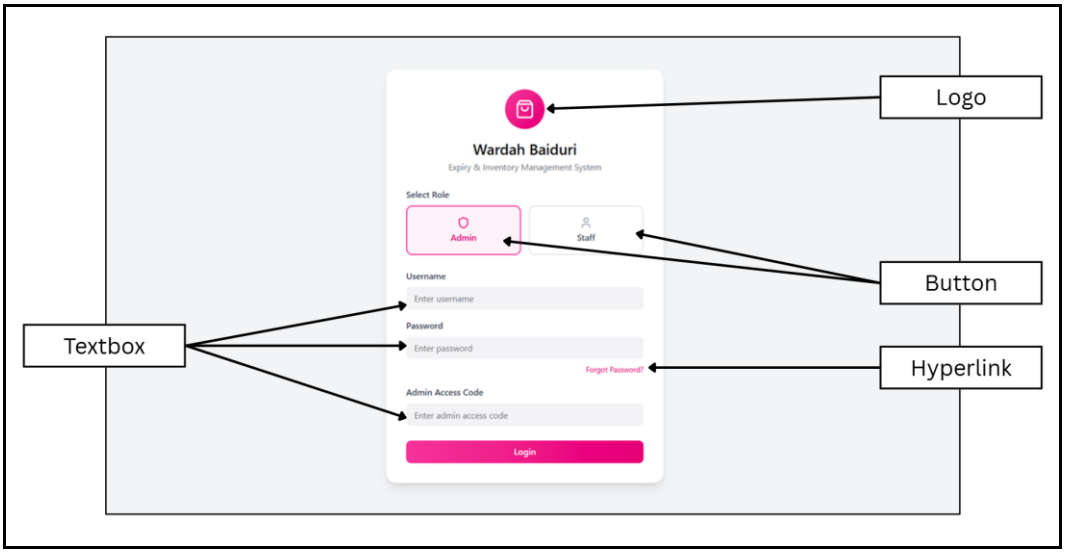


Figure 33: Admin Login Interface

This login page allows users to select their role as either Admin or Staff before entering credentials. The form includes fields for Username, Password, or Admin Access Code for administrator. Users can recover forgotten passwords through the "Forgot Password?" link. The interface implements role-based access control to distinguish between administrators and staff members.

Figure 34: Admin Dashboard Page

The main dashboard displays key system metrics including Total Products, Total Batches, Critical Alerts, and Warnings through statistic cards at the top. The primary section shows "Upcoming Expiry - Priority Action Required" with a table listing products sorted by FEFO method. Each batch displays

Product name, Batch Number, Quantity, Expiry Date, Days Left, and Status. Color-coded badges indicate urgency: red for critical status, yellow for warnings, and green for safe items.

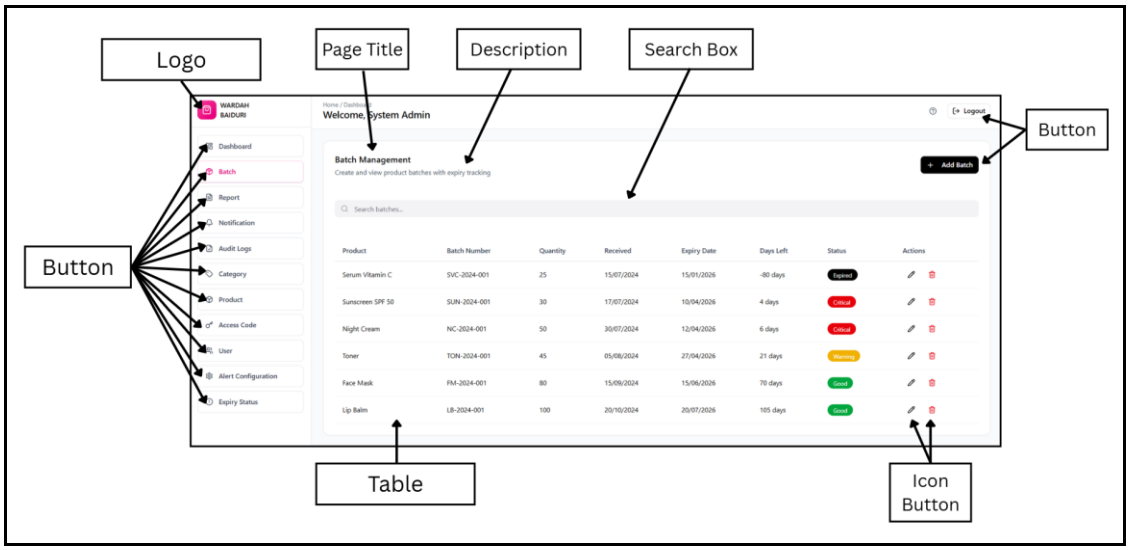


Figure 35: Batch Management Page

This interface shows all product batches with expiry tracking information. The table displays Product name, Batch Number, Quantity, Received date, Expiry Date, Days Left, and Status with color-coded indicators. Administrators can add new categories using the "+ Add Batch" button, and edit or delete existing batch using the icon buttons in the Actions column.

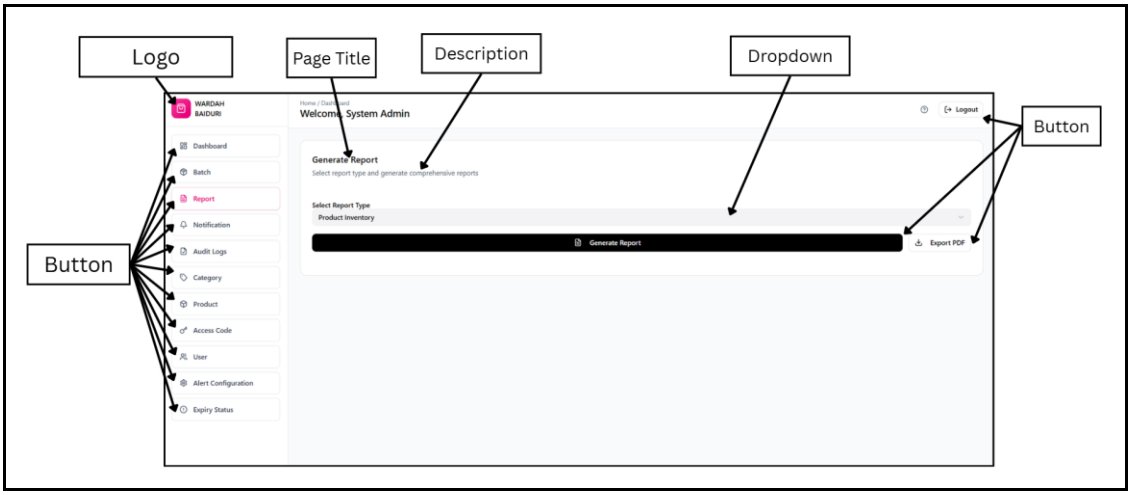


Figure 36: Reports Page

This page allows administrators to generate various system reports. Users select a report type from the dropdown menu and click "Generate Report" to create the document. The "Export PDF" button enables downloading reports for external use.

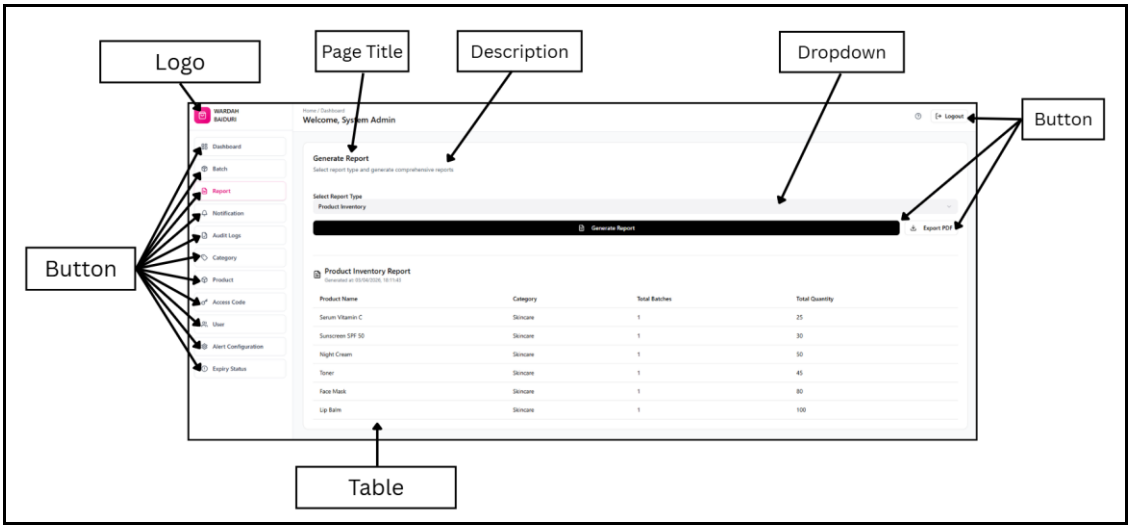


Figure 37: Reports Page for Product Inventory

This shows a generated Product Inventory Report which contains timestamp. The report shows Product name, Category, Total Batches and Total quantity of all products in the system. For auditing or review purposes, administrators can also export the report as PDF.

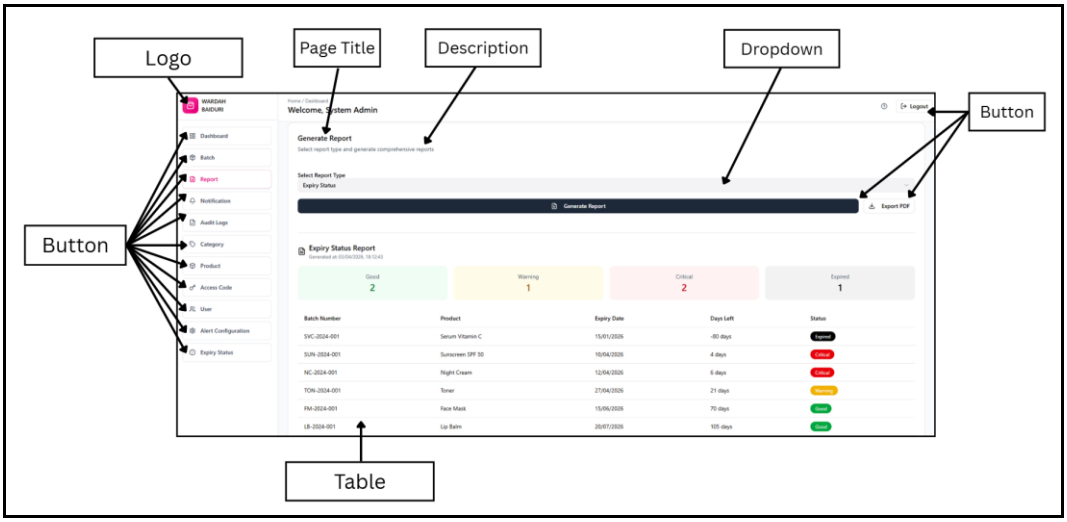


Figure 38: Reports Page for Expiry Status

This shows a generated Expiry Status Report displaying all product batches with their current expiry condition. The table in the report contains Product name, Batch Number, Expiry Date, Days Left, and Status with coloured badges for Expired, Critical, Warning and Good conditions. This report can be exported to PDF by administrators for inventory review.

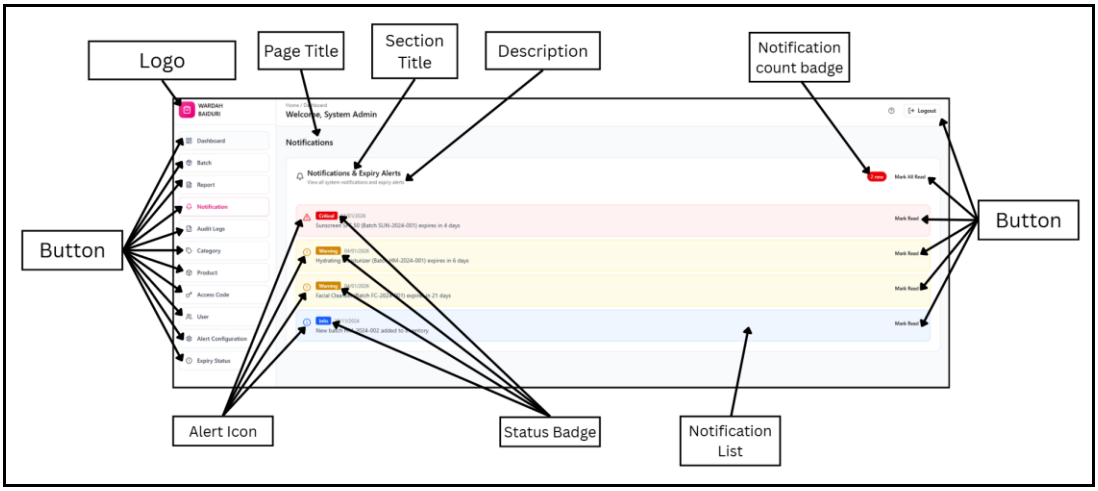


Figure 39: Notifications Page

This page shows all the expiry alert notifications and system updates to the admin. Critical alerts show up in red, warnings in yellow and information notices in blue. With each notification, the product, batch number, days remaining until expiry and timestamp can be found. A notification count badge shows how many alerts are unread. Admin can mark individual notification as read or use “Mark All Read” to Mark all notifications.

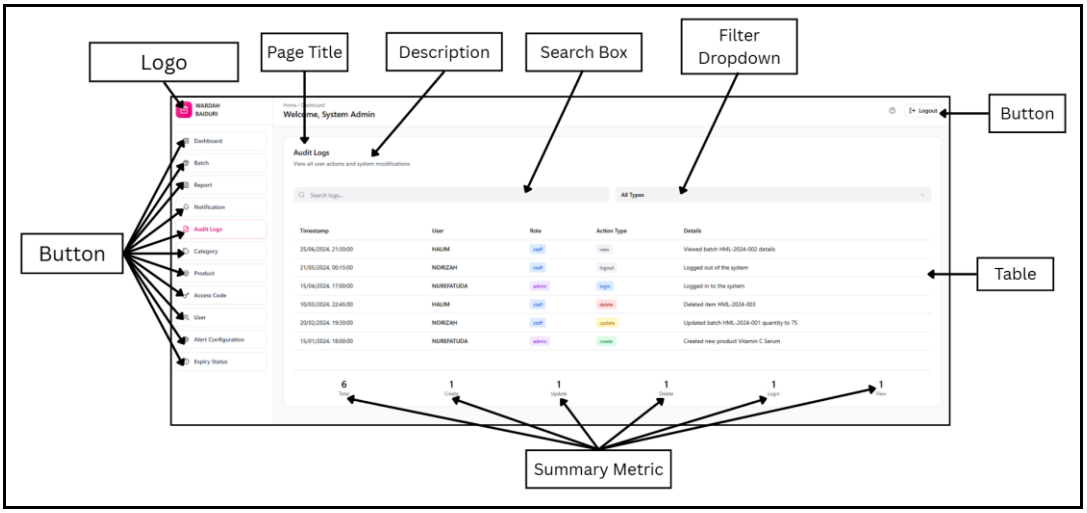


Figure 40: Audit Logs Page

This interface records all user actions and system changes for security monitoring. Each activity is shown with Timestamp, User, Role Action Type, Details. The action types, which appear as color-coded badges, are view, logout, login, delete, update and create. A filter dropdown to filter by action type. At the bottom, some summary statistics give fast information about total system actions.

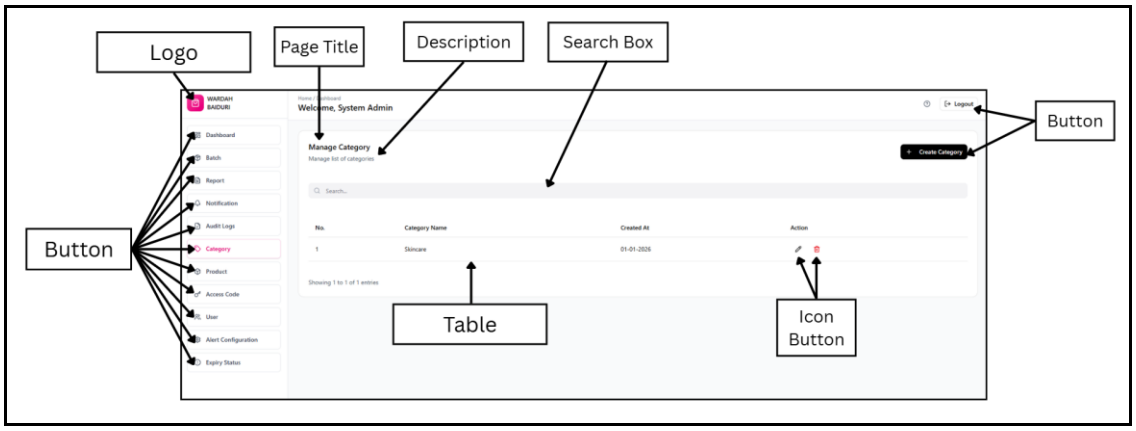


Figure 41: Category Management Page

This page handles product category that used to categorize inventory items in the system. For each entry the table displays Category Name and Action options. The administrators can add new categories through the "+ Add Category" button, and edit or delete existing categories using icon buttons in the Actions column.

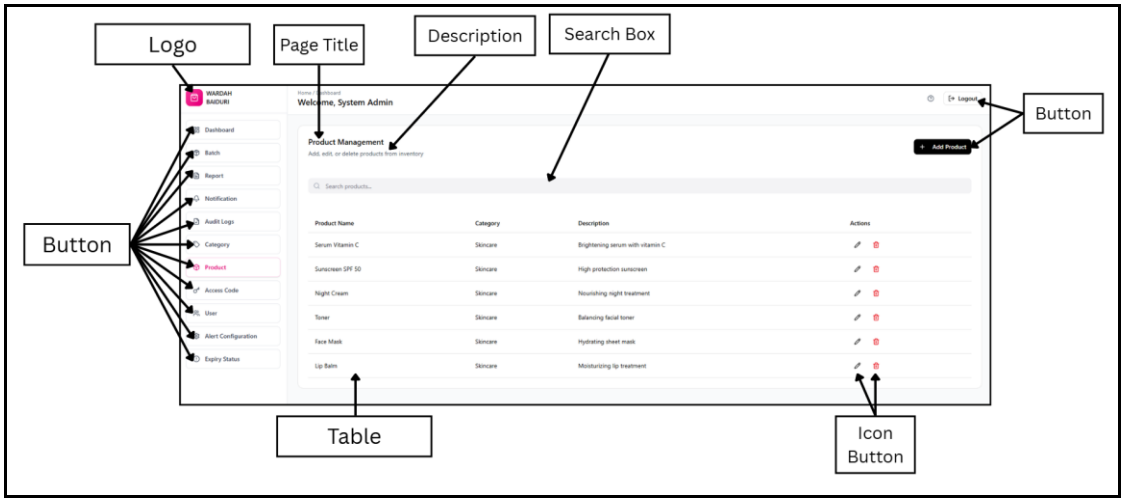


Figure 42: Product Management Page

This page displays all products in the inventory system. The administrators can search for specific products using the search bar they have, or add new products with the "+ Add Product" button. Each item has a Product Name, Category, Description. Actions column: Edit and delete icon buttons in this column enable administrators to edit or delete products from the system.

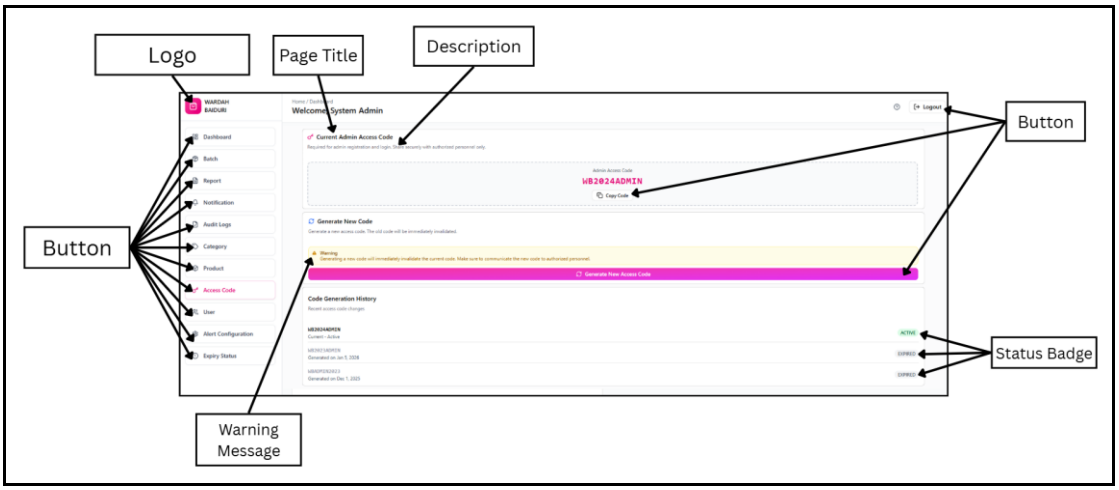


Figure 43: Admin Access Code Management Page

This page manages the secure access code required for admin login. The current active code is displayed with a copy function for easy sharing. Administrators can generate new codes, which immediately invalidates the previous code, with a warning message displayed to confirm the action. The Code Generation History section shows past codes with their status badges as Active or Expired.

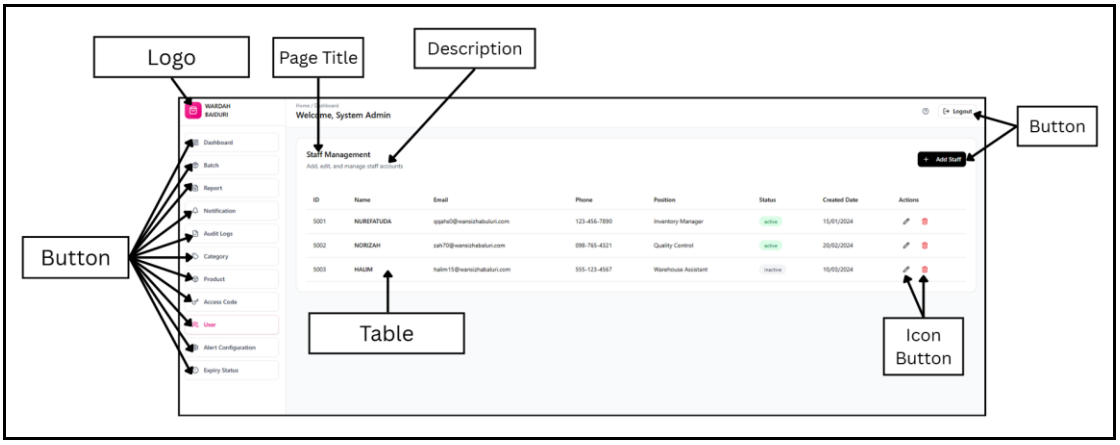


Figure 44: Staff Management Page

This page manages all staff user accounts in the system. The table displays staff ID, Name, Email, Phone, Position, Status, Created Date, and Actions. Administrators can add new staff members using the "+ Add Staff" button, edit or delete existing accounts, or deactivate users as needed. Status badges indicate whether accounts are active or inactive.

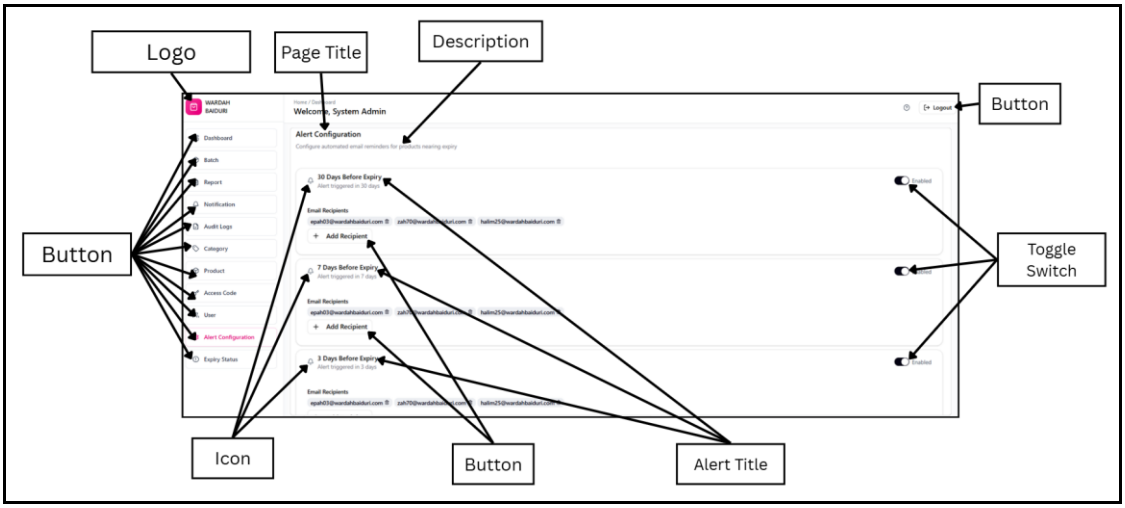


Figure 45: Alert Configuration Page

This page configures automated email notifications for products approaching expiry. Multiple alert intervals are available including 30 Days, 7 Days, and 3 Days Before Expiry. Each alert can be enabled or disabled using toggle switches on the right. Administrators can assign multiple email recipients for each alert interval and add new recipients as needed using the "+ Add Recipient" button.

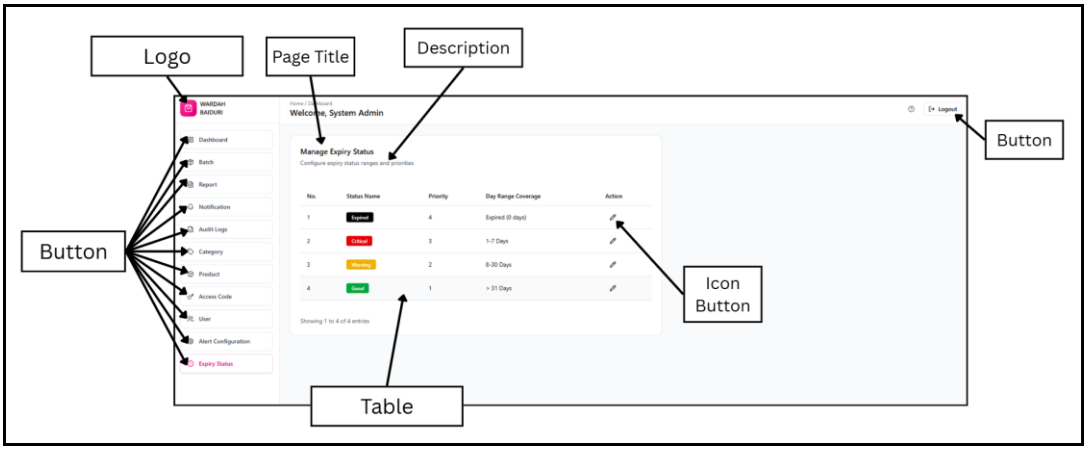


Figure 46: Expiry Status Configuration Page

This page allows administrators to configure the day range thresholds that determine the expiry status classification for each product batch. The table displays Status Name, Priority, Day Range Coverage, and Action for each status level including Expired, Critical, Warning, and Good. Administrators can edit the minimum and maximum day ranges for each status category using the icon button in the Actions column.

6.2.2 Staff Section

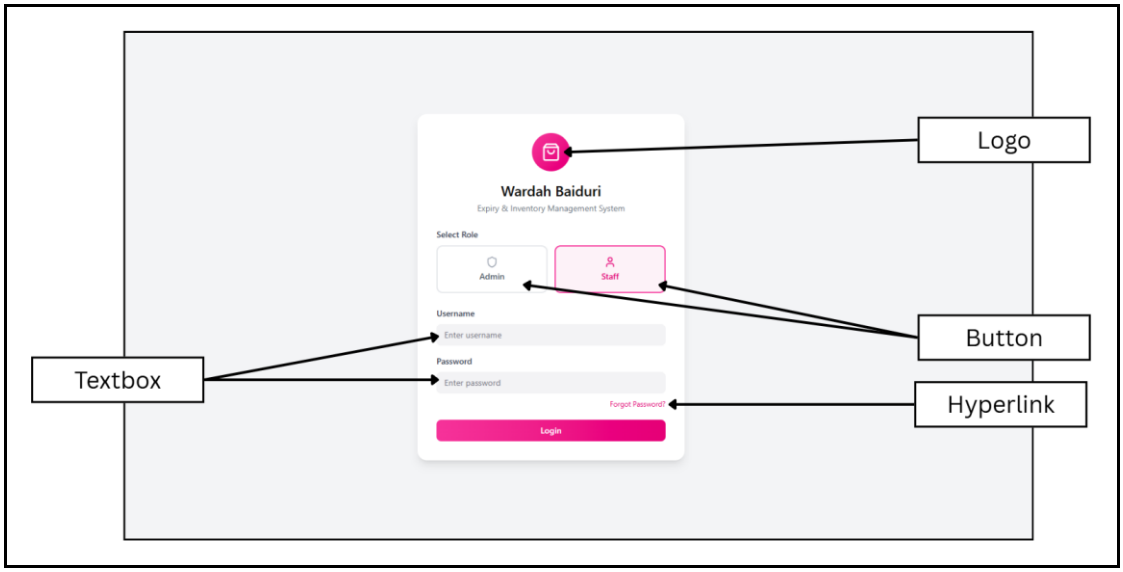


Figure 47: Staff Login Interface

Figure 47 shows the staff login page for the EXPI-STOCK system. Staff members select the "Staff" role and enter their username and password to access the system. Unlike the Admin login, the Staff login does not require an access code. A "Forgot Password?" link is provided for password recovery. After successful login, staff are directed to their dashboard.

Figure 48: Staff Dashboard Page

Figure 48 shows the staff dashboard with statistic cards at the top displaying Total Products, Good Status, Warning, and Critical items. The main section displays the "Upcoming Expiry - Priority Items" table showing Product name, Batch Number, Quantity, Expiry Date, Days Left, and Status. Items are color-coded: red for critical, yellow for warning, and green for safe stock to help staff quickly prioritize which products require immediate attention.

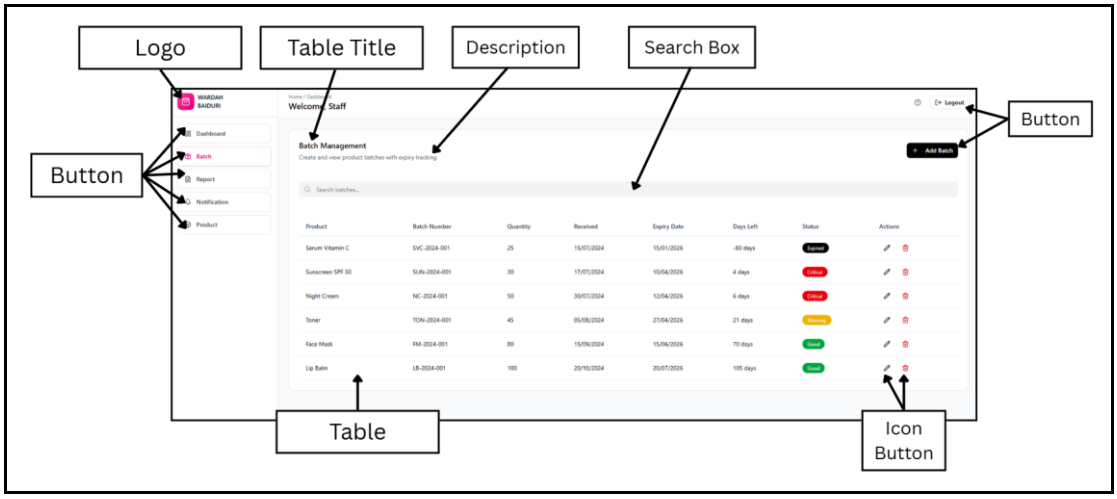


Figure 49: Staff Batch Management Page

Batch Management page and accessible to staff members as shown in Figure 49 It contains Product name, Batch Number, Quantity, Received date, Expiry Date days left and status indicator with color coding. The staff can add new categories using the "+ Add Batch" button and can edit or delete existing batch by using the icon buttons in Actions

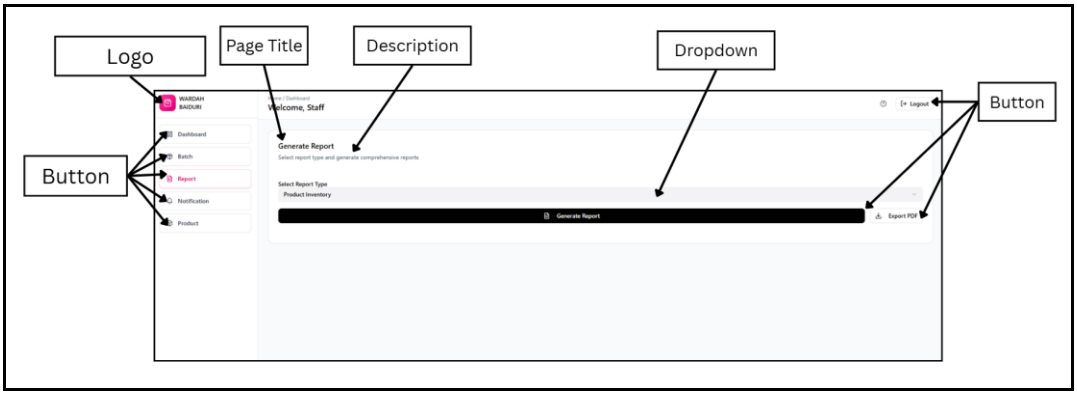


Figure 50: Reports Page

Figure 50 shows the Reports page for staff members. Staff can select a report type from the dropdown menu and click "Generate Report" to create the document. The "Export PDF" button enables downloading the generated report for external use or record-keeping purposes.

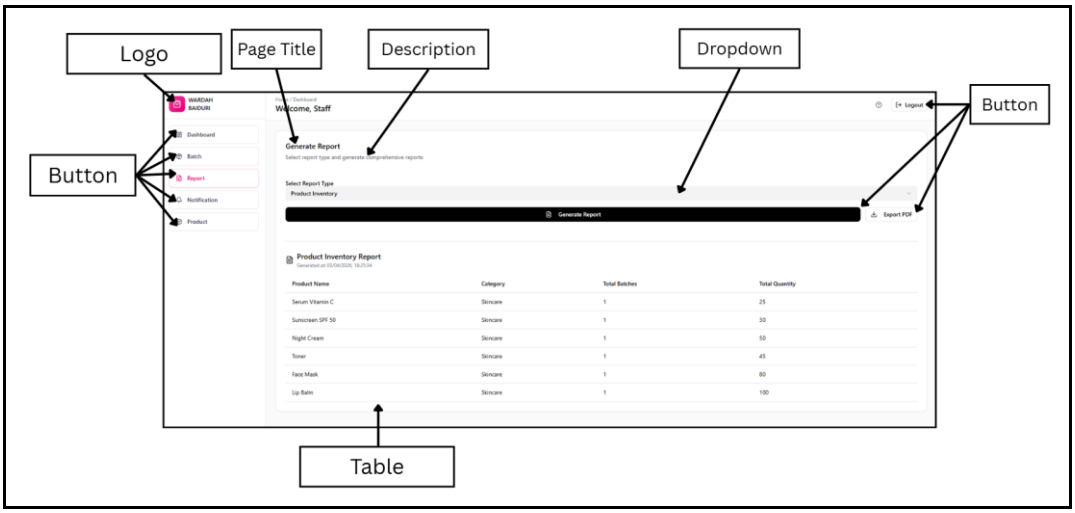


Figure 51: Reports Page for Product Inventory

The following is a generated Product Inventory Report on the Staff side in Figure 51. This report shows Product Name, Category, Total Batches and Total Quantity for all items across the system with a timestamp of when this report was generated. Staff member can export this report as PDF for the purpose of stock review or record.

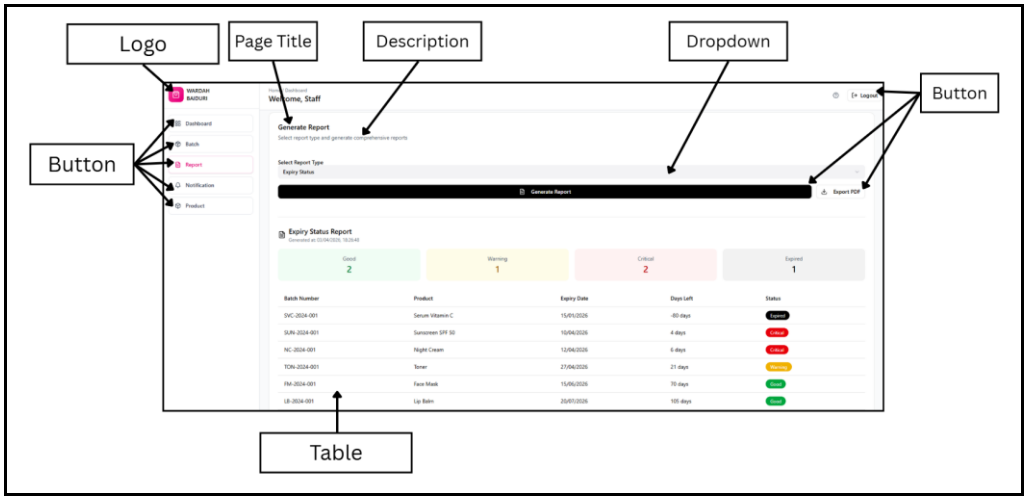


Figure 52: Reports Page for Expiry Status

Figure 52 shows a generated Expiry Status Report on the staff side. The report is structured in such a way that at the top, we see color coded summary cards with the count of Expired, Critical, Warning and Good batches followed by detailed table describing each batch along with Product name, expiry date, days to expiry and status. This report can be exported as PDF by staff to assist in making inventory decisions.

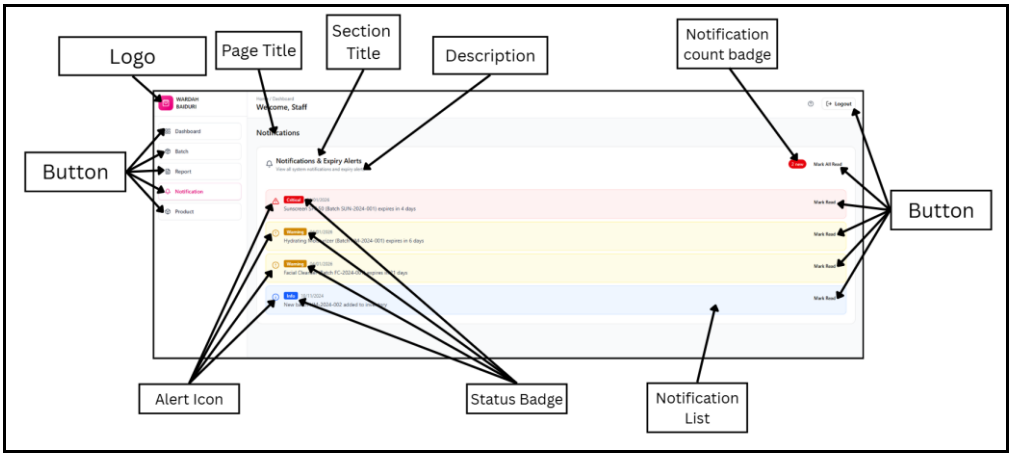


Figure 53: Staff Notifications Page

The staff's view is seen in the Notifications page all expiry camouflage and also system updates as shown by figure 53. Critical alerts show up in red, warnings show up in yellow and informational notices appear in blue. The notifications entry segments the product and batch number, expiry days left, timestamp Per notification along with a status badge. Staff can also mark individual notifications as read with a “Mark Read” button or mark them all at once using the “Mark All Read” option.

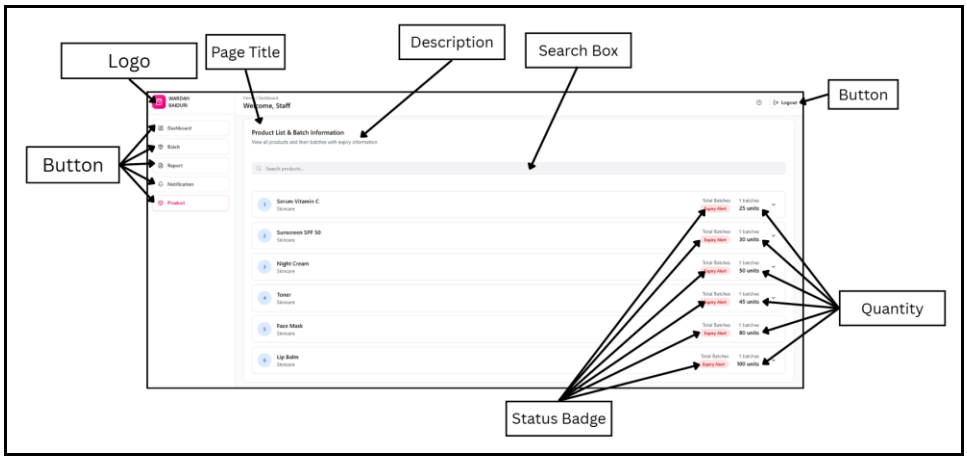


Figure 54: Product Page

Figure 54 presents the Product Catalog page where staff can view all products and their associated batch information. The search bar allows quick product lookup by name. Each product entry displays the product name, category, and quantity details. Products with approaching expiry dates are highlighted with color-coded status badges, allowing staff to quickly identify items that require priority action.

6.3 Database Design

This subsection shows the database design of the EXPI-STOCK system and provides a brief description of each table used to store and manage inventory data. This database was pulled and created using normalisation techniques in order to keep the data intact, avoid redundancy of entering same data over the time as well as also helps generate real-time queries for expiry checking and alerts. Each table is documented with a data dictionary that describes the data items, their data types, formats and example values.

6.3.1 Data Dictionary

A data dictionary provides a structured definition of the attributes within a database table. It outlines essential details such as data item names, data types, formats, descriptions, and examples for each attribute in the EXPI-STOCK Business Inventory Management System.

6.3.1.1 User Table

Table 9: Data Dictionary Users Table

Table Name	Data Item	Data Type	Data Format	Description	Example
users	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each user record in the database	1
	uuid	String	char (36)	Universally unique identifier (UUID) automatically generated for each registered user	ed5690ee-cb31-44c8-8d83-b30beed57d23
	name	String	Text	Full name of the user as registered in the system	System Admin
	email	String	Text	Email address of the user for authentication and notification purposes	admin@expistock.com
	email_verified_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the user email was verified. NULL if not yet verified	NULL

	phone_number	String	Text	Contact phone number of the user in Malaysian format	0123456789
	position	String	Text	Job position or role title of the user within the organization	System Administrator
	username	String	Text	Unique username chosen by user during registration for system login	admin
	password	String	Text	Hashed password using bcrypt algorithm for secure authentication	$2y$12$ABC...
	remember_token	String	Text	Token used to remember authenticated sessions. NULL if not active	NULL
	is_active	Boolean	tinyint(1)	Status indicating whether the user account is currently active (1) or deactivated (0)	1
	last_login_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time of the user's most recent login to the system	08-04-2026 03:57:18
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the user account was created in the system	29-03-2026 13:36:43
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the user record was last modified	08-04-2026 03:57:18

The users table stores information about every user on the system, including administrators and staff. The id field is an auto-incremented integer serving as a primary key that guarantees the uniqueness, while uuid such as ed5690ee-cb31-44c8-8d83-b30beed57d23 is a globally unique identifier. For example, the name field contains a full name like System Admin and email such as admin@expistock.com which is used for authentication & notifications. Text Entry (phone_number and position): phone_number stores contact numbers in Malaysian format and position store the user job title like System Administrator. Username is a unique key used for login, while password stores the bcrypt hashed password. The is_active field acts as a boolean indicator of whether the account is active (1) or deactivated (0). last_login_at tracks the last time

a user accessed the system, and audit fields store timestamps for when records were created and updated.

6.3.1.2 Roles Table

Table 10: Data Dictionary Roles Table

Table Name	Data Item	Data Type	Data Format	Description	Example
roles	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each role record	1
	name	String	varchar (255)	Name of the role assigned to users in the system. Values: admin or staff	admin
	guard_name	String	varchar (255)	Authentication guard name used by Laravel Spatie permission package	web
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the role was created in the system	29-03-2026 13:36:42
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the role record was last modified	29-03-2026 13:36:42

The roles table defines the access control roles within the system using Laravel Spatie Permission package. The unique identifier is the primary key id for each role. In the name field, you specify the role using one of two predefined values, admin for an Administrator user and staff for a Staff user, to allow access to various system feature and functions. The guard_name field refers to web which is the authentication guard Laravel uses. Audit fields include created_at and updated_at which timestamp when the role is created or modified.

6.3.1.3 Model Has Roles Table

Table 11: Data Dictionary Model Has Roles Table

Table Name	Data Item	Data Type	Data Format	Description	Example
model_has_roles	role_id	Integer	Numeric	Foreign key referencing the id in roles table, indicating which role is assigned	1
	model_type	String	varchar (255)	The model class that the role is assigned to	App\Models\ User
	model_id	Integer	Numeric	Foreign key referencing the id of the user the role is assigned to	1

The model_has_roles table, a pivot table managed by Laravel Spatie Permission package which is the one that connects users with their assigned roles. The role_id field is a foreign key that references the id of roles table. model_type stores a full class name of model like App\Models\User and model_id is specific user id. This table allows for dynamic assignment of roles at a user level, as well role-based access control in the entire system.

6.3.1.4 Category Table

Table 12: Data Dictionary Category Table

Table Name	Data Item	Data Type	Data Format	Description	Example
category	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each category record	1
	uuid	String	char(36)	Universally unique identifier automatically generated for each category	28f6275c-70a7-4dc3-aa20-2148024977eb
	name	String	varchar(255)	Name of the product category for	Skincare

				organizational purposes	
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the category was created in the system	29-03-2026 13:41:51
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the category record was last modified	29-03-2026 13:41:51

This table includes all of the product categories used to classify products in EXPI-STOCK. id the primary key which is an auto increment integer uuid like 28f6275c-70a7-4dc3-aa20-2148024977eb acts as the globally unique identifier specific for each category. The name field holds the category such as Skincare, Supplement, Herbal, Self-Care, Health Food, Health Drink, Cosmetic or Hair Care. Audit fields are created_at and updated_at timestamping the category when it has been created and last modified.

6.3.1.5 Product Table

Table 13: Data Dictionary Product Table

Table Name	Data Item	Data Type	Data Format	Description	Example
product	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each product record	1
	uuid	String	char (36)	Universally unique identifier automatically generated for each product	ece409cb-465c-488f-9dae-a34a3461f977
	category_id	Integer	Numeric	Foreign key referencing the id in category table for product classification	1
	name	String	varchar (255)	Name of the health and beauty product	Apple Fiber Vasia
	brand	String	varchar (255)	Brand name or manufacturer of the product	Vasia
	retail_price	Decimal	decimal (10,2)	Standard selling price of the product in Malaysian Ringgit (MYR)	55.00
	supplier_na	String	varchar	Name of the supplier or	Vasia

	me		(255)	distributor providing the product	Resources Sdn Bhd
	supplier_contact	String	varchar (255)	Contact information including phone number or email of the supplier	03-7890 1122
	is_active	Boolean	tinyint(1)	Status indicating whether the product is currently active (1) or discontinued (0)	1
	description	Text	Text	Detailed description of the product including key features and benefits	Natural apple fiber for digestive health
	image	String	Text	File path to the product image stored in the server storage	products/image.jpg
	created_by	Integer	Numeric	Foreign key referencing the user id of the administrator who added the product	1
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the product was added to the system	29-03-2026 16:48:33
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the product record was last modified	29-03-2026 16:48:33

The product table stores the master list of all products in EXPI-STOCK inventory This example uses the primary key id to uniquely identify each product, and uuid as a globally unique identifier. Field category_id is a foreign key, which links product with its category. Besides, product information contains several fields: name which keeps the name of the reference for eventual commercial use example Apple Fiber Vasia, brand that identifies the manufacturer like Vasia, and retail_price that stores the normal selling price e.g: 55.00 MYR currency decimal point format with two fractions. Supplier information for procurement is recorded via supplier_name as Vasia Resources Sdn Bhd and supplier_contact such as 03-7890 1122. The description field contains the detailed product information and image stores only the path of file related to the product in server storage. The is_active boolean field specifies whether the product is active (1) or discontinued (0). There is a created_by field referring to the administrator that added the product and audit fields like created_at or updated_at timestamps.

6.3.1.6 Batch Table

Table 14: Data Dictionary Batch Table

Table Name	Data Item	Data Type	Data Format	Description	Example
batch	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each batch record	1
	uuid	String	char (36)	Universally unique identifier automatically generated for each batch	94b66e97-3643-4665-9836-0dc8ba47bbbb
	product_id	Integer	Numeric	Foreign key referencing the id in product table, linking the batch to its product	2
	batch_number	String	varchar (255)	Batch identification number created when the batch is added to the system	AFV-2026-001
	quantity	Integer	Numeric	Current quantity of products remaining in the batch	24
	receive_date	Date	DD-MM-YYYY	Date when the batch was received and added to inventory	10-01-2026
	expiry_date	Date	DD-MM-YYYY	Date when the product batch will expire and should not be sold	11-04-2026
	created_by	Integer	Numeric	Foreign key referencing the user id of the person who created the batch record	1
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the batch record was created in the system	29-03-2026 17:15:00
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the batch record was last modified	04-04-2026 01:45:00

The expiry tracking system, is based on the batch table which records individual product batches along with their corresponding expiration dates and quantities. The primary key id is unique for each batch and uuid field is a globally unique identifier. The product_id plays the role of a foreign key that connects the tables, which is intended to allow multiple batches of the same product, with distinct expiry dates. The batch_number field stores the traceability quality code like AFV-2026-001. The quantity field shows the number of items that are currently available (sales/stock adjustment) Date fields include receive_date which denotes when the batch was entered into inventory like 10-01-2026 and expiry_date like 11-04-2026 which indicates when product is unsellable. The used_by field refers to the user who created the batch record and audit fields contain the created_at and updated_at timestamps for record management.

6.1.1.7 Expiry Status Table

Table 15: Data Dictionary Expiry Status Table

Table Name	Data Item	Data Type	Data Format	Description	Example
expiry_ status	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each expiry status record	9
	uuid	String	char (36)	Universally unique identifier automatically generated for each expiry status	a175a7a4-9742-4608-95db-01524f5f7d2b
	name	String	varchar (255)	Name of the expiry status category. Values: Expired, Critical, Warning, Good	Expired
	min_day	Integer	Numeric	Minimum number of days until expiry for this status category	-9999
	max_day	Integer	Numeric	Maximum number of days until expiry for this status category	0
	priority	Integer	Numeric	Priority order for displaying expiry status where 1 is the highest priority	1

	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the expiry status record was created in the system	04-04-2026 01:42:39
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the expiry status record was last modified	05-04-2026 09:08:31

The expiry_status table holds configuration data such as range of days to consider the products as expired. Each status record is uniquely identified by its primary key id while uuid acts as a global unique identifier. The name field establishes the status category, with values: Expired (overdue), Critical (near due date), Warning (within assertation date range) and Good. The min_day and max_day fields establish the range of days until expiry for each disposition category, eg Expired uses -9999 to 0 days, Critical 1 to 3 days, Warning 4 to 7 days and Good 8 to 9999. This gives you control over the order which each API will be displayed – which is determined by the priority field where 1 is highest. The Manage Expiry Status function allows administrators to adjust how many days a document covers. Audit fields like created_at and updated_at timestamps.

6.3.1.8 Notification Table

Table 16: Data Dictionary Notification Table

Table Name	Data Item	Data Type	Data Format	Description	Example
notification	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each notification record	1
	uuid	String	char (36)	Universally unique identifier automatically generated for each notification	58820a1a-421a-4c1f-8a2c-720f202da04a
	user_id	Integer	Numeric	Foreign key referencing the id in users table indicating which user the notification belongs to	1
	status	String	varchar (255)	Current delivery status of the notification	sent
	description	Text	Text	Detailed content or message of the notification	Product expiring in 30

				sent to the user	days
	is_read	Boolean	tinyint(1)	Status indicating whether the notification has been viewed (1) or unread (0)	0
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the notification record was created in the system	29-03-2026 13:45:37
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the notification record was last modified	29-03-2026 13:45:37

This table has records of all alert notifications that have been generated by the system for products close to their expiry dates. The id is a primary key and the uuid is a unique identifier for every notification. The user_id field is a foreign key that relates the notification to the particular user it belongs to. The status field stores the current delivery status of the notification, while description holds the content of detailed notification message. The is_read field will be a boolean indicating whether the notification has been read in the dashboard (1) or not yet viewed (0). Audit fields created_at and updated_at timestamps to keep track of record lifecycle.

6.3.1.9 Alert Configuration Table

Table 17: Data Dictionary Alert Configuration Table

Table Name	Data Item	Data Type	Data Format	Description	Example
alert_configuration	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each alert configuration record	1
	uuid	String	char(36)	Universally unique identifier automatically generated for each alert configuration	848295a4-5ad5-44a2-9bef-b3c0cae7d7f6
	user_id	Integer	Numeric	Foreign key referencing the id in users table indicating which administrator configured	1

				the alert	
	day_left	Enum	enum('30','7', '3')	Number of days before expiry when the alert notification should be triggered. Predefined values: 30, 7, or 3	30
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the alert configuration was created in the system	29-03-2026 13:45:37
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the alert configuration record was last modified	29-03-2026 13:45:37

The alert_configuration table holds the automated expiry notifications settings specified by the administrator. The primary key id uniquely identifies each configuration record and uuid is a globally unique identifier The user_id field is a foreign key that comes from the administrator who configured the alert. day_left is an enumerated type which specifies when notifications should be sent. The predefined values are 30, 7 and 3 days before expiration The Alert Configuration function allows administrators to create or remove alert configurations. Audit fields for created_at and updated_at timestamps recording when the configuration was created and last modified.

6.3.1.10 Audit Logs Table

Table 18: Data Dictionary Audit Logs Table

Table Name	Data Item	Data Type	Data Format	Description	Example
audit_logs	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each audit log entry	1
	uuid	String	char (36)	Universally unique identifier automatically generated for each audit log record	58820a1a-421a-4c1f-8a2c-720f202da04a
	user_id	Integer	Numeric	Foreign key referencing the id in users table who performed the action	1
	action	String	varchar (255)	Type of action performed by the user. Values: Login, Create, Update, Delete	Login

	description	String	varchar (255)	Detailed description of the action performed including the affected record	Login to system
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the action was performed and logged in the system	29-03-2026 13:37:15
	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the audit log record was last modified	29-03-2026 13:37:15

The audit_logs table keeps track of all user actions and system changes for auditing purposes. Id - the primary key which uniquely identifies each audit log and uuid, unique globally as an identifier. This is a foreign key in the user_id field referencing to the actual user taking this action. The action field classifies the operation in values like Login, Create, Update and Delete. The description field explained what action is getting performed. For example: Login to system or Create new product name Apple Fiber Vasia (what record got affected). The audit fields capture created_at and updated_at timestamps, noting exactly when each event took place.

6.3.1.11 Access Code Table

Table 19: Data Dictionary Access Code Table

Table Name	Data Item	Data Type	Data Format	Description	Example
access_code	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each access code record	1
	user_id	Integer	Numeric	Foreign key referencing the id in users table of the administrator who generated the code	1
	code	String	varchar (8)	Secure randomly generated alphanumeric code required for staff registration into the system	XU4EI8IY
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the access code was generated in the system	29-03-2026 13:41:06

	updated_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the access code record was last modified	08-04-2026 03:50:01

The access_code table handles secure codes that need to be used for staff account registrations making sure only authorized personnel can register into the system. The access code primary key id is unique for each access code record. The user_id column is a foreign key to the administrator who created the code. The code field contains the generated alphanumeric code, such as XU4EI8IY that then needs to be submitted when adding your staff. Only one access code is valid at any time, as when a new access code is generated the previous one is overridden. These audits fields include created_at and updated_at timestamps that track when the code was generated and last modified.

6.3.1.12 Password Reset Tokens Table

Table 20: Data Dictionary Password Reset Tokens Table

Table Name	Data Item	Data Type	Data Format	Description	Example
password_reset_tokens	email	String	varchar (255)	Email address of the user who requested the password reset, used as the primary identifier	admin@expistock.com
	token	String	varchar (255)	Secure randomly generated token included in the password reset link sent via email	TOKEN123ABC
	created_at	Timestamp	DD-MM-YYYY HH:MM:SS	Date and time when the password reset request was initiated by the user	20-01-2026 10:00:00

This table handles password recovery requests, enabling users who forget their passwords to securely reset them. The email field represents the primary key that connects the reset request with the specific user account. An alphanumeric token like TOKEN123ABC that is included in the password reset link sent out in an email which is secure and randomly generated. The created_at field shows when the reset request was made. This means that when a user opts to change their password, the system creates a new token and an email with the link for password reset is sent to the stored mail ID.

6.3.1.13 Sessions Table

Table 21: Data Dictionary Sessions Table

Table Name	Data Item	Data Type	Data Format	Description	Example
sessions	id	String	varchar (255)	Unique session identifier string automatically generated for each active user session	KlrmWUWOVb...
	user_id	Integer	Numeric	Foreign key referencing the id in users table. NULL if the session is unauthenticated	1
	ip_address	String	varchar (45)	Internet Protocol (IP) address from which the session was initiated for security tracking	127.0.0.1
	user_agent	Text	Text	Browser and operating system information of the user who initiated the session	Chrome/146 on Windows 10
	payload	Text	longtext	Serialized session data containing authentication state and other session variables	YTo0Ontzojo...
	last_activity	Integer	Numeric	Unix timestamp recording the last time the session was active in the system	1775620757

The sessions table manages active user sessions in the EXPI-STOCK system using Laravel session management. The id field is a unique string identifier automatically generated for each active session. The user_id field is a foreign key referencing the authenticated user, with NULL value for unauthenticated sessions. The ip_address field records the Internet Protocol (IP) address from which the session was initiated for security tracking and user_agent captures the browser and operating system information. The payload field stores serialized session data containing authentication state and other session variables. The last_activity field is a Unix timestamp recording the last time the session was active, used to manage session expiration.

6.3.1.14 Migrations Table

Table 22: Data Dictionary Migrations Table

Table Name	Data Item	Data Type	Data Format	Description	Example
migrations	id	Integer	Numeric	Auto-incremented primary key that uniquely identifies each migration record	1
	migration	String	varchar (255)	Name of the Laravel migration file that was executed to build the database structure	0001_01_01_000000_create_users_table
	batch	Integer	Numeric	Batch number grouping migrations that were run together in the same execution	1

This stores the active user sessions for our EXPI-STOCK system using Laravel session management. Each active session is assigned a unique random string identifier (the id field) by the server. If the session is unauthenticated then it stores NULL in user_id field, otherwise it is stored as a foreign key of authenticated user. The ip_address field logs the Internet Protocol (IP) address from which the session was started for security monitoring and user_agent contains browser and operating system details. Specifically, the payload field holds serialized session data on the authenticated state, among other session variables. The last_activity field is a Unix timestamp that holds the time of the most recent session, used for determining whether to expire the session or not.

6.3.2 Data Flow Diagram (DFD)

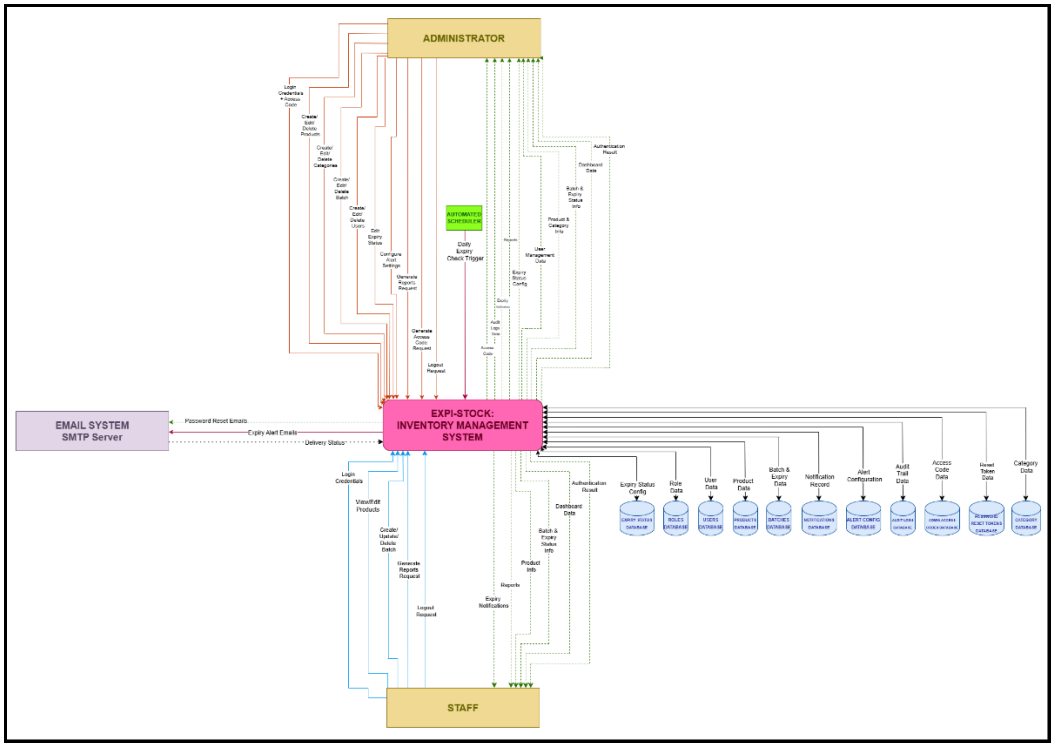


Figure 55: EXPI-STOCK Data Flow Diagram (DFD)

The above Data Flow Diagram is a high-level overview of how data flows in the EXPI-STOCK Business Inventory Management System, illustrating the relationships between users, processes and data stores. This system is the central entity for a product inventory, batch and expiry tracking and notifications. This diagram shows three main external entities which are Administrator, Staff Members, and Email System (SMTP Server) along with the Automated Scheduler that interacts with the system in order to perform a variety of inventory management tasks and expiry tracking.

a) External Entities

- **Administrator**

  The Administrator serves as the primary external entity with comprehensive system access and management capabilities. Administrators interact with the EXPI-STOCK system by providing login credentials combined with a secure access code for authentication, creating, editing, and deleting products in the inventory database, creating, editing, and deleting product categories, creating, editing, and deleting product batches with expiry information, creating, editing, and deleting user accounts, editing expiry status day range configurations, configuring alert settings for notification thresholds, generating reports for inventory

analytics, and generating secure access codes for administrator. In return, the system provides authentication results, dashboard data displaying key metrics, batch and expiry status information, product and category information, user management data, expiry status configuration, reports, expiry notifications for approaching or expired products, audit logs data for security monitoring, and newly generated access codes.

- **Staff Members**

  Staff Members represent the operational users who utilise the system for daily inventory tasks and monitoring. Staff interact with the EXPI-STOCK system by providing login credentials, viewing and editing products to check inventory status, creating, updating, and deleting batch records, generating reports, and requesting logout. The system responds by providing authentication results, dashboard data, batch and expiry status information, product information, reports, and expiry notifications alerting about approaching expiration dates.

- **Email System (SMTP Server)**

  The Email System represents the external SMTP server used for sending automated notifications and communications. The EXPI-STOCK system sends expiry alert emails when products approach expiration dates based on configured intervals, and password reset emails containing secure tokens for credential recovery. The Email System provides delivery status feedback confirming whether emails were successfully sent or failed, enabling the system to track notification delivery.

- **Automated Scheduler**

  The Automated Scheduler represents the background process that runs continuously without manual intervention. The scheduler triggers a daily expiry check to evaluate all active batches against expiration thresholds. This automated process queries the Batches Database, calculates days until expiry, identifies batches meeting alert criteria, generates notification records, and triggers email notifications through the Email System, ensuring continuous monitoring and timely alert generation.

b) Central System Process

- **EXPI-STOCK Inventory Management System**

The EXPI-STOCK is the core processor that makes all interaction between external sources and data stores. It ensures user authentication and authorisation that verifies credentials against registered data, enforcing role-based access control as needed. It handles product and category management for CRUD operations to store data in the Products and Category databases. Batch management is tracked towards expiry by keeping detailed records with automatic status assignment. Automated notification generation is pushed through evaluation of the batch's status daily followed by creating a record in Notification Records based on Alert Configuration settings. Delivery of alerts is handled via email notifications composed according to SMTP Server parameters All user actions are captured into Audit Logs Database providing an audit trail.Comprehensive secure secret key storage comes with administrator registration plus processing password resets using tokens sent via email.

c)Data Stores

- **Users Database**

The Users Database stores information about all registered users including user credentials, role, phone number, active status, and timestamps. Data flows include writing user data during registration and account creation, reading credentials during authentication, retrieving user information for user management displays, and updating records for account modifications.

- **Roles Database**

The Roles Database stores the role definitions used for role-based access control within the system. Data flows include the system reading role data to enforce access permissions for Administrator and Staff users throughout all system operations.

- **Products Database**

The Products Database maintains the master catalog of all inventory products including product name, category, brand, description, retail price, supplier information, product

image, and active status. Data flows include administrators and staff creating new products, the system reading product information for displays, and updating or deleting product records as required.

- **Category Database**

  The Category Database stores product category information used to organise and classify inventory products. Data flows include administrators creating, editing, and deleting categories, and the system reading category data when displaying.

- **Batches Database**

  The Batches Database forms the core of expiry tracking, storing batch records with product references, batch numbers, quantities, receive date, expiry date, and timestamps. Data flows include creating batch records, reading batch information for displays, updating quantities and batch details, and the Automated Scheduler reading batch data during daily expiry checks.

- **Expiry Status Database**

  The Expiry Status Database stores the classification thresholds that define the day ranges for each expiry status level such as safe, warning, critical, and expired. Data flows include administrators editing the day range configuration, and the system reading status thresholds to classify batches accordingly.

- **Notifications Database**

  The Notifications Database tracks all automated alerts generated for products approaching expiry dates. Data flows include creating notification records when batches meet alert thresholds, reading notification data for dashboard displays, and updating read status when users acknowledge notifications.

- **Alert Configuration Database**

  The Alert Config Database stores settings for automated expiry notifications including alert intervals and recipient configurations. Data flows include administrators configuring alert

settings, and the system reading configuration data during automated daily checks to determine when and how notifications should be sent.

- **Audit Logs Database**

  The Audit Logs Database maintains records of all significant user actions for security monitoring and compliance. Data flows include the system writing log entries for all authentication events, CRUD operations, and configuration changes, and administrators reading audit logs for security monitoring and activity tracking.

- **Admin Access Codes Database**

  The Admin Access Codes Database manages secure access codes required for administrator to login the system. Data flows include administrators generating new access codes, the system storing and validating codes during administrator account creation, and updating code status after use.

- **Password Reset Tokens Database**

  The Password Reset Tokens Database handles password recovery by storing secure reset tokens. Data flows include the system storing reset token data when a password reset is requested, and reading token records to validate reset requests before allowing password changes.

6.3.3 Entity Relational Diagram (ERD)

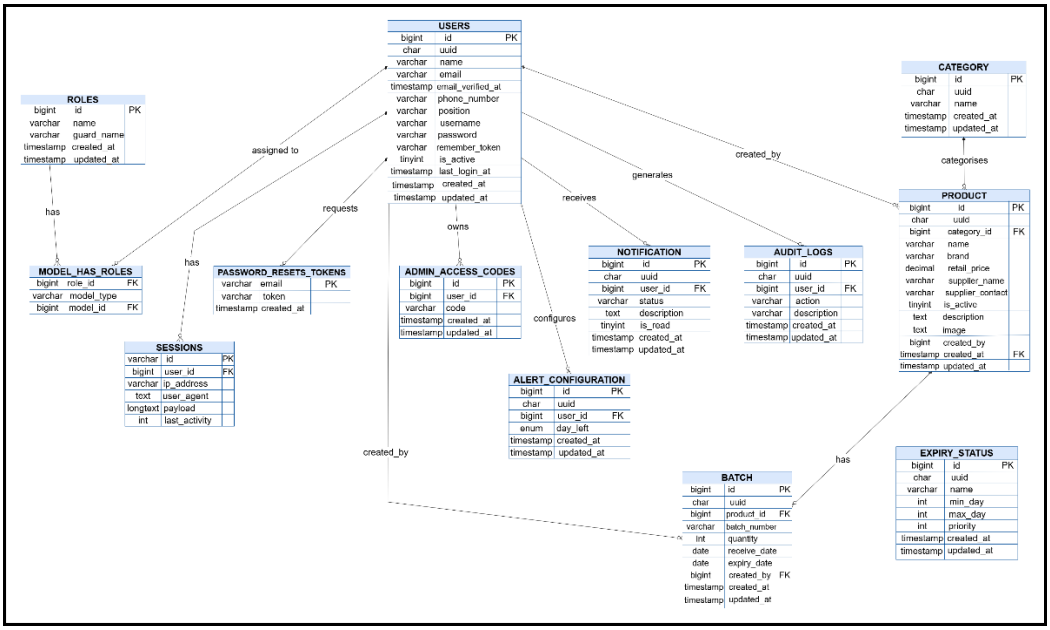


Figure 56: EXPI-STOCK Entity Relationship Diagram (ERD)

The above-drawn Entity Relationship Diagram (ERD) is the graphical representation of the database used in EXPI-STOCK Business Inventory Management System, which shows how different entities are associated with each other and their attributes. This system has ten main entity classes that together offer full tracking of inventory, expiry management, and notification automation. The ERD does not represent these database tables but there are three other tables generated by the Laravel framework (those for migrations, sessions and model_has_roles) that exist in the database to facilitate authentication, session management and role-based access control however they were excluded from the ERD as they manage internal workings of the framework itself and do not correlate with core business entities.

1. **Core Entities and Relationships**

- **USERS Entity**

  The USERS entity serves as the foundation of the system, storing essential information about all registered users including administrators and staff members. This entity contains attributes such as id as the primary key, uuid as a unique identifier, name, email, email_verified_at, phone_number, position, username, password, remember_token, is_active status, last_login_at, created_at, and updated_at timestamps. The USERS entity maintains one-to-many relationships with multiple other entities including PRODUCT, BATCH, ACCESS_CODE, ALERT_CONFIGURATION, NOTIFICATION, and

AUDIT_LOGS, acting as the central point for tracking user actions and ownership throughout the system.

- **ROLES Entity**

  The ROLES entity defines the access control levels available within the system. Attributes include id as the primary key, name identifying the role, guard_name specifying the authentication guard used, created_at, and updated_at timestamps. This entity is managed through the Spatie Laravel Permission package and maintains a many-to-many relationship with USERS via the model_has_roles pivot table, allowing flexible role assignment across the system.

- **CATEGORY Entity**

  The CATEGORY entity logically groups products together to allow for easier inventory management. Attributes are id (primary key), uuid, name, created_at, and updated_at timestamps. This entity maintains a one-to-many relationship with PRODUCT, which can be only assigned to one category multiple products.

- **PRODUCTS Entity**

  The PRODUCTS Maintains a master catalog of all health and at Beauty product in stock. Which include primary key id, uuid, foreign key category_id referencing CATEGORY table, name, brand, retail_price, supplier_name, supplier_contact, is_active status description image, created_by which is a foreign key that references USERS, created_at and updated_at timestamps. This creates a one-to-many relationship with BATCH, where each product can have many batches with different expiration dates.

- **BATCH Entity**

  The BATCH entity is at the heart of the expiry tracking, representing infivual product batches with their receive and expiry date. The entity includes a primary key of id, as well as properties for uuid, product_id (foreign key to PRODUCT), batch_number, quantity, receive_date, expiry_date and creator info created_by (foreign key to USERS) followed by timestamps created_at and updated_at. It makes up a vital aspect of the effective expiry monitoring feature in the system.

- **EXPIRY_STATUS Entity**

  The EXPIRY_STATUS entity defines the expiration condition classification thresholds. ID, UUID, EACH status label NAME MIN DAY AND MAX DAY RANGE FOR WHICH STATUS IT IS FROM THE TABLE ORDER WITH PRIORITY Timestamp created at timestamp updated. This entity has no foreign key relationship in the database it is just a reference configuration used by application logic whenever it needs to classify batches based on the remaining days of expiry.

2. Supporting Entities

- **ALERT_CONFIGURATION Entity**

  The ALERT_CONFIGURATION entity contains preferences, per user, for automated expiry notifications. It has the following attributes – id (primary key), uuid, user_id (foreign key linked to USERS), day_left (an enum value of 30, 7 or 3 days indicating when notification should be triggered) and created_at and updated_at timestamps. In this entity, there is a many to one relation with USERS.

- **AUDIT_LOGS Entity**

  The AUDIT_LOGS entity provides comprehensive tracking of all user actions and system modifications for security monitoring and compliance purposes. Key attributes include id as the primary key, uuid, user_id as a foreign key referencing USERS, action, description, created_at, and updated_at timestamps. This entity maintains a many-to-one relationship with USERS, ensuring every recorded action is traceable to a specific user.

- **ADMIN_ACCESS_CODES Entity**

  The ADMIN_ACCESS_CODES composite responsible for secure access codes that must be entered during administrator registration. key, foreign key (referencing USERS) user_id, code, created_at and updated_at timestamps. This has a many to one relationship with USERS, associating each code access with its creator.

- **PASSWORD_RESET_TOKENS Entity**

The PASSWORD_RESET_TOKENS entity manages requests for password recovery, allowing users to securely reset their passwords. It contains the following attributes: email (primary key), token (the secure reset token), created_at. This is linked to USERS through the email field that binds each reset token with an existing user account.

- **NOTIFICATION Entity**

The NOTIFICATION entity records all system notifications, including product expiry notifications, that were generated and delivered to users. columns: id primary key, uuid, user_id foreign key to USERS, status, description, is_read what if notification has been read (default 0), created_at and updated_at timestamps. This entity has a many-to-one relationship with USERS.

6.4 Flow of the System

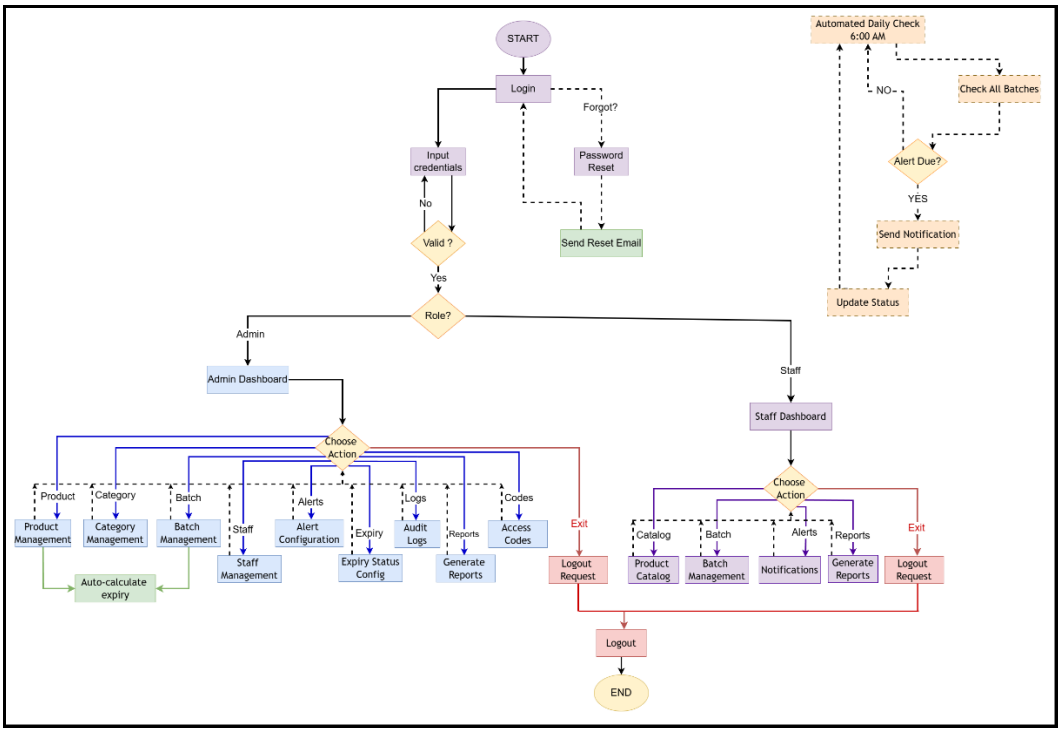


Figure 57: EXPI-STOCK Flow of the System Diagram

EXPI-STOCK is a web-based inventory and expiry tracking system which uses role for admin and staff members. The flow of the process starts from log in to user and splits into two different paths as per the

function done by the user along with a background automated process running continuously for checking expiry of inventory. Here is a breakdown explaining each part and process in the system.

a) Authentication Flow

Users start at the Login page were credentials are checked with Users Database. They need to re-enter valid credentials or click on an option that allows them to reset their password in case the entered password is incorrect, where a token will be delivered via secure email. After submitting the password reset email, it redirects users back to the Login page for authentication with their new credentials. If validated, the system then checks their roles and leads them to their respective dashboards as per given access.

b) Administrator Functions

The Admin Dashboard serves as the central control hub providing comprehensive system management capabilities. Through the Choose Action menu, administrators can access the following functions.

Through Product Management, administrators can create, edit, or delete products with complete details including name, category, brand, description, retail price, and supplier information. Category Management allows administrators to organise products by creating, editing, or deleting product categories to maintain a structured inventory catalog.

Batch Management is central to the system's expiry tracking capabilities, allowing administrators to create, edit, or delete product batches by entering batch number, quantity, receive date, and expiry date. The system automatically calculates days until expiry and assigns appropriate batch status along with urgency levels for priority management.

Staff Management allows administrators to add new staff accounts, edit existing user details, modify permissions, or deactivate accounts for security purposes. Alert Configuration enables administrators to set automated notification intervals at 30, 7, and 3 days before product expiry, configure recipient email addresses, and customise alert settings that are immediately applied to the notification system.

Expiry Status Config allows administrators to define and edit the day range thresholds that classify each batch into its corresponding expiry status level such as safe, warning, critical, or expired, ensuring the classification criteria align with business requirements.

Through Generate Reports, administrators can create analytical reports including Product Inventory Reports and Expiry Status Reports with export capabilities. The Audit Logs function provides comprehensive system activity records showing timestamps, user IDs, action types (CREATE, READ, UPDATE, DELETE, LOGIN, LOGOUT), affected entities, and modification details for security monitoring and compliance tracking. Access Codes management allows generation of secure registration codes required for creating new administrator accounts, with automatic deactivation of previous codes to maintain security.

After completing any action, the system returns the administrator to the Choose Action menu to select another function. When finished, the administrator can select Logout Request to terminate the session.

c) Staff Functions

The Staff Dashboard provides operational access to inventory management functions tailored for daily operations. Through the Choose Action menu, staff members can access the following functions.

The Product Catalog function allows staff to view all products with search and filter capabilities, displaying comprehensive information including batch details, quantities, and expiry dates, with products approaching expiry highlighted accordingly.

Batch Management allows staff to create, update, or delete batch records and maintain accurate inventory quantities as products are sold. The system automatically recalculates remaining quantities and batch status while logging all updates. Any batch changes also trigger the Auto-calculate expiry process, which automatically recalculates days until expiry and updates batch status accordingly.

Through Notifications, staff members can view all expiry alerts and system updates in chronological order, colour-coded by urgency with red badges for critical alerts, yellow badges for warnings, and blue badges for informational notices. Each notification displays product name, batch number, days until expiry, and timestamp, with options to mark notifications as read.

Generate Reports allows staff to produce inventory and expiry status reports for reference during daily operations. After completing any action, the system returns the staff member to the Choose Action menu to select another function. When finished, the staff member can select Logout Request to terminate the session.

d) Automated Daily Check

Operating independently from user interactions, the system maintains an automated daily check process that triggers at 6:00 AM every morning. This automated workflow begins by checking all batches, querying the Batches Database to retrieve all active batches and calculating days until expiry for each. The process then compares calculated days against configured alert thresholds of 30, 7, and 3 days before expiry.

When alert thresholds are met, the system sends notifications by creating notification records, retrieving recipient email addresses from Alert Configuration, composing personalised emails using configured templates with batch details, and sending them via SMTP server while logging delivery status. Following notification delivery, the system updates the status of each batch based on remaining days, recalculating urgency levels to ensure inventory status remains current. If no alert threshold is met, the process loops back to continue checking the remaining batches. This continuous monitoring ensures that notifications are delivered proactively without manual intervention and that both administrators and staff always have access to up-to-date expiry information.

e)Session Termination

Both administrators and staff can submit a Logout Request from their respective dashboards at any time. Both logout requests lead to a shared Logout process that terminates the active session and directs the flow to the END point, ensuring secure access control and proper session management.

6.5 Conclusion

The design phase has been a fundamental step for building EXPI-STOCK Business Inventory Management System as it lays the primary building blocks. The interface design was instrumental in planning and visualizing the potential user experience, with 2, security-focused wireframes showcasing a user-friendly experience via intuitive navigation, color-coded status indicators for visual clarity, as well as role-based access to both administrator & staff sections.

A database design has also been discussed, outlining each table with all its data items with the corresponding data types, formats, descriptions and example values. The data model contains nine tables which are related to one another in a way that will make sure the integrity, security and efficient

performance of the DB. The data dictionary improves the understanding of the structure and management of data in EXPI-STOCK.

Besides, various informative figures have been utilized, such as the ERD, DFD and other visual tailored blocks to elaborate the structure of the system and data flows. All billion diagrams are unified into one holistic blueprint to steer the development team toward implementing EXPI-STOCK that responds to Wardah Baiduri's fundamental need: from expired product loss; inefficiency of manual tracking; up to ineffective FIFO stock rotation practices.

7 IMPLEMENTATION

7.1 Introduction

This chapter will focus on the implementation which is phase of EXPI-STOCK Business Inventory Management System and also describe the execution details related to project. The implementation stage is where the theoretical design comes to life in a fully-functioning software product, with all its modules developed, integrated, and tested for compliance to end-user requirements. In this part of the report details behind. The tools that were used on creating EXPI-STOCK itself, the interface for system in real time according to what the wireframes were prepared beforehand and most importantly; functions of importance towards facets of system security.

7.2 Execution Platform

This section describes the execution platform where we developed and ran EXPI-STOCK system. The platform fit the required development tools, was appropriate for an application designed to be web based and existed within the project environment.

7.2.1 Windows 11

Figure 58: Windows 11 (Wright, 2021)

Through the development process to EXPI-STOCK, Windows 11 was the most improvised operating system from where the entire web application hardware device developed and executed. Wright (2021) discusses how Windows 11 has been reimagined to be much more efficient on ensuring that the operating system runs at full potential. This new upgrade will help benefit better to ease up project execution were switching between and running multiple applications at a time becomes lot easier due to faster start-up times, smoother app-transition and quick responsive ability.

7.3 Implementation Tools

This section is presented the main implementation tools used in developing the EXPI-STOCK system. They were chosen for their utility to rapidly build web-based applications and their ease of use in integration with the technology stack selected, as well as coding, debugging and deployment workflows.

7.3.1 Visual Studio Code

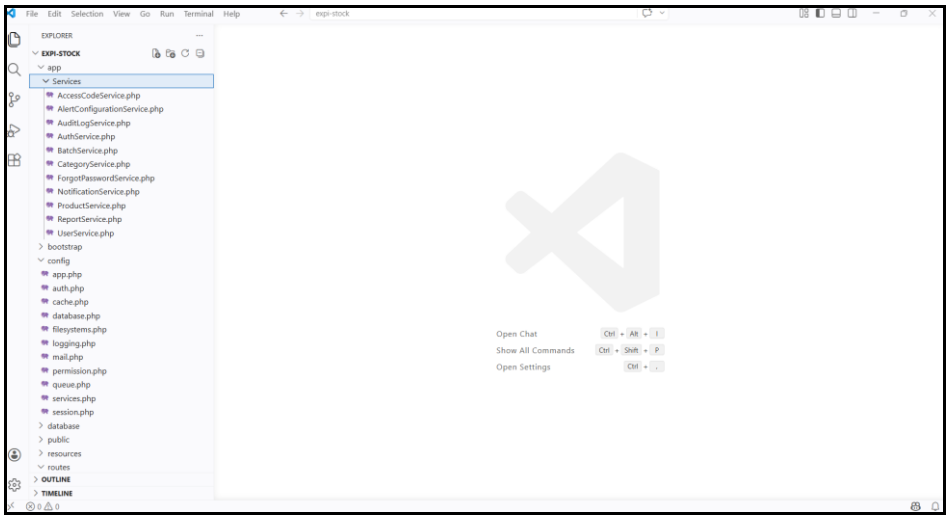


Figure 59: Homepage to Visual Studio Code

Visual Studio (VS) Code is one of the plenty coding software that help developers to start their programming journey. Eventually, after finding many criteria of lightweight, simplicity and powerful code editor platform finally VS Code became the perfect choice to use daily. Supporting many types of programming languages, you can say that VS Code helps to make you more productive with syntax and semantic highlighting, bracket-matching, auto-indentation, box-selection etc., making your implementation millisecond fast and for the better. Microsoft (2022) states inadequate that those capabilities include syntax highlighting, code autocompletion (IntelliSense), debugging, and an integrated terminal which lead to improvements in developer productivity and quality of code. It also provides a massive extension

marketplace, which many of you used for things like real-time previewing and code formatting and backend programming productivity. The unique simplicity of VS Code and its perfect set of features during that time made it a natural candidate to build and maintain the entire application stack, from backend scripts to frontend interfaces.

7.3.2 MySQL Database

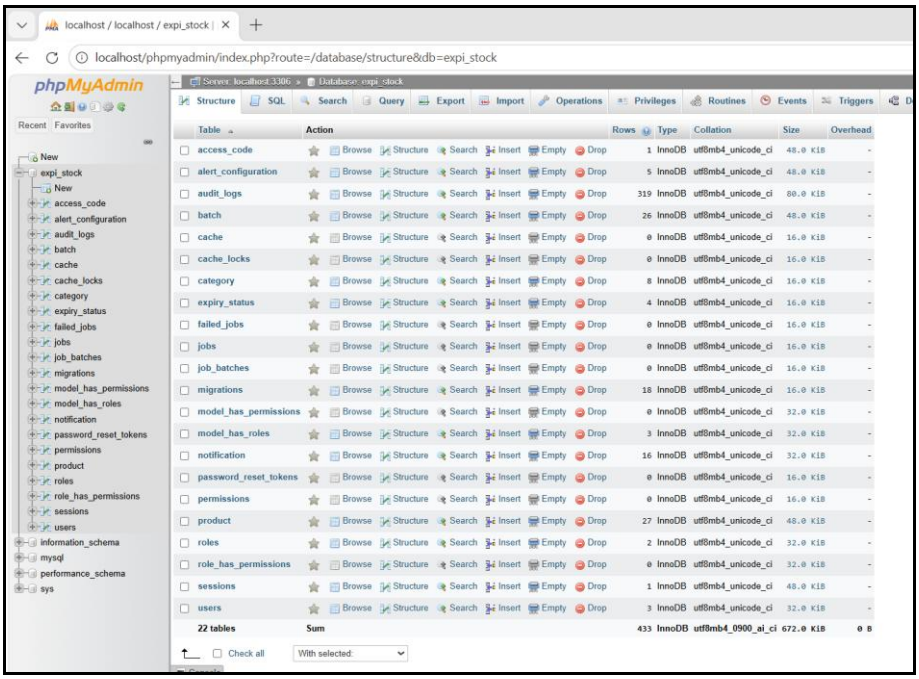


Figure 60: phpMyAdmin

MySQL is the most popular open source-database management system acting as repositories. SQL (Structured Query Language) is a command language that allows for data manipulation of the data within a database. Examples will include inserting, updating, retrieving and deleting rows (records) stored in tables inside the relational database. phpMyAdmin is a web-based, open-source MySQL and MariaDB database administration tool written in the Hypertext Preprocessor (PHP) programming language that enables configuration and management of databases via a graphical interface. We used the following services as it follows simple way of querying and designed specifically to host web services.

7.3.3 HTML

Figure 61: HTML (OWS, 2025)

HyperText Markup Language (HTML) is the well-recognized markup language used to create and construct web-pages. HTML is the backbone: It creates the structure and web page design with elements like headings, paragraphs, hyperlinks, graphics and forms for all webpages. HTML was used to design the base of the user interface for this project, where it helps in displaying all the content and inputs from the users as clearly and readably as possible. Due to its compatibility with other front-end technologies like CSS and JavaScript, it provides the dual advantage of enhancing user interface (UI) as well as improving the user experience (UX). Each of the web pages described consists only of HTML and possibly some very simple CSS or JavaScript, but its simplicity (in conjunction with wide browser support and ease of use) led to adopting it as a required tool for front-end web development within OWS (2025).

7.3.4 CSS

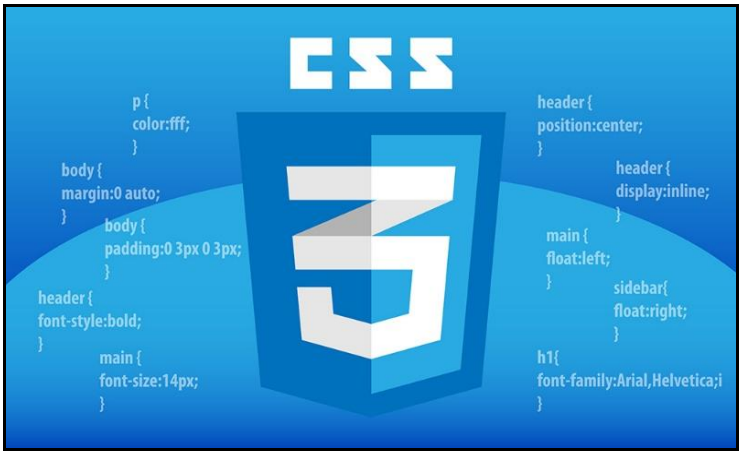


Figure 62: CSS (Oxford Web Studio, 2023)

CSS stands for Cascading Style Sheets and it uses to style HTML tag on a web page. It also decouples content from design, enabling developers to use styles like colors, fonts, spacing and layouts once in multiple pages. Css helps create flexible and visually engaging layouts without the need to rewrite the basic html code (Oxford Web Studio, 2023). We used CSS in this project to enhance the visual

presentation of a user interface, providing neat, responsive and user-friendly designs that ensure all pages of EXPI-STOCK system have consistent look and feel.

7.3.5 PHP

Figure 63: PHP (Ded9, 2022)

PHP (Hypertext Preprocessor) is a popular general-purpose scripting language that is especially suited to web development and creating dynamic content. PHP continues to be widely used for server-side scripting along with its ability to integrate well with databases and general support in hosting (Ded9, 2022). It is inserted in HTML and run on the server side. this makes it a great option for form submission, data processing or database interaction. The project used PHP with the Laravel framework to handle backend logic and features like user authentication, role-based access control, automated expiration reminders and database access. The database's compatibility with MySQL and the ability to easily integrate into web applications made it a solid choice for adding dynamic, data-driven functionality to EXPI-STOCK.

7.3.6 Hardware

Table 23: Hardware Specifications Table

Specification	Details
Device	HP Laptop 15-fc0xxx
Windows Edition	Windows 11 Home Single Language
Processor	AMD Ryzen 5 7430U with Radeon Graphics 2.30 GHz
Memory (RAM)	16.0 GB
System Type	64-bit operating system, x64-based processor
Storage	477 GB

7.4 System Interface

The following section introduces the EXPI-STOCK implemented system interfaces deployed in a live environment. The user interfaces are illustrated using screenshots from the live system, demonstrating the end result of the implementation phase for each type of user in this case study i.e. administrator and staff.

7.4.1 Admin Interface

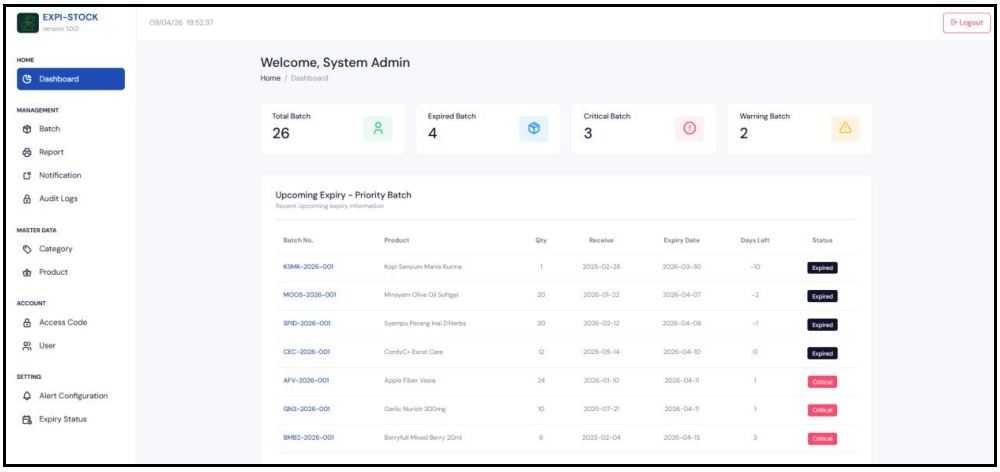


Figure 64: Admin - Dashboard Interface Part 1

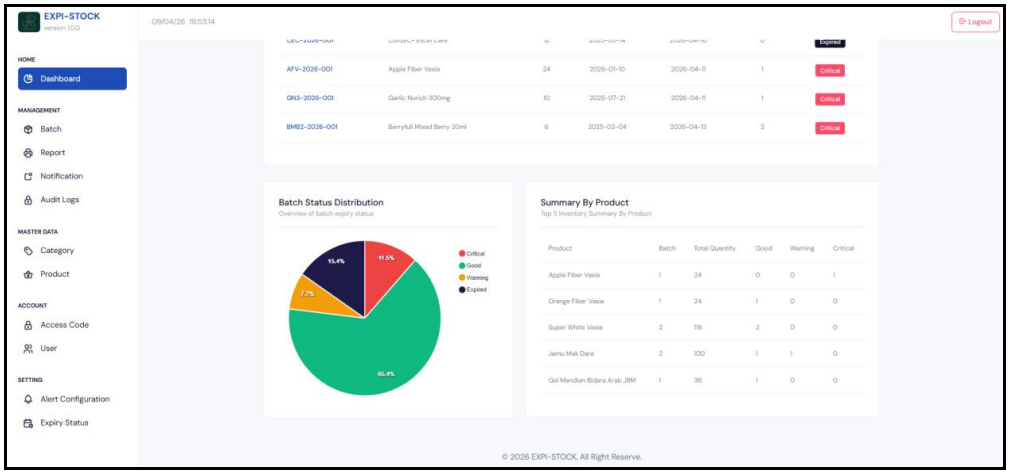


Figure 65: Admin - Dashboard Interface Part 2

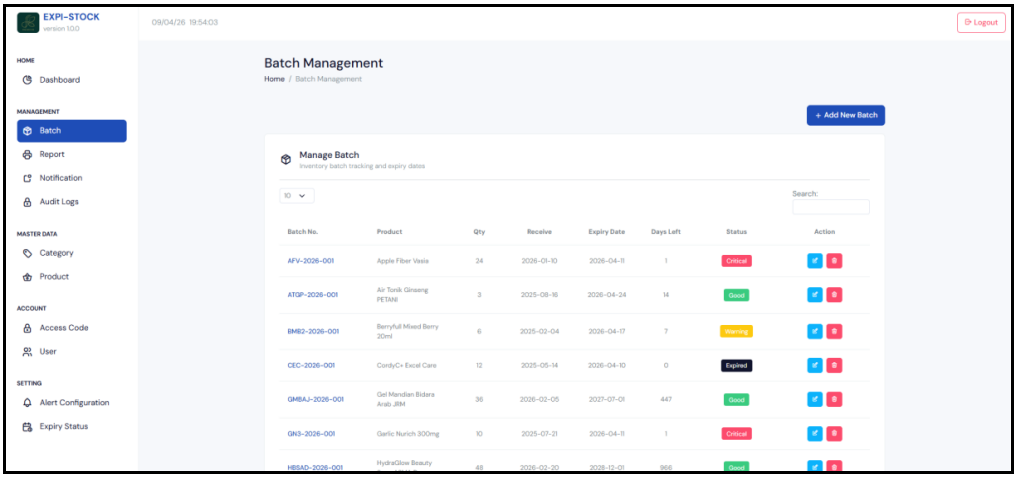


Figure 66: Admin - Batch Management Interface

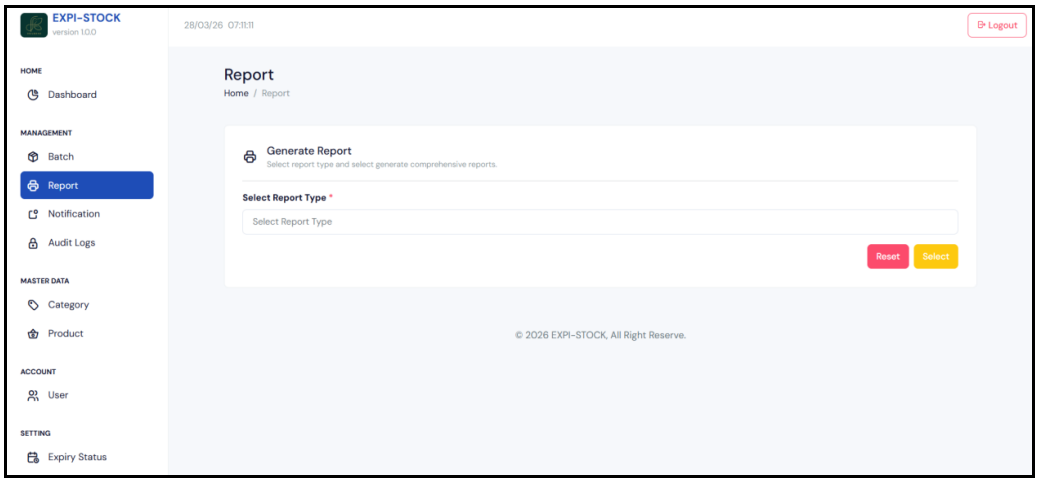


Figure 67: Admin - Report Interface

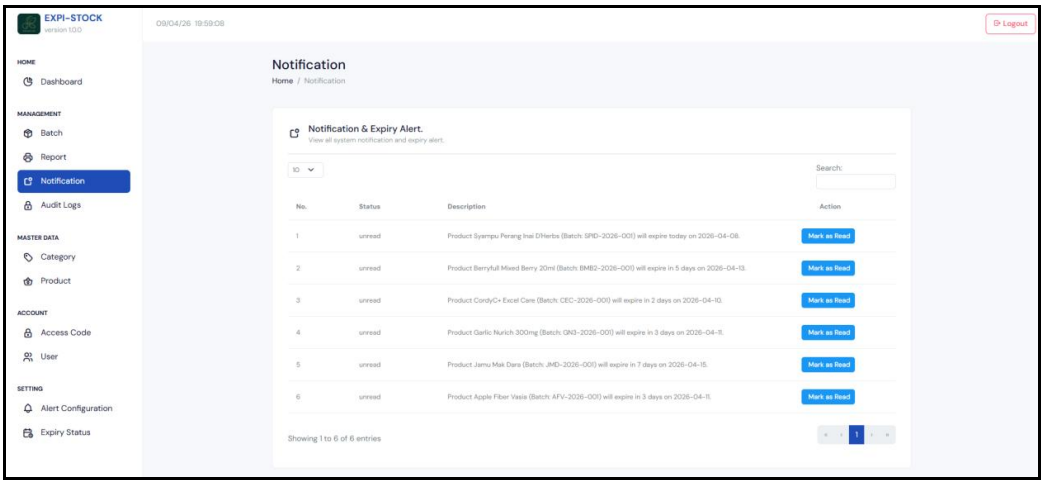


Figure 68: Admin - Notification Interface

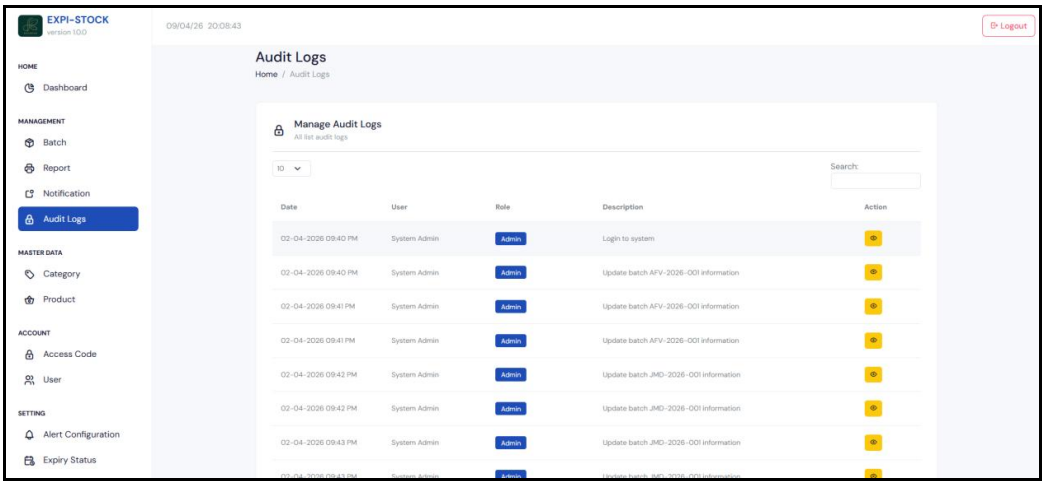


Figure 69: Admin - Audit Logs Interface

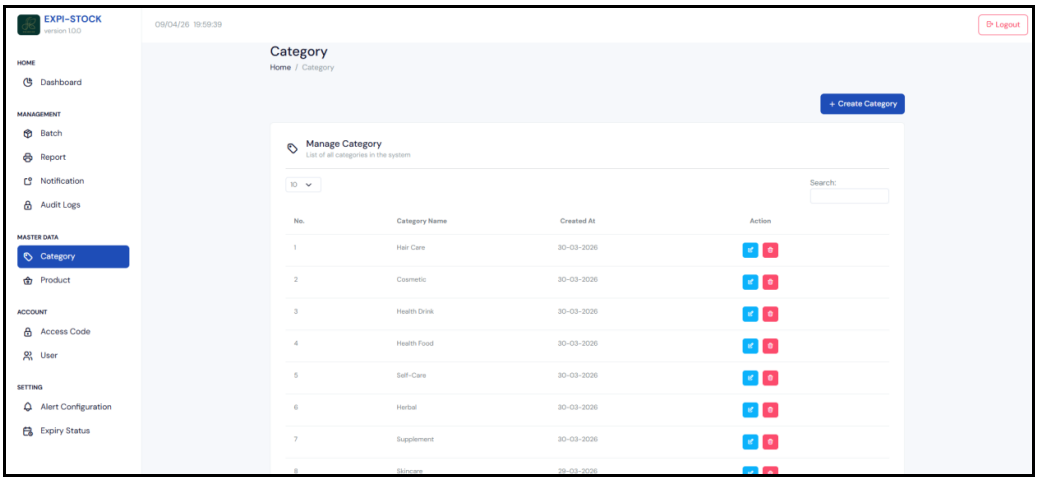


Figure 70: Admin - Category Interface

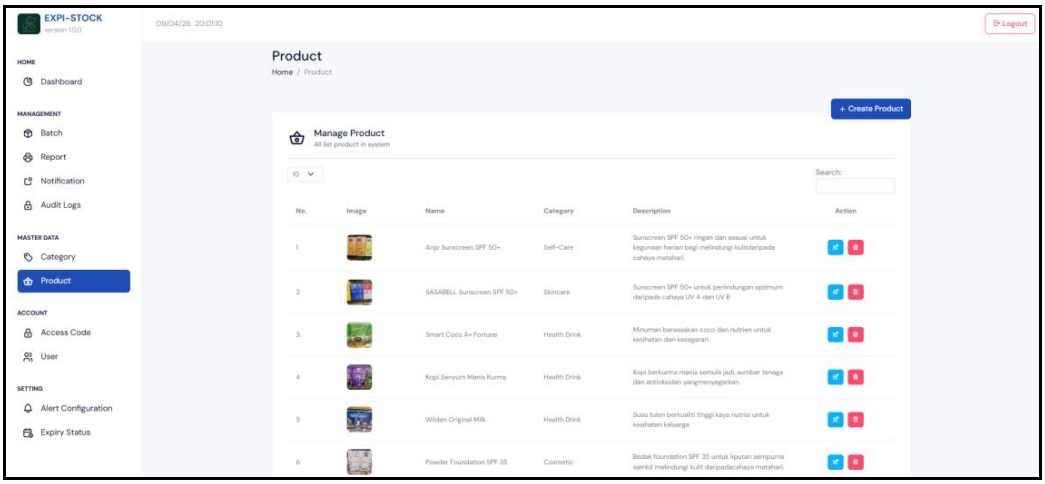


Figure 71: Admin - Product Interface

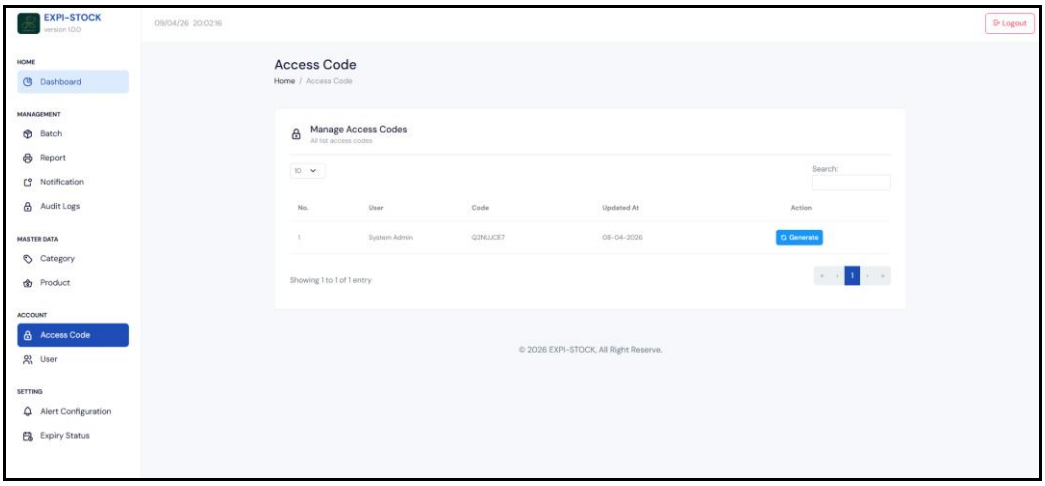


Figure 72: Admin - Access Code Interface

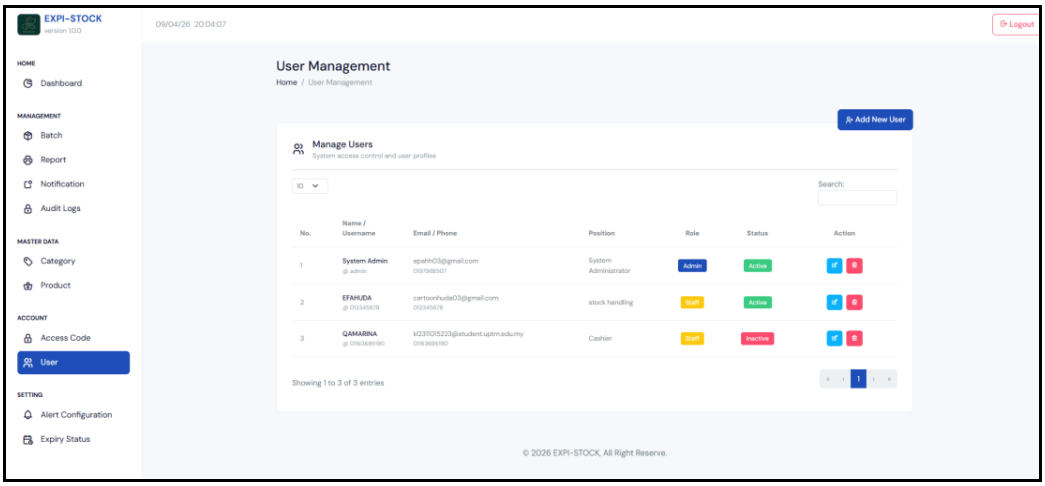


Figure 73: Admin - User Management Interface

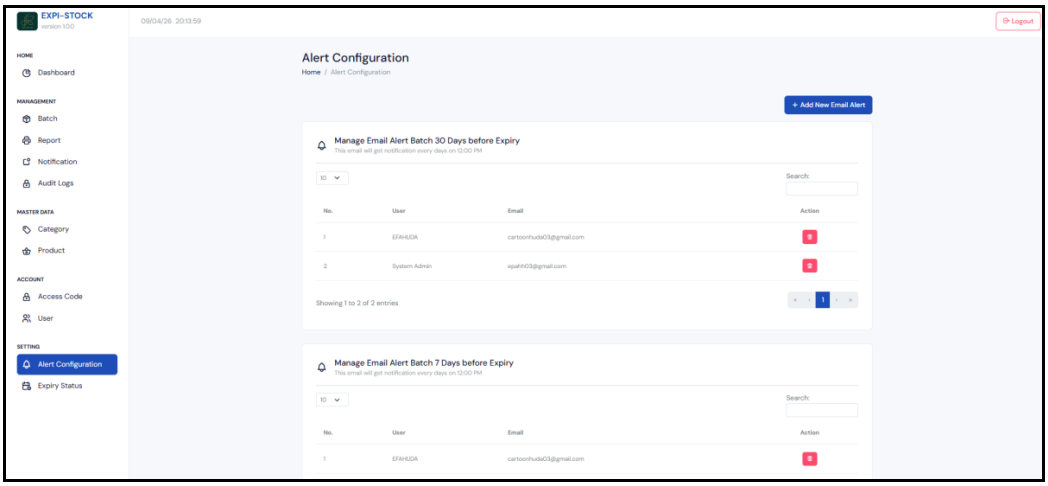


Figure 74: Admin - Alert Configuration Interface Part 1

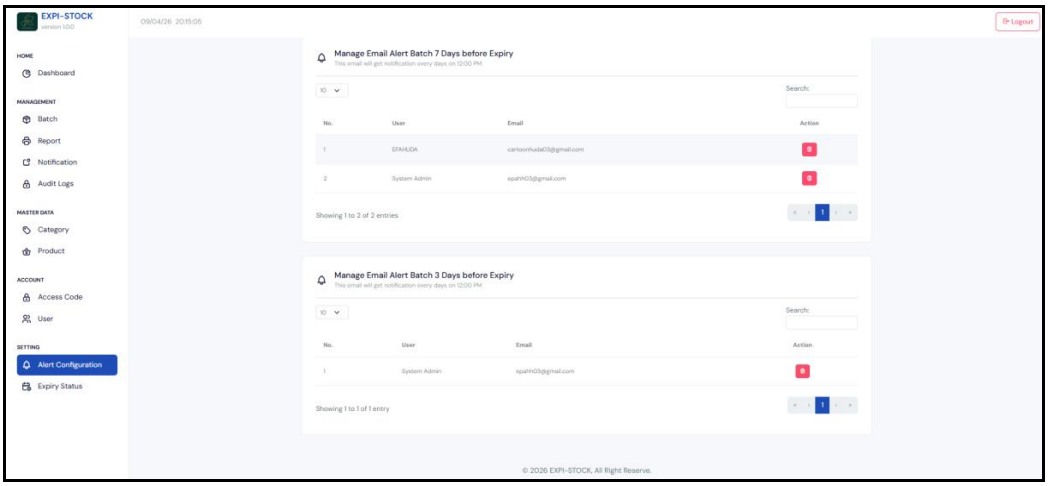


Figure 75: Admin - Alert Configuration Interface Part 2

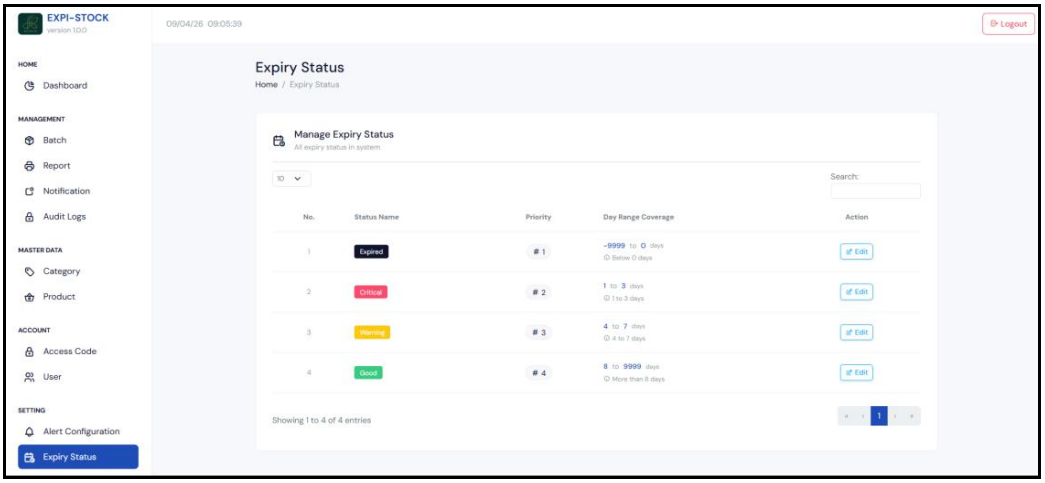


Figure 76: Admin - Expiry Status Interface

The above figures display all the interfaces or pages of EXPI-STOCK for the Admin side. Administrators have access to a comprehensive dashboard displaying key metrics including Total Products, Total Batches, Critical Alerts, and Warnings (Figure 64 and 65). Administrators can manage product batches including creating, editing, and deleting batch records with expiry tracking in Figure 66, generate inventory and expiry status reports with PDF export capabilities in Figure 67, view and manage expiry alert notifications in Figure 68, and review all audit logs for system activity monitoring in Figure 69. Additional administrative features include category management in Figure 70, product management in Figure 71, access code management for secure admin registration in Figure 72, user account management for both admin and staff users in Figure 73, email alert configuration in Figures 74 and 75, and expiry status threshold configuration in Figure 76.

7.4.2 Staff Interface

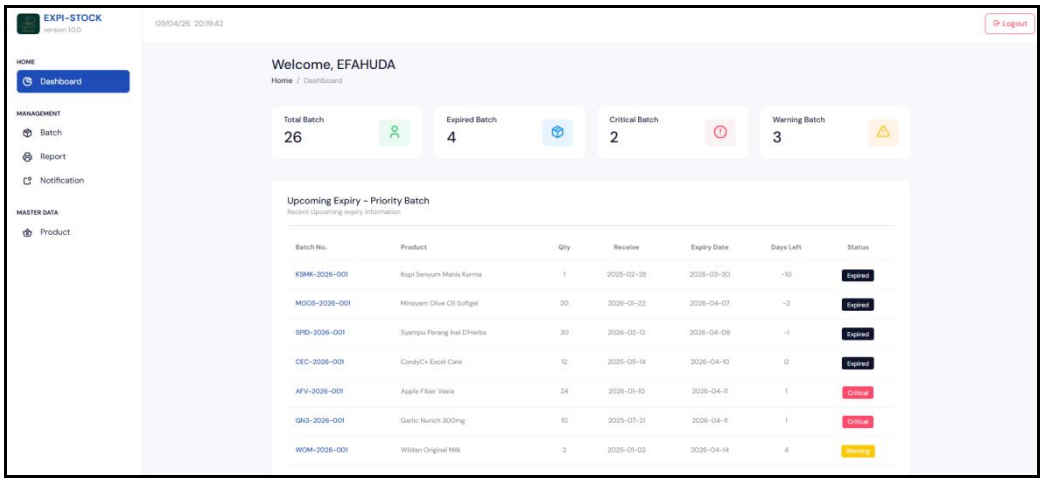


Figure 77: Staff - Dashboard Interface Part 1

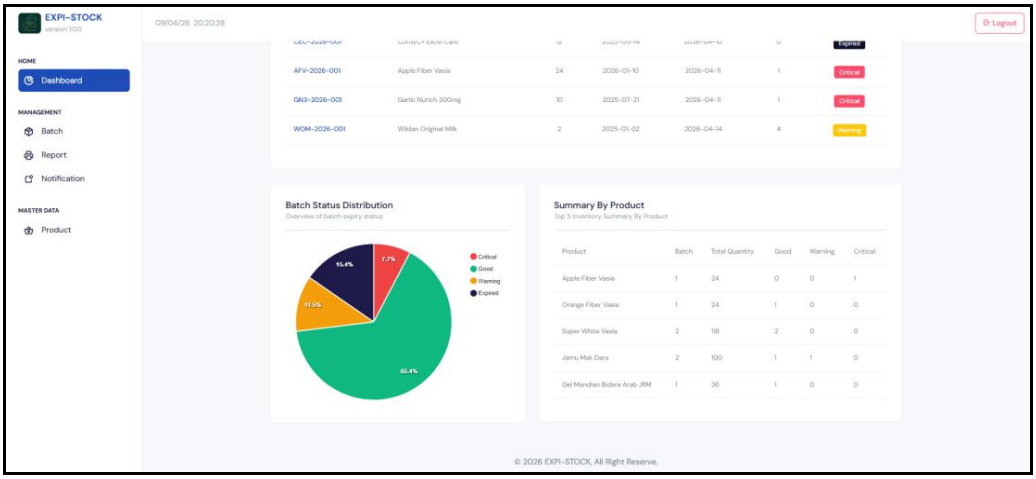


Figure 78: Staff - Dashboard Interface Part 2

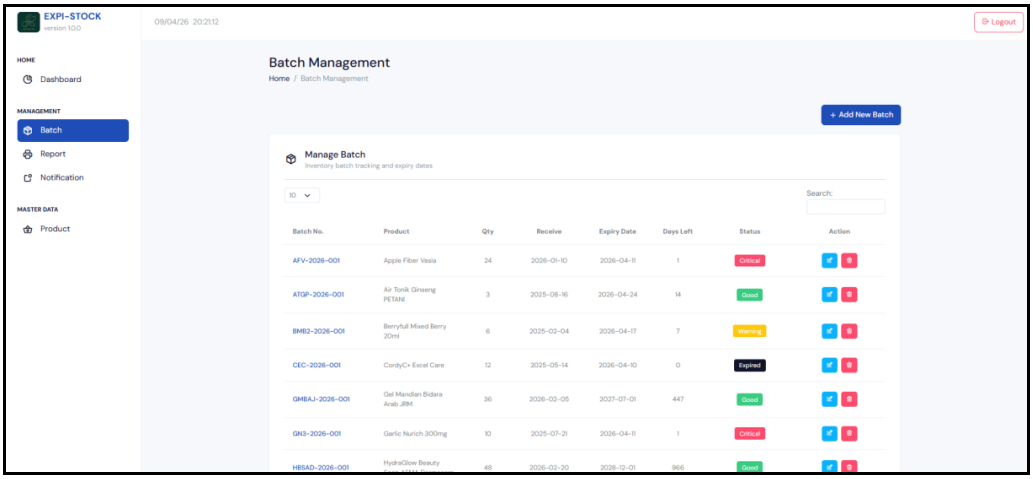


Figure 79: Staff - Batch Management Interface

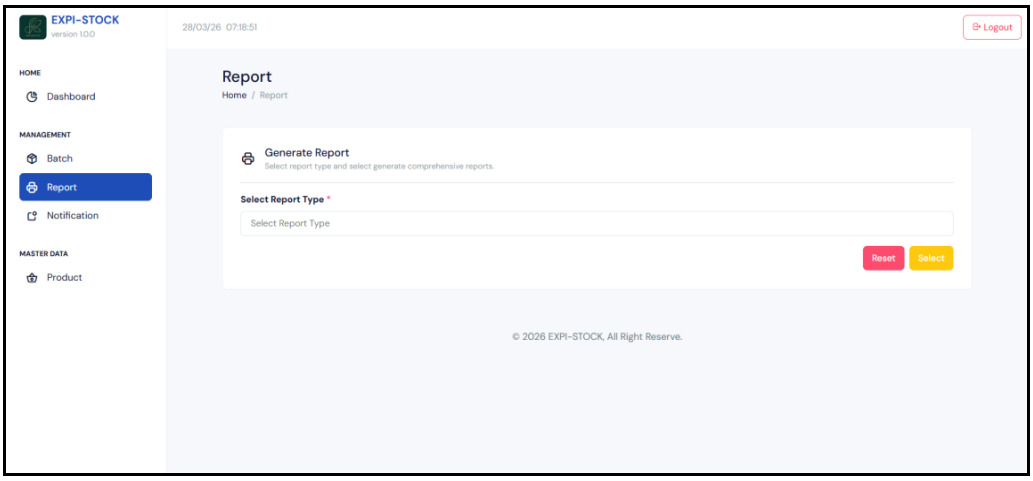


Figure 80: Staff - Report Interface

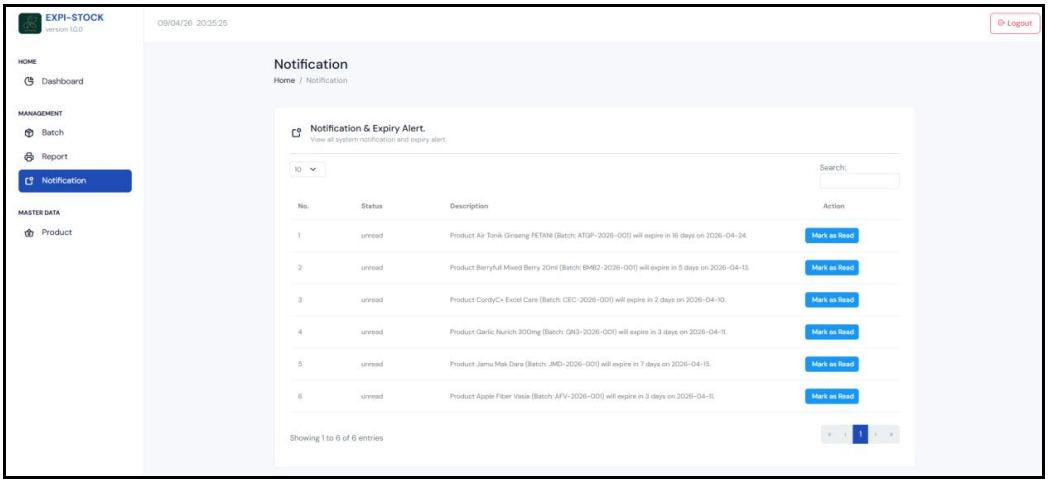


Figure 81: Staff - Notification Interface

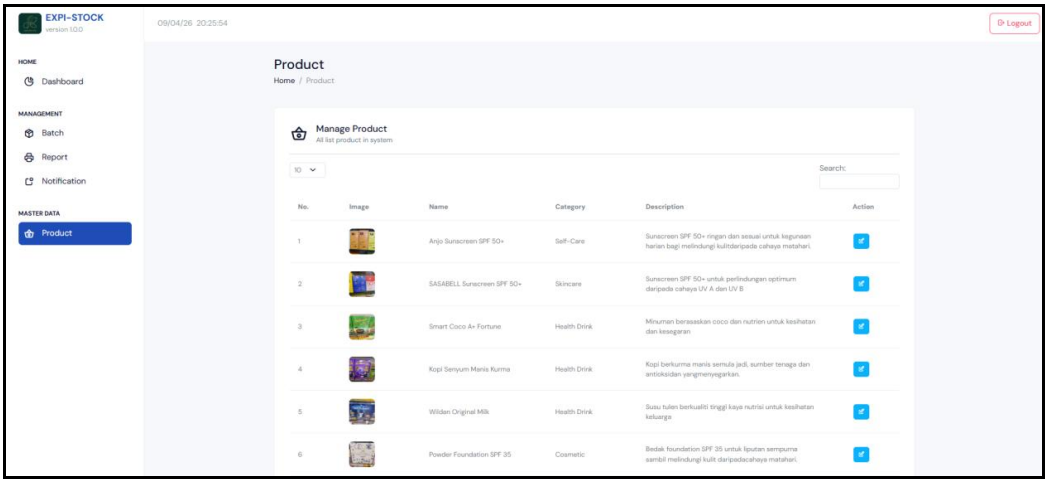


Figure 82: Staff - Product Interface

The above figures display all the interfaces or pages of EXPI-STOCK for the Staff side. Staff members can view a dedicated dashboard showing product summary cards for Good, Warning, Critical,

and Total items, along with FIFO priority lists in Figures 77 and 78. Staff can perform batch management operations including viewing, adding, editing, and updating batch quantities as shown in Figure 79. Staff are also able to generate and export product inventory and expiry status reports in Figure 80, view expiry alert notifications with details on products requiring immediate attention in Figure 81, and browse the product catalog with search and filter capabilities in Figure 82.

7.5 Significant Functions

This section highlights the significant functions implemented within the EXPI-STOCK system that are critical to its core operational objectives. These functions underpin the system's ability to automate expiry tracking, generate timely notifications, and enforce FIFO and FEFO stock rotation across all product batches managed by Wardah Baiduri.

7.5.1 Database

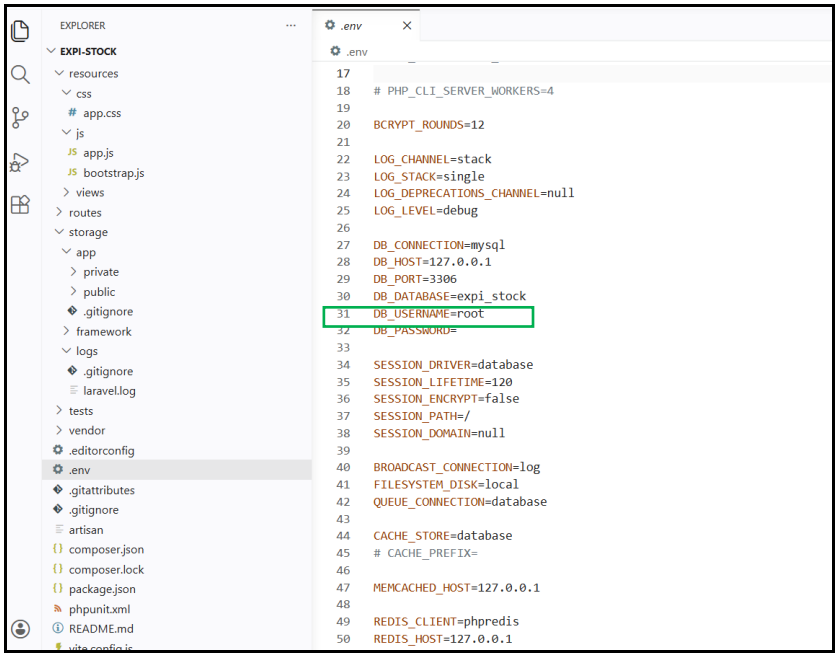


Figure 83: Code Segment for Database

The code segmentation above displays the implementation of the database connection configuration used to initiate connection between the EXPI-STOCK system and its MySQL database. Laravel's database configuration, however, specifies the variables and connection parameters as necessary including the hostname of the database system, name of the database you will be using, username, password string. This allows us to declare the database connection configuration only in the environment configuration file and reference this same connection in all models and controllers, avoiding unnecessary repetition of the declaration of connection parameters.

7.5.2 Login

```php
Route::get('/', function (){
    return redirect()->route('login');
});

// authentication
Route::get('/login', [AuthController::class, 'loginView'])->name('login');
Route::get('/logout', [AuthController::class, 'logout'])->name('logout');
Route::post('/login/check', [AuthController::class, 'login'])->name('login.check');

// forgot password
Route::get('/forgot_password', [ForgotPasswordController::class, 'forgotPasswordView'])->name('forgot_password');
Route::post('/forgot_password', [ForgotPasswordController::class, 'sendResetEmail'])->name('forgot_password.send');

Route::get('password/reset/{token}', [ForgotPasswordController::class, 'showResetForm'])
    ->name('password.reset');


Route::post('password/reset', [ForgotPasswordController::class, 'updatePassword'])
    ->name('password.update');
```

Figure 84: Code Segment for Login

Once the user fills in their email/username and password inside the login form, the data will then be posted to the back-end for Laravel's authentication system to process the provided information. The authentication system verifies whether the user exists, checks whether the password matches the bcrypt hashed value stored in the database, and also validates the Admin Access Code for administrator logins as an additional security measure. If the login attempt succeeds, the user will then be redirected to their role-appropriate dashboard, while a failed attempt will prompt the user to retry again with an appropriate error message displayed.

7.5.3 Logout

```php
// authentication
Route::get('/login', [AuthController::class, 'loginView'])->name('login');
Route::get('/logout', [AuthController::class, 'logout'])->name('logout');
Route::post('/login/check', [AuthController::class, 'login'])->name('login.check');
```

Figure 85: Code Segment for Logout

The logout function is responsible for ending a user's session securely in the EXPI-STOCK system. This is a significant component in managing authenticated user access and system security. The logout mechanism destroys the current authenticated session and redirects the user back to the login page, ensuring that all confidential data is cleared from the session and that the user must log in again to obtain access, thus maintaining the security integrity of the system.

7.5.4 Set the Time for Get Email Expiry Alert

Figure 86: Code Segment for Set the Time for Get Email Expiry Alert

The code segment above configures the automated scheduling of expiry alert emails using Laravel's task scheduling system. The scheduler is configured to run a daily check at 6:00 AM every morning, triggering the automated process to evaluate all active product batches against the configured alert thresholds. This automated daily check ensures that expiry notifications are sent proactively without requiring manual intervention from administrators or staff, providing continuous monitoring of the inventory expiry status throughout each day.

7.5.5 Expiry Email Notification

Figure 87: Code Segment for Expiry Email Notification

The code segment above illustrates the implementation of the automated expiry email notification feature in EXPI-STOCK. When the daily scheduled task runs, the system queries all active product batches, calculates the days remaining until each batch expires, and compares these values against the configured alert thresholds of 30, 7, and 3 days. When a threshold is matched, the system generates a

notification record in the database, composes an email containing the product name, batch number, expiry date, and days remaining, and dispatches it to all configured recipient email addresses via the SMTP mail server. This ensures that administrators and relevant staff are promptly informed about products approaching their expiry dates.

7.5.6 Audit Logs

Figure 88: Code Segment for Audit Logs

The code segments above illustrate the implementation of the Audit Logs feature in EXPI-STOCK, which is responsible for recording all significant user actions and system modifications. Every time an authenticated user performs an action such as Login, Logout, Create, Update, or Delete, the system automatically writes an entry into the audit_logs table containing the user ID, action type, a detailed description of what was performed, and the timestamp of the action. This comprehensive audit trail ensures accountability and traceability, allowing administrators to review all system activities through the Audit Logs interface for security monitoring and compliance purposes.

7.6 Conclusion

This chapter gives comprehensive explanation about the various tools used to develop EXPI-STOCK, a web-based system. This extends from the software being used to the programming languages, but also the hardware specification too. The front-end was developed in such a way as to use HTML and CSS language to show the user interface elements of the system maintaining a visual structured layout for better user experience whereas Laravel PHP is used for all backend functionalities which include authentication, role-based access control, automated expiry monitoring, and database functionality. The interfaces that are demonstrated in this chapter are consistent with the requirements defined previously during the design and planning stages.

8 TESTING

8.1 Introduction

The testing phase of EXPI-STOCK (Expiry Stock Inventory Management System) development life cycle is discussed in this chapter. Testing is the most essential activity in the software development lifecycle that helps you verify and validate whether a system meets its specifications and works as intended under different conditions. An overview of the types and approach to testing undertaken for EXPI-STOCK using standard development methodologies, such as Waterfall: All levels of stages completed through quality assurance (QA) processes before releasing the system into production.

This testing stage has many goals such as identifying and fixing any defects within system functionality, checking that all functional & non-functional requirements are met, confirming that the system meets the operational requirements of Wardah Baiduri Health and Beauty Product Shop, ensuring both Administrator & Staff user roles can complete their tasks without errors.

8.2 Unit Testing

Unit testing of each function and module before integration with other components System Testing, tested each unit with its expected input and output behaviour to catch errors as early in the testing cycle as possible.

8.2.1 Overview

Unit testing tests the smallest testable parts of an EXPI-STOCK system in isolation to verify each function works as intended. For each test case, it defines what user role is making the action, wat unit function are we testing, a description of what needs to go in the test, provided input, expected output, actual output during the throughput and pass/fail status.

Unit tests have been made for both Administrator and Staff roles with all essential functional requirements outlined on Chapter 4. This is comprised of logging in and session validation, handling batch

and product management operations, setting alert properties, creating dashboards and reports, managing account profiles for users, verifying access to the audit log, notification processing and interacting with categories.

8.2.2 Unit Testing Results

Table 24: Unit Testing Table

User Role	Unit/Function	Description	Input	Expected Output	Actual Output	Status
Admin	Login (Admin)	Validates admin credentials with access code	Enter username, password and Admin Access Code	Redirected to Dashboard	Successfully logged in; Dashboard displayed	**Pass**
Admin	Dashboard View	Displays batch summary, alerts and charts	Navigate to Dashboard	Total Batch, Expired, Critical, Warning counts displayed with charts	All counts and charts displayed correctly	**Pass**
Admin	Add New Batch	Admin creates a new stock batch entry	Fill in Product, Quantity, Receive Date, Expiry Date and click Save Batch	Batch saved; appears in Batch Management list	Batch created successfully; notification scheduled	**Pass**
Admin	Edit Batch	Admin updates existing batch information	Click edit icon, modify fields, click Save Changes	Batch record updated in database	Batch updated; changes reflected in list	**Pass**
Admin	Delete Batch	Admin removes a batch from inventory	Click delete icon; confirm in pop-up	Batch removed from Batch Management list	Batch deleted; list updated	**Pass**
Admin	Generate Report (Product Inventory)	Generates a product inventory report	Select Product Inventory report type, click Select	Report table displayed with product, category, total batches,	Report generated; Export PDF button available	**Pass**

User Role	Unit/Function	Description	Input	Expected Output	Actual Output	Status
				total quantity		
Admin	Generate Report (Expiry Status)	Generates expiry status report	Select Expiry Status report type, click Select	Report shows expired, critical, warning counts per product with earliest expiry date	Expiry status report generated correctly	**Pass**
Admin	Export PDF Report	Exports the generated report as a PDF file	Click Export PDF button after generating report	PDF file downloaded with correct report data	PDF exported successfully with accurate data	**Pass**
Admin	Notification Centre	Displays system expiry alert notifications	Navigate to Notification page	List of unread alerts with product, batch, days left, Mark as Read button	All notifications listed with correct details	**Pass**
Admin	Mark Notification as Read	Marks an alert notification as read	Click Mark as Read on a notification	Notification status updated to read	Notification marked as read; status updated	**Pass**
Admin	Audit Logs View	Admin views all user activity logs	Navigate to Audit Logs	All system actions listed with date, user, role, description	Audit log entries displayed correctly	**Pass**
Admin	Audit Logs Detail	Admin views detail of a specific audit log entry	Click view icon on an audit log entry	Date, action, and description shown for selected entry	Audit log detail page displayed correctly	**Pass**
Admin	Manage Category (Add)	Admin creates a new product category	Click Create Category, enter name, click Save Changes	New category appears in category list	Category created and listed successfully	**Pass**
Admin	Manage Category (Edit)	Admin edits an existing category name	Click edit icon, modify category	Category name updated in	Category name updated	**Pass**

User Role	Unit/Function	Description	Input	Expected Output	Actual Output	Status
			name, click Save Changes	list	successfully	
Admin	Manage Category (Delete)	Admin deletes a category	Click delete icon; confirm in pop-up	Category removed from list	Category deleted; list refreshed	**Pass**
Admin	Manage Product (Add)	Admin creates a new product with image	Fill in category, name, brand, price, supplier, upload image, enter description, click Save Changes	Product saved and appears in Product list with image	Product created and displayed correctly	**Pass**
Admin	Manage Product (Edit)	Admin updates product information	Click edit icon, modify fields, click Save Changes	Product record updated	Product information updated successfully	**Pass**
Admin	Manage Product (Delete)	Admin deletes a product	Click delete icon; confirm in pop-up	Product removed from list	Product deleted successfully	**Pass**
Admin	Access Code Management	Admin generates new access code for login	Navigate to Access Code page, click Generate	New access code generated and displayed	New code generated; updated in system	**Pass**
Admin	User Management (Add)	Admin registers a new staff or admin user	Click Add New User, fill full name, email, phone, role, position, click Create User	New user account created; default password set to phone number	User created; account accessible with credentials	**Pass**
Admin	User Management (Edit)	Admin edits a user account	Click edit icon, modify fields, click Update User	User record updated in database	User account updated successfully	**Pass**
Admin	User Management (Delete)	Admin deletes a user account	Click delete icon; confirm in pop-up	User account removed; login blocked	User deleted; login no longer	**Pass**

User Role	Unit/Function	Description	Input	Expected Output	Actual Output	Status
					possible	
Admin	Alert Configuration (Add)	Admin sets up email alert for batch expiry	Click Add New Email Alert, select days before and email, click Save Changes	Alert configuration saved; user receives alerts at configured interval	Alert saved and active	**Pass**
Admin	Alert Configuration (Delete)	Admin removes an alert configuration	Click delete icon on alert; confirm in pop-up	Alert configuration removed; emails no longer sent	Alert deleted successfully	**Pass**
Admin	Expiry Status Configuration	Admin edits day range for Good/Warning/Critical/Expired labels	Click edit on a status level, modify min/max days, click Update Status	Expiry status thresholds updated; batch statuses recalculated	Thresholds saved; batch statuses updated accordingly	**Pass**
Admin	Logout	Destroys session and logs admin out	Click Logout button; confirm in pop-up	Session ended; redirected to Login page	Admin logged out; login page displayed	**Pass**
Staff	Login (Staff)	Validates staff credentials without access code	Enter username and password, click Login	Redirected to Staff Dashboard	Staff dashboard displayed after login	**Pass**
Staff	Dashboard View	Displays batch overview, FIFO priority and charts	Navigate to Dashboard	Total Batch, Expired, Critical, Warning counts and charts shown	All metrics and charts displayed correctly	**Pass**
Staff	Batch Management (View)	Staff views all batches with expiry status	Navigate to Batch Management page	List of batches with product, qty, receive date, expiry date, days left, status	All batch records displayed with correct status badges	**Pass**
Staff	Add New Batch (Staff)	Staff creates a new batch entry	Click Add New Batch,	Batch created and	Batch saved and appears	**Pass**

User Role	Unit/Function	Description	Input	Expected Output	Actual Output	Status
			select product, enter quantity, receive date, expiry date, click Save Batch	listed in Batch Management	in list	
Staff	Edit Batch (Staff)	Staff updates batch details	Click edit icon, modify fields, click Save Changes	Batch record updated in database	Changes reflected in batch list	**Pass**
Staff	Delete Batch (Staff)	Staff deletes a batch entry	Click delete icon; confirm in pop-up	Batch removed from list	Batch deleted; list updated	**Pass**
Staff	Report (Product Inventory)	Staff generates product inventory report	Select Product Inventory, click Select, click Export PDF	Report showing product, category, total batches, total quantity	Report generated; PDF exported correctly	**Pass**
Staff	Report (Expiry Status)	Staff generates expiry status report	Select Expiry Status, click Select, click Export PDF	Report showing expiry counts per product with earliest expiry	Expiry report generated; PDF downloaded	**Pass**
Staff	Notification (View & Mark Read)	Staff views alerts and marks them as read	Navigate to Notification, click Mark as Read	Alert updated to read status	Notification marked as read	**Pass**
Staff	Manage Product (Edit)	Staff edits product details	Click edit icon, update fields, click Save Changes	Product record updated	Product updated successfully	**Pass**
Staff	Logout (Staff)	Destroys staff session	Click Logout; confirm in pop-up	Session ended; redirected to Login page	Staff logged out; login page shown	**Pass**

All unit test cases for both Administrator and Staff role show Pass status as shown in Table 8.1 All the essential functionality from authentication, to product and batch management; alert configuration, FIFO

priority display for users, notification management and even audit log tracking has been validated correctly against specified requirement.

8.3 Integration Testing

EXPI-STOCK code in the tool was also relied on to conduct integration testing to examine how various modules of the system interacted with one another, checking whether data circulated accurately between components and verifying that combined functionalities behave as expected. It ensured the validation of end-to-end workflows from frontend interfaces, backend processing logic, database operations, to automated notification delivery.

8.3.1 Overview

Integration testing checks that individual modules interact as expected with each other. Unlike unit testing that tests components in isolation, integration testing ensures that data flows correctly between modules and the expected system-level behaviour is produced by combined components.

In EXPI-STOCK, we chose to focus on integration testing two major pipelines and some child pipelines consisting of operations that the parent cowers a loop iterating over, which included: the authentication and session management pipeline; the batch creation and notification scheduling pipeline; the expiry monitoring and FIFO display update pipeline; automated alert generation and email delivery pipeline; admin audit log recording pipeline; and, role-based access enforcement pipeline.

8.3.2 Integration Test Scenarios

Table 25: Integration Testing Table

Integration Scenario	Modules Involved	Description	Input	Expected Result	Status
Login to Dashboard Flow	Authentication Module → Session Manager → Dashboard Module	After valid login, session is created and dashboard loads with correct user-role-specific data	Enter valid credentials and click Login	Dashboard displayed with role-specific menu and data	**Pass**
Batch Creation to	Batch Management	When a batch is created with an	Admin fills in batch form and	Batch saved; notification alerts scheduled in database for	**Pass**

Integration Scenario	Modules Involved	Description	Input	Expected Result	Status
Notification Scheduling	Module → Database → Notification Scheduler	expiry date, the system automatically schedules alert notifications at configured thresholds (30, 7, 3 days)	submits	configured days before expiry	
Expiry Monitoring to FIFO Display Update	Automated Monitoring Process → Batch Status Updater → FIFO Priority Display	Daily process recalculates days until expiry for all batches and updates the colour-coded FIFO priority display accordingly	System clock triggers daily check	Batch statuses (Good/Warning/Critical/Expired) updated; FIFO display reflects new status with correct colour codes	**Pass**
Alert Generation to Email Delivery	Notification Scheduler → Email Composer → SMTP Service → User Inbox	When threshold is reached, notification record is created, email is composed with correct batch details and sent via SMTP	Batch reaches configured expiry threshold	Email alert delivered to configured recipient email with correct product and batch information	**Pass**
Admin Audit Log Recording	Admin Actions → Audit Logger → Database → Audit Logs Module	Every admin action (login, create, update, delete) is captured and stored in the audit log with timestamp and user ID	Admin performs product add, batch edit, and user management actions	All performed actions recorded accurately in Audit Logs with date, user, role, and description	**Pass**
Role-Based Access Enforcement	Authentication Module → Role Manager → Route Guard	Staff users are restricted from accessing admin-only pages such as User Management, Audit Logs, and Access Code settings	Staff logs in and attempts to navigate to admin pages	Access denied for admin pages; staff restricted to permitted features only	**Pass**

All integration test scenarios passed successfully. The test results confirm that data flows correctly between the authentication, batch management, notification scheduling, FIFO display, audit logging, and role access control modules. No critical integration defects were identified during this phase.

8.4 System Testing

System testing evaluates the complete, integrated EXPI-STOCK system to verify that it meets the specified functional and non-functional requirements in an environment similar to the production environment. At this level, all modules that have passed unit and integration testing are combined and tested as a whole, focusing on realistic end-to-end scenarios and overall system behaviour rather than individual components.

For the EXPI-STOCK project, system testing was carried out using the full set of modules including authentication and session management, batch management, product management, expiry monitoring and FIFO display, automated notification and email alert delivery, report generation, user and category management, audit logging, and role-based access control. Sample data that reflects typical Wardah Baiduri operations was used, along with devices such as laptops and mobile phones running modern web browsers.

System testing was divided into two categories:

Functional Testing: checks whether EXPI-STOCK correctly performs all required business processes, such as managing batches, tracking expiry dates, sending automated alerts, generating reports, and enforcing role-based access.

Non-Functional Testing: focuses on performance, usability, security, and data consistency to ensure the system is practical and reliable for day-to-day operations at Wardah Baiduri.

8.4.1 Functional Testing

Functional testing verifies that the system behaves according to its functional requirements by testing business processes, user interactions, and data processing scenarios. In system-level functional testing, test cases are derived from the use cases and real-world workflows rather than from internal code structure.

For EXPI-STOCK, functional system testing focused on end-to-end workflows that involve multiple roles and modules, including:

- Admin login and dashboard loading with role-specific data.
- Batch creation and its effect on automated notification scheduling.
- Daily expiry monitoring and FIFO colour-coded priority display updates.
- Report generation (Product Inventory and Expiry Status) and PDF export.
- Role-based access enforcement between Admin and Staff users.

Testing was performed by executing the full scenarios using the web interface and verifying that the results on screen matched the expected behaviour and database records. Table 8.4 summarises the main functional system tests.

Table 26: Functional Testing conducted for EXPI-STOCK

Unit/Function	Description	Input (Test Procedure)	Expected Output / Error	Status
End-to-end Admin Login and Dashboard Load	To verify that an admin can log in with access code and the dashboard loads correctly with all metrics.	Admin enters valid username, password, and access code; clicks Login.	Dashboard displayed with Total Batch, Expired, Critical, Warning counts and charts; role-specific menu visible; no errors occur.	**PASS**
End-to-end Batch Creation with Notification Scheduling	To ensure that creating a batch automatically schedules expiry alert notifications at configured thresholds.	Admin fills in batch form (Product, Quantity, Receive Date, Expiry Date) and submits.	Batch saved in database; notification alerts scheduled for configured days (e.g., 30, 7, 3 days before expiry); batch appears in Batch Management list.	**PASS**
Expiry Monitoring and FIFO Priority Display Update	To test that the daily expiry monitoring process correctly recalculates batch statuses and updates the FIFO display.	System clock triggers daily monitoring process; batch expiry dates are evaluated against current date.	All batch statuses (Good/Warning/Critical/Expired) updated correctly; FIFO display reflects new colour-coded status; no batches with incorrect status remain.	**PASS**
Report Generation and PDF Export	To verify that both Product Inventory and Expiry Status reports are generated correctly and can be exported as PDF.	Admin navigates to Reports, selects report type (Product Inventory or Expiry Status), clicks Select, then clicks Export PDF.	Report table displayed with accurate data (product name, category, total batches, total quantity or expiry counts); PDF file downloaded successfully with correct report contents.	**PASS**
Role-Based Access Control Enforcement	To confirm that staff users are restricted from	Staff logs in and attempts to navigate to admin-only pages	Access denied for all admin-only pages; staff redirected to permitted pages; no admin data is exposed to staff users.	**PASS**

Unit/Function	Description	Input (Test Procedure)	Expected Output / Error	Status
	accessing admin-only pages and functions.	(User Management, Audit Logs, Access Code Management).		

The functional system tests confirm that EXPI-STOCK correctly supports the main business workflows required by Wardah Baiduri. All critical flows including admin and staff login, batch creation and notification scheduling, expiry monitoring, report generation, and role-based access control operated as expected. Any minor issues encountered during testing were corrected and retested immediately until the behaviour matched the requirements.

8.4.2 Non-Functional Testing

Non-functional testing evaluates characteristics of the system that are not directly related to specific functions, such as performance, usability, reliability, and security. These aspects are important to determine whether the system is practical and comfortable to use in real-world operational conditions at Wardah Baiduri Health and Beauty Product Shop.

For EXPI-STOCK, non-functional system testing focused on four key areas:

1. **Usability and responsiveness**: whether the system is easy to use and displays correctly on different screen sizes including desktop and mobile devices.
2. **Performance under moderate load**: whether the system remains responsive when multiple users (Admin and Staff) perform actions simultaneously.
3. **Security and access control**: whether only authorised users can access protected pages and perform role-specific functions.
4. **Data consistency and integrity**: whether batch quantities, notification statuses, and audit records remain accurate across all modules after multiple operations.

Table 27: Non-Functional Testing conducted for EXPI-STOCK

Unit/Function	Description	Input (Test Procedure)	Expected Output / Error	Status
Usability and Responsiveness	To examine layout, readability, and navigation on different devices including desktop and mobile.	Access the EXPI-STOCK system using a laptop and a mobile phone; perform typical tasks such as logging in, viewing batch list, updating batch quantity, and viewing notifications;	Layout automatically adapts to screen size; buttons and text are readable on both devices; main actions (add batch, update quantity, mark notification as read) are easy to locate; no horizontal scrolling is required for core tasks on mobile.	**PASS**

Unit/Function	Description	Input (Test Procedure)	Expected Output / Error	Status
		observe layout and interaction behaviour.		
Performance Under Moderate Load	To observe how the system behaves when multiple users perform actions within a short time window.	Simulate multiple users (Admin and Staff) performing simultaneous actions such as adding batches, viewing dashboards, and generating reports using multiple browser sessions.	System remains responsive; page load times remain within acceptable range; all submitted data is processed correctly without duplication or loss; no timeout errors occur.	**PASS**
Basic Security and Access Control	To confirm that only authorised users can access protected pages and sensitive functions.	Attempt to access system URLs without logging in; log in as Staff and try to open admin-only URLs; attempt to perform admin actions (User Management, Audit Logs) as a Staff user.	Unauthenticated users are redirected to the login page; Staff users cannot access admin pages and receive an appropriate restriction response; Admin users can access all admin sections and perform sensitive actions.	**PASS**
Session Handling and Logout	To ensure that sessions are managed securely and invalidated properly after logout.	Log in as different roles (Admin, Staff); perform some actions; click Logout and confirm; press back button or attempt to re-open protected URLs after logout.	After logout, previous session cannot be reused; accessing protected URLs redirects back to login page; no sensitive information is displayed after session termination.	**PASS**
Data Consistency and Integrity	To verify that batch quantities, expiry statuses, and notification records remain consistent	Perform a sequence of operations: add batches, update quantities, trigger expiry monitoring, mark notifications as read; then compare data	Batch quantities, expiry statuses, and notification records are consistent across all relevant pages and database records; no missing records or mismatched values; audit log accurately reflects all performed actions.	**PASS**

Unit/Function	Description	Input (Test Procedure)	Expected Output / Error	Status
	across all system modules after various operations.	across dashboard, batch list, notification centre, and audit logs.		

The non-functional testing results confirm that EXPI-STOCK is usable, responsive, and secure enough for typical usage at Wardah Baiduri outlets. The system handles moderate concurrent activity without noticeable performance degradation, enforces role-based access correctly for both Admin and Staff roles, and maintains consistent data across all modules. Minor cosmetic adjustments identified during this phase such as spacing on smaller screens were corrected before final deployment.

8.5 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) was conducted to validate that the EXPI-STOCK system meets the operational requirements and expectations of the actual end users at Wardah Baiduri. Testing was done by both administrators and staff completing tasks within the inventory system that closely matches what would occur in a real-world environment to ensure every feature works, is intuitive to use, and completes all known goals established during requirements.

8.5.1 Overview

User Acceptance Testing (UAT) is the last stage of the formal testing process, which aims to ensure that the EXPI-STOCK application meets a set of requirements and functions according to real-world operational expectations by doing so with its intended users at Wardah Baiduri Health and Beauty Product Shop. UAT was designed across 12 test scenarios that aligned with the business functional goals of EXPI-STOCK for both Administrator and Staff user roles.

Participants were instructed to complete specific tasks, and they assessed system performance relative to expected results. Test scenarios included login and access to the dashboard, batch and product management, expiry notification handling, alert configuration but were not limited to report generation, account management of users with various roles in the application system, audit log tracking and role-based access control.

8.5.2 UAT Test Cases

Table 28: User Acceptance Testing Table

Test Case ID	Test Scenario	User Role	Test Steps	Expected Result	Actual Result	Status
TC-UAT-01	Admin Login and Dashboard Access	Admin	1. Open system URL 2. Enter valid admin credentials and access code 3. Click Login	Dashboard displayed with total products, batches, alerts and charts	Dashboard displayed correctly with all metrics	**Pass**
TC-UAT-02	Staff Login and Dashboard Access	Staff	1. Open system URL 2. Enter valid staff credentials 3. Click Login	Staff dashboard displayed with product list and notifications	Staff dashboard accessible with relevant features	**Pass**
TC-UAT-03	Add New Product and Create Batch	Admin	1. Navigate to Product Management 2. Add new product with all fields 3. Navigate to Batch Management 4. Create batch for the product	Product saved; batch created with correct expiry date and notification schedule	Product and batch created; notification triggered	**Pass**
TC-UAT-04	View and Update Batch Quantity	Staff	1. Navigate to Batch Management 2. Select a batch 3. View batch details 4. Edit quantity and save	Batch quantity updated; inventory reflects new value	Quantity updated in real-time	**Pass**
TC-UAT-05	View Expiry Status and Batch Status Badges	Staff	1. Navigate to Batch Management 2. Observe colour-coded status badges 3. Check batch sorting by days left	Batches labelled as Expired, Critical, Warning, or Good with colour-coded badges	Status badges displayed correctly per configured day ranges	**Pass**
TC-UAT-06	View and Respond to	Staff	1. Navigate to Notifications	Notifications displayed with	All notification details accurate	**Pass**

Test Case ID	Test Scenario	User Role	Test Steps	Expected Result	Actual Result	Status
	Expiry Notifications		2. View alert details 3. Click Mark as Read on a notification	product name, batch, days remaining and urgency; status updated to read	and actionable	
TC-UAT-07	Configure Alert Thresholds	Admin	1. Navigate to Alert Configuration 2. Add email alert recipients 3. Configure days before expiry 4. Save configuration	Thresholds saved; future notifications follow new schedule; recipients receive email alerts	Alert thresholds saved and applied; emails delivered	**Pass**
TC-UAT-08	Generate Expiry Status Report	Admin	1. Navigate to Reports 2. Select Expiry Status report type 3. Click Select 4. Click Export PDF	Report generated showing all near-expiry and expired batches with product name, category, total batches and earliest expiry	Report generated with correct data; PDF exported	**Pass**
TC-UAT-09	Generate Product Inventory Report	Admin	1. Navigate to Reports 2. Select Product Inventory report type 3. Click Select 4. Click Export PDF	Report showing product name, category, total batches, total quantity	Inventory report generated; PDF downloaded successfully	**Pass**
TC-UAT-10	Create and Manage Staff Account	Admin	1. Navigate to User Management 2. Click Add New User 3. Fill in name, email, phone, role, position 4. Click Create User 5. Verify staff can login	Staff account created; staff able to login with provided credentials	Account created; login successful	**Pass**
TC-UAT-11	View Audit Logs	Admin	1. Login as Admin 2. Perform	All performed actions recorded	Audit log entries accurate and	**Pass**

Test Case ID	Test Scenario	User Role	Test Steps	Expected Result	Actual Result	Status
			product add and batch create actions 3. Navigate to Audit Logs 4. Click View on an entry	with timestamp, user and description	complete	
TC-UAT-12	Role-Based Access Control	Staff	1. Login as Staff 2. Attempt to access admin-only pages (User Management, Audit Logs, Access Code) 3. Observe system response	Access denied message shown; staff cannot access admin functions	Admin-only pages restricted for staff	**Pass**

8.5.3 UAT Results Summary

The UAT results demonstrate strong user acceptance of the EXPI-STOCK system. The testing results show a Pass status on all of the 12 test cases confirming that the system works well in accordance with Wardah Baiduri operational requirements. Users that were made up of administrators and staff with various levels of technical experience were able to complete all tasks assigned in a reasonable amount of time and without significant trouble.

Key findings from the UAT testers completed all Login and Dashboard Access scenarios with Admin and Staff roles at 100% success rate; the batch status colour-coded display received unanimous indication of usefulness, the automated expiry notification was confirmed to be fully functional, role-based access control accurately restricted staff from accessing admin-only pages. Other minor recommendations included a barcode scan feature and direct-report download shortcuts.

8.5.4 Client Testing and Result

Client testing is the final and most crucial validation of EXPI-STOCK system development. This stage is a live end-user from Wardah Shop to test the system in real-world operational conditions. The client ran through all major system features, providing frank and in-depth feedback on functionality, usability, performance and overall satisfaction.

8.5.4.1 Client Information

Figure 89: Interviewee Picture

Name: Zurini Binti Ismail

Position: G01/G02 Mydin Mall Taman Rinting, 81750 Masai, Johor

Date: 27 March 2026

Purpose of Interview: This client testing session aims to gather real-world functional feedback on the EXPI-STOCK system through direct hands-on evaluation by the primary stakeholder. The session covers functionality testing for both Staff and Administrator roles, as well as usability, performance, and general feedback.

Figure 90: Interview with client at Wardah Baiduri, Mydin Mall

8.5.4.2 Testing and Result - Functionality Feedback (Staff Side)

The following questions below will cover basic functionalities available to Staff users in the EXPI-STOCK system: product viewing; batch management; expiry status display; best-before notifications, and mobile access.

Table 29: UAT Client Functionality Feedback #1

QUESTION	"Were you able to log in to the EXPI-STOCK system successfully using your given credentials?"
ANSWER	"Yes. The login process was straightforward. I just entered my username and password and was redirected to the staff dashboard immediately."

Table 30: UAT Client Functionality Feedback #2

QUESTION	"Were you able to view the product list and search for products easily?"
ANSWER	"Yes. The search and filter function worked very well. I could search by product name and category without any difficulty."

Table 31: UAT Client Functionality Feedback #3

QUESTION	"Were you able to view batch information including expiry dates and days remaining?"
ANSWER	"Yes. The batch details page clearly shows the expiry date, batch number, current quantity, and how many days are left before the product expires."

Table 32: UAT Client Functionality Feedback #4

QUESTION	"Were you able to update batch quantities after selling products?"
ANSWER	"Yes. I selected the batch, entered the quantity sold, and the system updated the inventory automatically."

Table 33: UAT Client Functionality Feedback #5

QUESTION	"Were you able to understand and use the colour-coded expiry status indicators (Expired, Critical, Warning, Good)?"
ANSWER	"Yes, the colour coding is very helpful. Red means the product has expired, orange/red means it is expiring very soon (critical), yellow means it is approaching expiry (warning), and green means it is still safe. It makes it easy to know which products to prioritize without needing to remember everything manually."

Table 34: UAT Client Functionality Feedback #6

QUESTION	"Were you able to view expiry notifications and understand what action to take?"
ANSWER	"Yes. The notifications clearly stated the product name, batch number, and how many days were left. I knew immediately which products needed attention."

Table 35: UAT Client Functionality Feedback #7

QUESTION	"Were you able to access and use the system on a mobile device while working on the shop floor?"
ANSWER	"Yes. The system adjusted well to my phone screen. I could check batch details and update quantities while assisting customers without going back to the counter."

8.5.4.3 Testing and Result - Functionality Feedback (Admin Side)

The following questions below assess the desktop administrative capabilities of EXPI-STOCK, from product and batch management to alert configuration, managing user accounts, generating reports, and an overall comparison with the previous manual-based system.

Table 36: UAT Client Functionality Feedback #8

QUESTION	"Does the system allow you to add new products and create product batches with ease?"
ANSWER	"Yes. The product and batch creation forms are simple to fill in. All the fields I need, such as product name, category, expiry date, quantity, and supplier details, are clearly labelled."

Table 37: UAT Client Functionality Feedback #9

QUESTION	"Does the system allow you to configure expiry alert thresholds according to your business needs?"
ANSWER	"Yes. I can set alerts at 30, 7, and 3 days before expiry. This gives me enough advance notice to plan promotions or arrange for returns."

Table 38: UAT Client Functionality Feedback #10

QUESTION	"Does the system allow you to generate inventory reports?"
ANSWER	"The system provides me with both a Product Inventory report and an Expiry Status report. It includes all the details I need from the product name, category, total batches and total quantity, as well as which batches are expired, critical or warning."

Table 39: UAT Client Functionality Feedback #11

QUESTION	"Does the system allow you to manage user accounts for your staff?"
ANSWER	"Yes. I can add new staff accounts, assign their role, and delete accounts when needed. The system is straightforward to manage."

Table 40: UAT Client Functionality Feedback #12

QUESTION	"Does the system manage inventory expiry more effectively compared to the previous manual method?"
ANSWER	"Definitely yes. Previously we used Excel spreadsheets and missed many products approaching expiry each month. With EXPI-STOCK, the automated alerts and colour-coded status display ensure nothing is missed, and the entire process takes far less time than before."

8.5.4.4 Testing and Result - Usability Feedback

The following questions assess the ease of use, interface design, and overall user experience of the EXPI-STOCK system from the client's perspective.

Table 41: UAT Client Usability Feedback #1

QUESTION	"Is the system interface easy to navigate and understand without extensive training?"
ANSWER	"Yes. The interface is clean and well-organised. Even staff members who are less familiar with technology found it easy to use after a short walkthrough. The colour-coded expiry status in particular requires no explanation once you see it."

Table 42: UAT Client Usability Feedback #2

QUESTION	"How would you rate the overall interface design of the EXPI-STOCK system?"
ANSWER	"The design is professional, clean, and visually consistent. The use of colours to indicate urgency levels is very practical and reduces the cognitive load on staff during busy periods."

8.5.4.5 Testing and Result - Performance Feedback

The following questions evaluate the system's responsiveness, loading speed, and stability during the testing session.

Table 43: UAT Client Performance Feedback #1

QUESTION	"Does the system respond quickly when performing actions such as loading the dashboard, searching for products, or updating batch quantities?"
ANSWER	"Yes. The system responds very quickly. Pages load within seconds and there is no noticeable lag even when filtering through a large number of products."

Table 44: UAT Client Performance Feedback #2

QUESTION	"Did the system experience any crashes, errors, or delays during the testing session?"
ANSWER	"No. The system performed stably throughout the entire testing session. I did not encounter any errors, page crashes, or unexpected delays."

8.5.4.6 Testing and Result - General Feedback

The following questions gather the client's overall impression and suggestions for future improvement of the EXPI-STOCK system.

Table 45: UAT Client General Feedback #1

QUESTION	"How would you rate your overall experience using the EXPI-STOCK system?"
ANSWER	"My overall experience has been very positive. The system is user-friendly, practical, and directly addresses the inventory challenges we face daily at Wardah Baiduri. The automated expiry notifications and colour-coded status display are particularly valuable features that will significantly reduce the time and effort spent on manual inventory tracking and prevent financial losses from expired products."

8.5.6 End User Survey

This section presents the results of the end-user survey conducted to evaluate the usability and functionality of the EXPI-STOCK system. The survey was distributed to 20 respondents comprising staff and administrators at Wardah Baiduri Health and Beauty Product Shop. Responses were collected using structured questions with Yes/No and Likert-scale options to assess system accessibility, ease of use, and overall satisfaction.

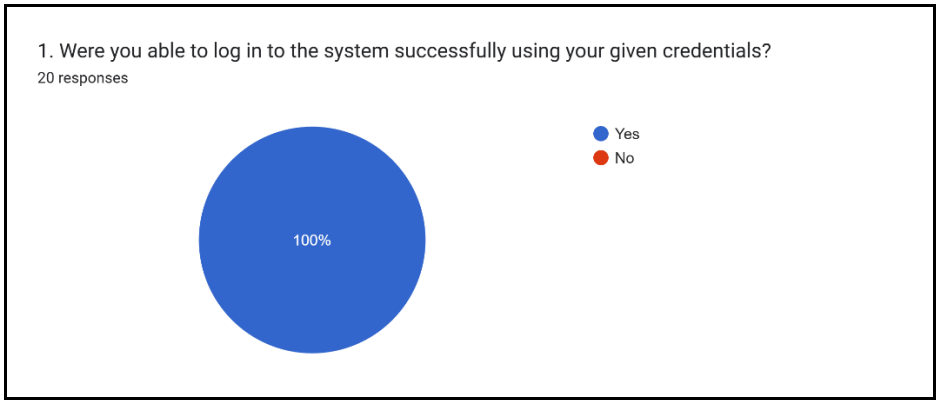


Figure 91: Question 1

Figure 91 shows a unanimous result, where 100% of respondents answered "Yes" to whether they were able to log in to the EXPI-STOCK system successfully using their given credentials. This indicates that the login flow, credential validation, and session redirection work reliably for all users. It confirms that the authentication module is stable and requires no corrective action prior to deployment.

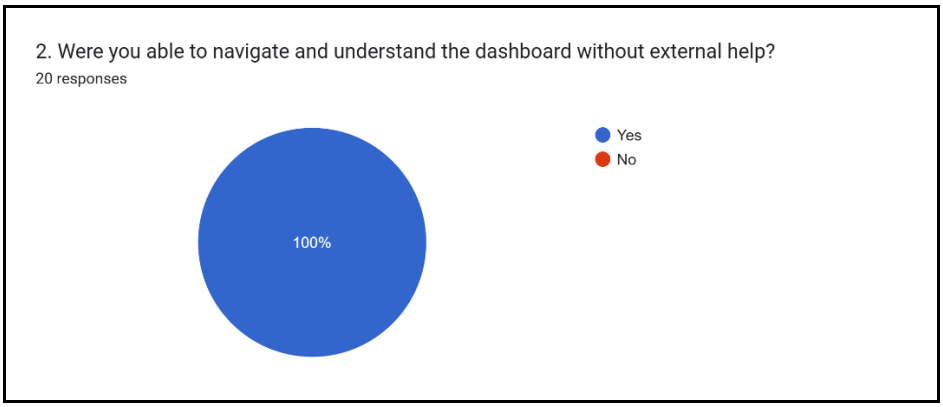


Figure 92: Question 2

Figure 92 shows that 100% of respondents answered "Yes" when asked whether they were able to navigate and understand the dashboard without external help. This result demonstrates that the dashboard layout, menu structure, and summary widgets are intuitive enough for users to orient themselves without guidance, reducing the need for extensive training.

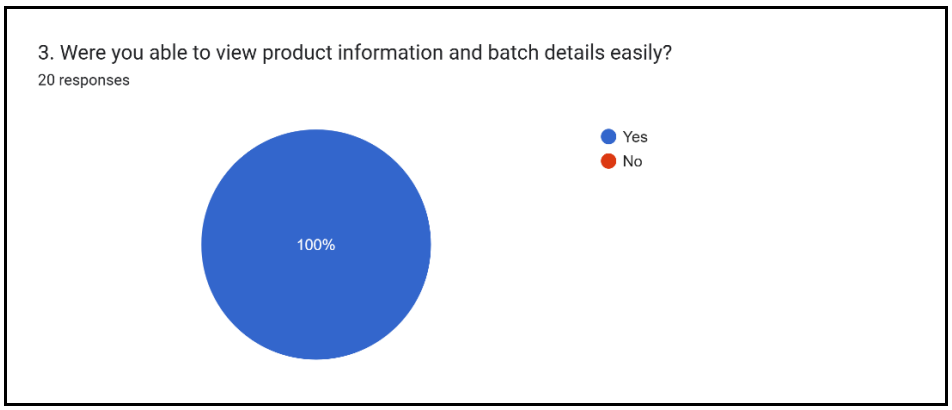


Figure 93: Question 3

Figure 93 shows that all 100% of respondents confirmed they were able to view product information and batch details easily. This indicates that the product listing and batch detail pages are well-structured, with all necessary information such as batch number, quantity, expiry date, and days remaining clearly presented and accessible to users.

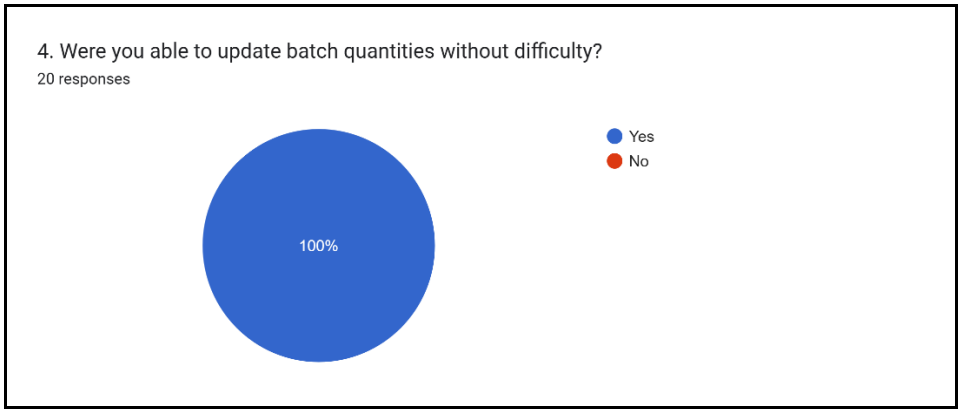


Figure 94: Question 4

Figure 94 shows a unanimous 100% "Yes" response to whether respondents were able to update batch quantities without difficulty. This confirms that the batch editing workflow is straightforward and error-free, allowing both staff and administrators to perform inventory updates efficiently during daily operations.

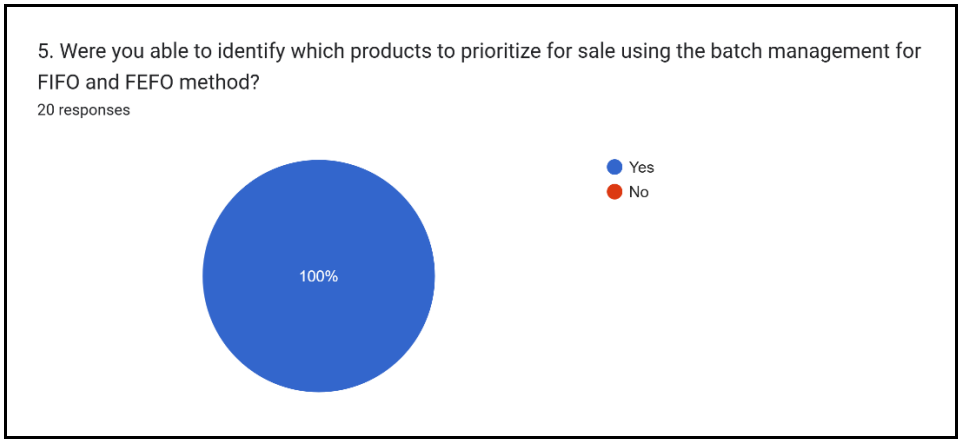


Figure 95: Question 5

Figure 95 shows that 100% of respondents answered "Yes" when asked whether they were able to identify which products to prioritize for sale using the FIFO and FEFO batch management method. This result confirms that the priority display in EXPI-STOCK clearly communicates which batches should be sold first, supporting better stock rotation and reducing the risk of product wastage.

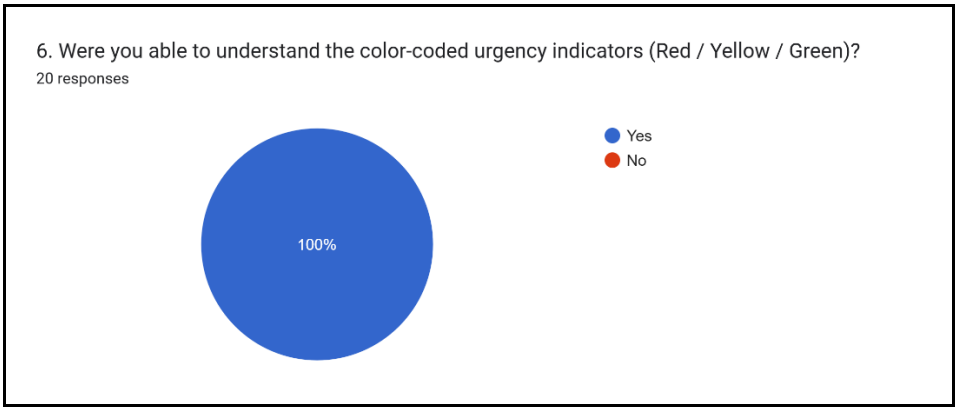


Figure 96: Question 6

Figure 96 shows that 100% of respondents answered "Yes" when asked whether they were able to understand the colour-coded urgency indicators (Red / Yellow / Green). This confirms that the colour-coding system for expiry status distinguishing between Expired, Critical, Warning, and Good batches is immediately clear to users without requiring additional explanation, which reduces cognitive load during busy working hours.

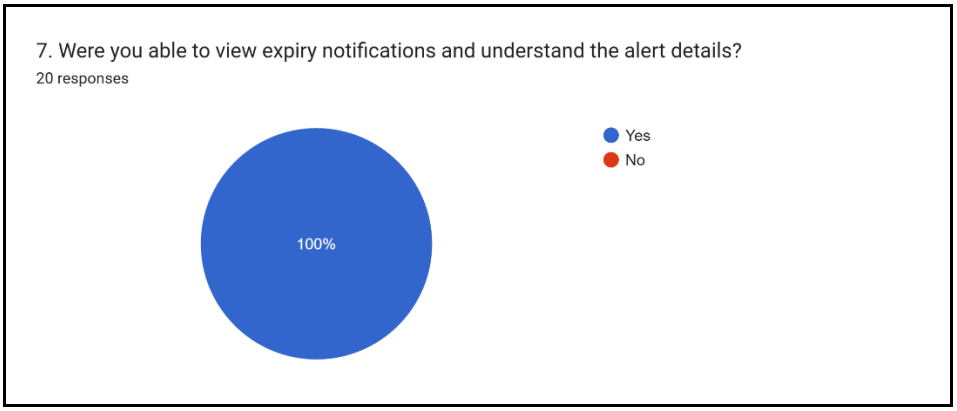


Figure 97: Question 7

Figure 97 shows a unanimous 100% "Yes" result for whether respondents were able to view expiry notifications and understand the alert details. This indicates that the notification centre successfully communicates the product name, batch reference, and days remaining in a clear and actionable format, enabling users to respond promptly to near-expiry stock.

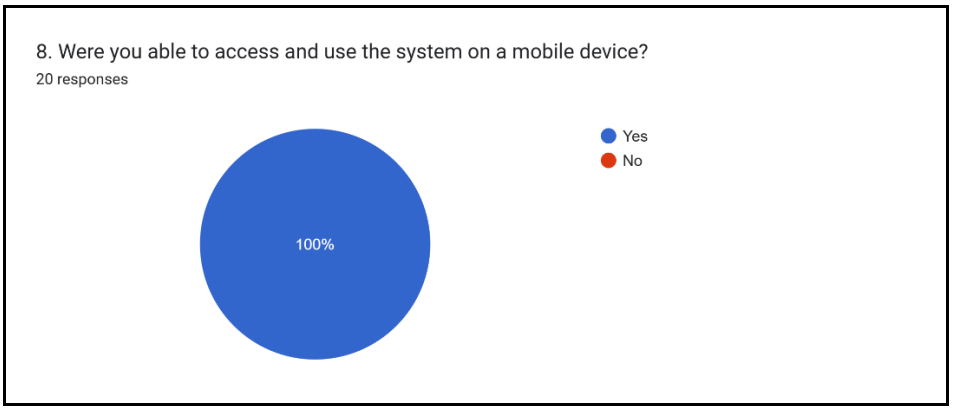


Figure 98: Question 8

Figure 98 shows that all 100% of respondents confirmed they were able to access and use the EXPI-STOCK system on a mobile device. This result indicates that the system's responsive design performs well across different screen sizes, allowing staff to check batch details and manage inventory directly from the shop floor without returning to a desktop workstation.

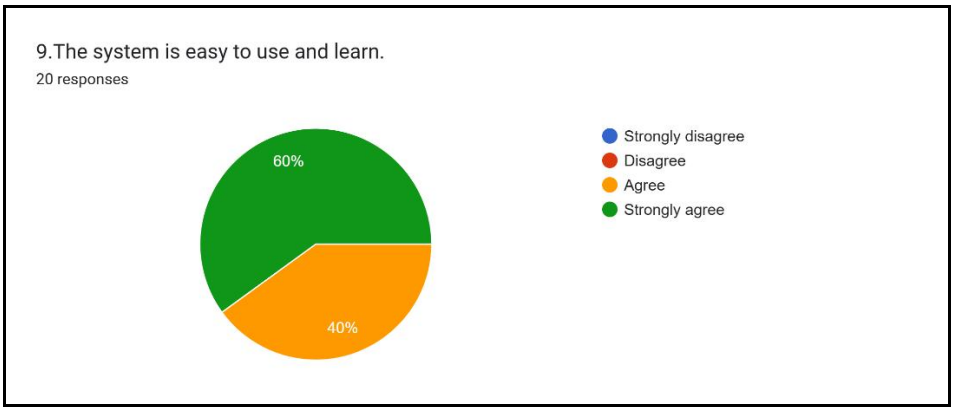


Figure 99: Question 9

Figure 99 shows a strongly positive response regarding ease of use, with 60% of respondents selecting "Strongly Agree" and 40% selecting "Agree" that the system is easy to use and learn. No respondents disagreed. This indicates that EXPI-STOCK has a low learning curve and can be adopted comfortably by staff with varying levels of technical familiarity, confirming that the interface design supports effective onboarding.

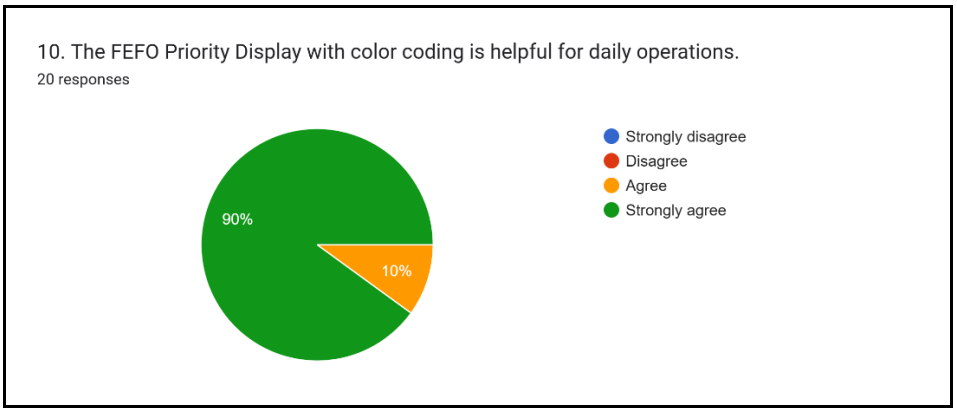


Figure 100: Question 10

Figure 100 shows very strong agreement on the usefulness of the FEFO Priority Display, with 90% of respondents selecting "Strongly Agree" and 10% selecting "Agree" that the colour-coded FEFO display is helpful for daily operations. No respondents disagreed. This result reinforces that the visual priority system is one of EXPI-STOCK's most valued features, directly supporting staff in making fast, informed decisions about stock handling.

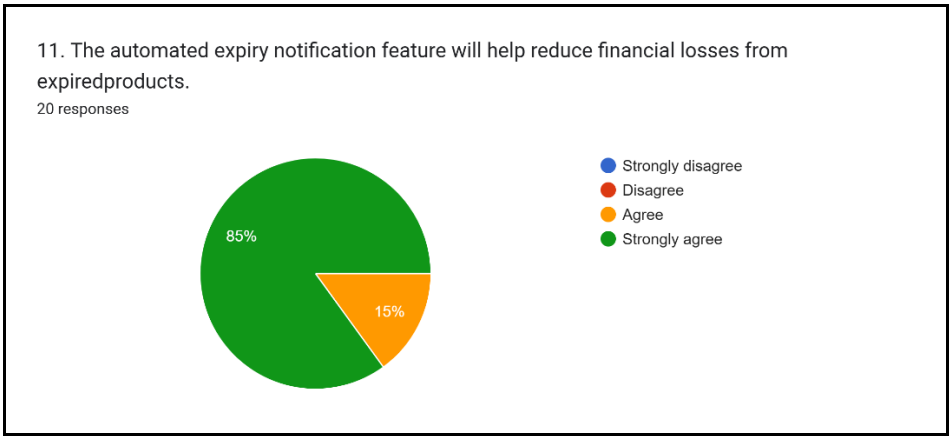


Figure 101: Question 11

Figure 101 shows that respondents strongly believe the automated expiry notification feature will help reduce financial losses from expired products, with 85% selecting "Strongly Agree" and 15% selecting "Agree". No respondents expressed doubt or disagreement. This reflects high confidence in the practical business value of the automated alert system, which was also highlighted during client testing as a major improvement over the previous manual tracking method.

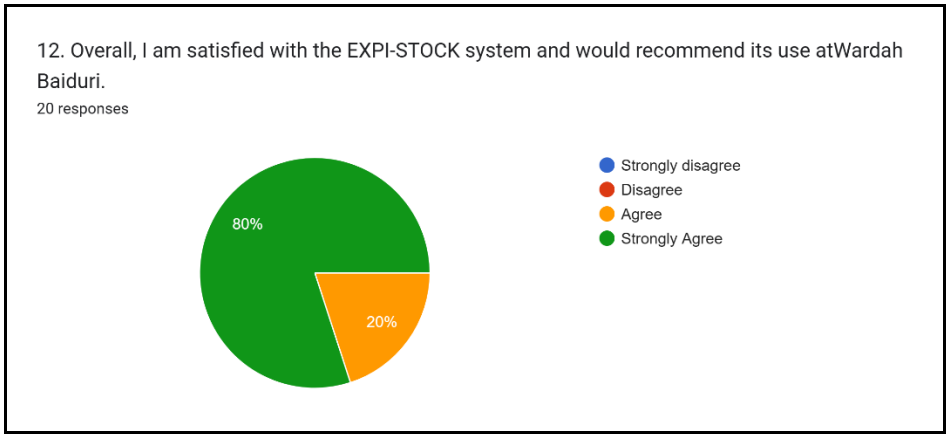


Figure 102: Question 12

Figure 102 shows that 80% of respondents "Strongly Agree" and 20% "Agree" that they are overall satisfied with EXPI-STOCK and would recommend its use at Wardah Baiduri. No respondents expressed dissatisfaction. This result reflects strong overall acceptance of the system across all tested dimensions functionality, usability, and performance and confirms that EXPI-STOCK is ready for full operational deployment at the shop.

8.6 Conclusion

The focus of this chapter is on some types of software testing such as unit testing, integration test, user acceptance test and client. This method is essential in order to assess system readiness throughout the early-stage development and testing. In unit testing, each individual function and module of EXPI-STOCK was validated in isolation, ensuring that all core operations such as login, batch management, report generation, alert configuration, and user account management work correctly for both Administrator and Staff roles.

Integration testing ensured that each of the modules within the system work correctly with one another as a whole integrated solution, where data flows correctly between authentication, batch management, notification scheduler, FIFO & FEFO display, audit logging and role-based access control features. System testing during UAT only confirmed that all 12 test scenarios were executed successfully across both user roles, without any high severity defects.

Meanwhile, user acceptance testing became the one of the most critical testing phases as it defined whether EXPI-STOCK met end user requirement defined earlier. The client testing supported this result, yielding positive feedback in every function, usability, and performance category from the primary stakeholder of Wardah Baiduri Health and Beauty Product Shop. The Auto expiry alerts and colour-coded

status display were specifically highlighted by the client as a considerable enhancement over the previously adopted approach of manually tracking inventory through Excel spreadsheets. Additional features like barcode scanning support and the ability to directly download reports from the dashboard has been earmarked for future development cycles.

9 PROJECT MANAGEMENT

9.1 Introduction

In this chapter we will describe the project management methods that were used throughout the development process of EXPI-STOCK (Expiry Stock Inventory Management System) for Wardah Baiduri Health & Beauty Product Shop Project management is defined as the application of knowledge, skills, tools and techniques to project activities to meet project requirements ((Project Management Institute, n.d.)). Rigorous planning, scheduling and risk management aspects are very important in software development projects so that scope, schedule, quality constraint can be consistently met.

Project Management activities for EXPI-STOCK entailed planning across both FYP 1 and FYP 2, monitoring developmental progress through a Work Breakdown Structure (WBS) and Gantt chart along with assessing risks relevant to technical implementation, schedule adherence and data security. In this chapter, the project schedule, WBS, Gantt chart and risk management plan applied to make sure that EXPI-STOCK was cooked-and-served successfully.

9.2 Project Schedule

The project schedule outlines the sequence of activities that were carried out during the process of developing EXPI-STOCK. A schedule helps the project manager to see what has to be done, when it needs to happen and if tasks depend on each other (ProjectManager. com, 2025). FYP 1 (Final Year Project Part I) on background study, requirements gathering and system design Chapter 1 - 5. FYP 2 included system implementation, testing, deployment and finalisation of the report, both FYPs running from 4 August 2025 to 8 April 2026. The task was planned as per requirements with core functions like batch tracking, expiry alerts and FIFO & FEFO priority display being highest priorities to allow for all major milestones like prototype demonstration, user acceptance testing (UAT) and final submission.

9.2.1 Work Breakdown Structure (WBS)

A Work Breakdown Structure (WBS) is a hierarchically-oriented and deliverable decomposition of all the work to be performed by the project team ((Project Management Institute, 2021); (ProjectManager.com, 2022)). It divides a large project into smaller, easier-to-manage parts so that planning and control become less complex ((Institute of Project Management, 2020)). The specific WBS for EXPI-STOCK was broken down into five key phases, as shown in Figure 9.2.1 below.

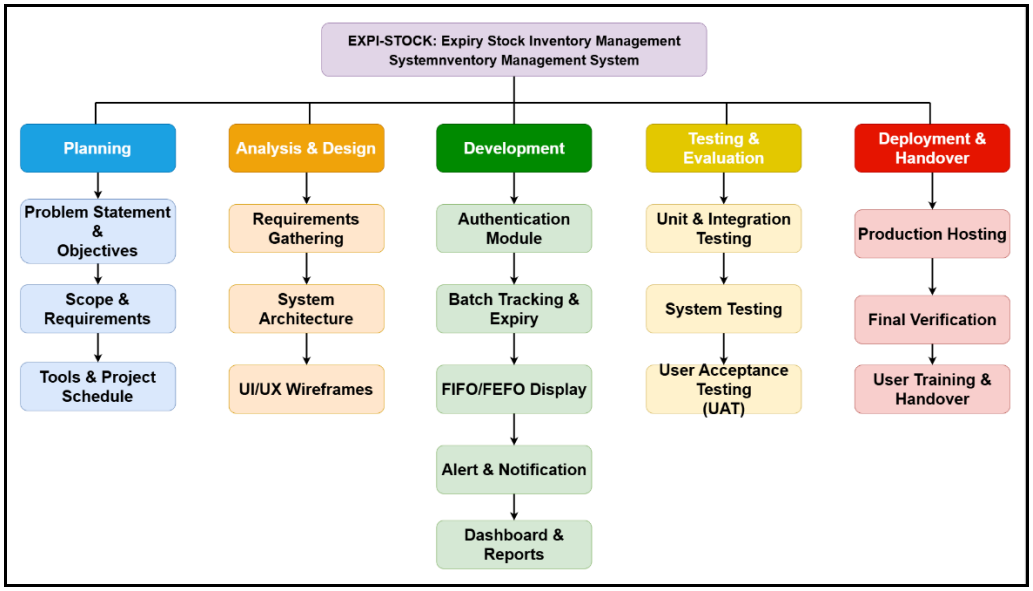


Figure 103: Work Breakdown Structure for EXPI-STOCK System

1)Planning

The planning phase is the make-up of the EXPI-STOCK project by stating the problem that arises in Wardah Baiduri, system objectives which would be clearly defined and a list of tools and timelines that will run through the whole development. Developing a bulletproof shop environment falls into this phase where the problem statement and system objectives are formalized to ensure that all development activities will continue to be tethered to operational aspects of the shop. At the same time, a scope and requirements overview is created that describes which features are included both within the boundary of the system and what falls outside of it. A standard tool used in development that includes use of HTML, CSS, JavaScript and PHP for the system, Firebase for database and hosting infrastructure, Figma for UI/UX design; overall project timeline is discussed.

2) Analysis & Design

The analysis and design process turns that overall idea into a solid plan to build the intended system. Wardah Baiduri's needs are obtained through organized discussions with the shop administrator and survey questionaries delivered to all staff, in order to capture both administration and methods. The system architecture and database schema are designed based on the gathered requirements, taking into account all main entities including products, batches, users, notifications and audit logs. Alongside, the design (UI/UX) wireframes for Admin and Staff using Figma Files attached handover after approval are built simultaneously to ensure that finally when we build this system, users will not need a visa engineer or advanced programming skills.

3) Development

EXPI-STOCK is developed at this phase following the approved design specifications. Write backend and frontend components, gradually combining them. And the security user authentication module is implemented at first to control access according to role for administrator and staff. Next in the list are product management and batch tracking modules, which is what mostly consist in an inventory system. Next, the FIFO/FEFO priority display is created with colour-coded urgency indicators; red for critical items expiring in seven days or less, yellow for items reaching expiration by thirty days and green for safe stock, to assist staff with important decisions on stock rotation. Even the automated alert and notification system using SMTP email integration is designed with configurable expiry thresholds. The admin dashboard, reporting module and mobile-responsive staff interface are ultimately built out and integrated to bring the system full circle.

4) Testing & Evaluation

The testing and evaluation phase guarantees that EXPI-STOCK will function properly, be stable and meet user expectations prior to deployment. Unit tests are performed for per-module validation such as to verify each function is working correctly in isolation including batch create routines, expiry date calculations and alert generation logic. Integration testing verifies that all modules talk correctly with each other when composed, as far as the flow is concerned from batch entry to leader delivery. System testing involves both functional requirements, including CRUD operations and FIFO sorting and notification accuracy, and non-functional requirements such as performance under concurrent users, browser compatibility and security controls. User Acceptance Test (UAT) is performed with real Wardah Baiduri employees using realistic operational scenarios to validate that the system meets practical business needs and is ready for deployment.

5) Deployment & Handover

This phase goes from a development environment in which we develop EXPI-STOCK into live operational use at Wardah Baiduri. The system is deployed to the web hosting server, where MySQL and phpMyAdmin as database managing system are configured along with SSL and HTTPS for secure access. We perform a final verification and smoke test to ensure that all live features such as tracking by batch, expiry alerts and FIFO priority display are working in production. Wardah Baiduri staff attend user training sessions to help them get acquainted with the system, which is why we also deliver a detailed user guide and training materials for future use. The project documentation which contains the FYP report is also finished and given to the supervisor and evaluation panel.

9.2.2 Gantt Chart

A Gantt chart is a diagrammatic tool for visual management of projects, which shows project activities as bars in the timeline to provide an at-a-glance understanding of when activities start and finish, their duration and overlaps between tasks ((ProjectManager. com, 2025); (TeamGantt, 2025)). The Gantt chart for EXPI-STOCK throughout the FYP 2 period of 4 August 2025 to 8 April 2026 is shown in Figure 9.2.2, structured by weekly intervals. As seen in Figure 9.2.3, the chart is segmented into five colour coded phases; Planning & Requirements (blue), Implementation (orange), Testing & QA (green), Deployment (yellow) and Documentation & Finalisation (brown).

The planning phase occupies the first three weeks of August 2025. Implementation is the longest phase, starting August but running through December 2025 due to complexity of FIFO/FEFO algorithm and SMTP notification integration. Testing takes place from Dec. 2025 to Feb. 2026. The final presentation is planned for 30 March 2026 while the final submission would be conducted on 8 April 2026, with deployment and documentation activities from March to April 2026.

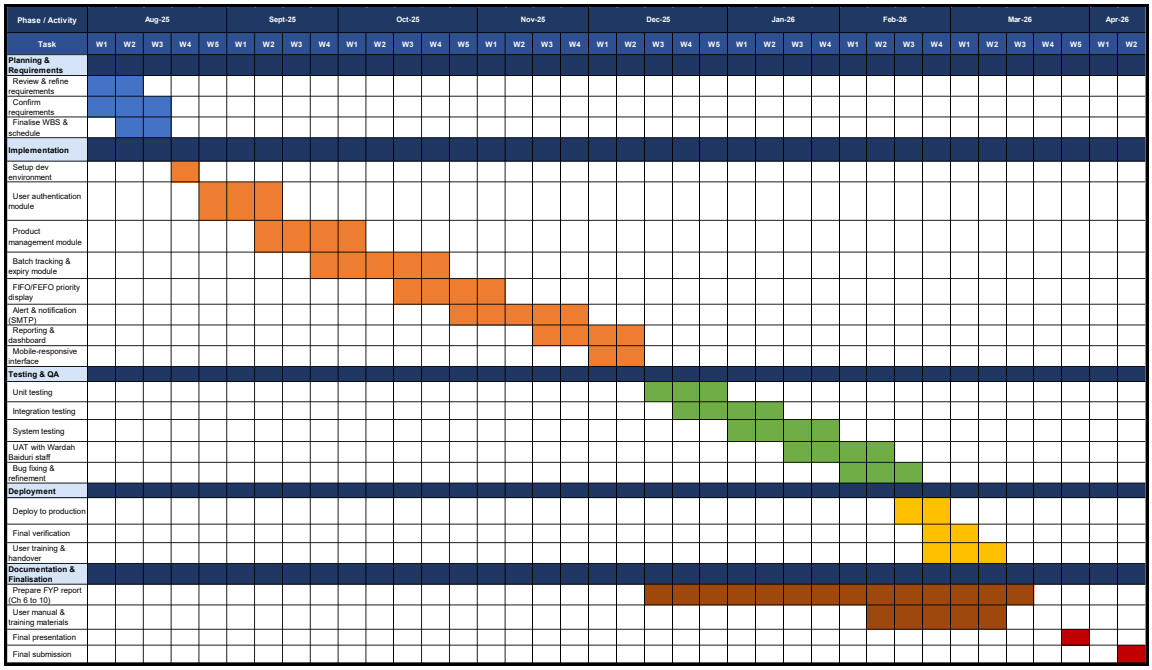


Figure 104: Gantt Chart for EXPI-STOCK Project Timeline

Figure 105: Gantt Chart Colour Legend

9.2.3 Project Schedule Timetable

The following Table 9.1 introduces the detailed project timetable for EXPI-STOCK Project based on each phase and activity with respective start date, end date and duration. A schedule timetable helps tracking progress, making sure that the all stages are being delivered on time (ProjectManager. com, 2025). Note that summary rows shaded blue represent central phase iterations and indented rows show individual tasks in each phase.

Table 46: Project Schedule Timetable for EXPI-STOCK (FYP 2)

No.	Phase / Activity	Start Date	End Date	Duration
1	**Planning & Requirements**	**4 Aug 2025**	**22 Aug 2025**	**3 weeks**
2	Review FYP1 requirements & refine scope	4 Aug 2025	15 Aug 2025	2 weeks
3	Confirm requirements with supervisor	4 Aug 2025	22 Aug 2025	3 weeks
4	Finalise project schedule & WBS	11 Aug 2025	22 Aug 2025	2 weeks

5	**Implementation**	**25 Aug 2025**	**19 Dec 2025**	**17 weeks**
6	Setup development environment	25 Aug 2025	1 Sep 2025	1 week
7	Develop user authentication module	1 Sep 2025	15 Sep 2025	2 weeks
8	Implement product management module	15 Sep 2025	6 Oct 2025	3 weeks
9	Implement batch tracking & expiry module	29 Sep 2025	27 Oct 2025	4 weeks
10	Develop FIFO/FEFO priority display	20 Oct 2025	10 Nov 2025	3 weeks
11	Develop automated alert & notification (SMTP)	3 Nov 2025	1 Dec 2025	4 weeks
12	Implement reporting & dashboard analytics	24 Nov 2025	15 Dec 2025	3 weeks
13	Implement mobile-responsive interface	8 Dec 2025	19 Dec 2025	2 weeks
14	**Testing & Quality Assurance**	**22 Dec 2025**	**23 Feb 2026**	**9 weeks**
15	Unit testing of individual modules	22 Dec 2025	12 Jan 2026	3 weeks
16	Integration testing across all modules	5 Jan 2026	26 Jan 2026	3 weeks
17	System testing (functional & non-functional)	19 Jan 2026	9 Feb 2026	3 weeks
18	User Acceptance Testing (UAT)	2 Feb 2026	23 Feb 2026	3 weeks
19	Bug fixing and system refinement	16 Feb 2026	23 Feb 2026	1 week
20	**Deployment**	**2 Mar 2026**	**16 Mar 2026**	**2 weeks**
21	Deploy to production server (Firebase)	2 Mar 2026	9 Mar 2026	1 week
22	Final verification & smoke testing	9 Mar 2026	16 Mar 2026	1 week
23	User training & handover	9 Mar 2026	23 Mar 2026	2 weeks
24	**Documentation & Finalisation**	**22 Dec 2025**	**8 Apr 2026**	**15 weeks**
25	Prepare FYP report (Chapters 6 to 10)	22 Dec 2025	23 Mar 2026	13 weeks
26	Prepare user manual & training materials	23 Feb 2026	23 Mar 2026	4 weeks

27	Final presentation	30 Mar 2026	30 Mar 2026	1 day
28	Final submission (report, slides, system)	8 Apr 2026	8 Apr 2026	1 day

The first few weeks were dedicated to planning and reviewing requirements (see Table 9.1). The timeframe from August to December 2025 was employed for implementation, with concurrent tasks designed to optimise efficient use of time. Testing and quality assurance occurred from December 2025 until February 2026. Deployment, documentation and the final presentation are for March to April 2026 with a finalised date of 30 March 2026 and submission due on 8 April 2026.

9.3 Risk Management

Risk management is the most important, critical process of identifying and evaluating all potential interactions that can impact the success of a software product or project (McGuire, 2024). While EXPI-STOCK was being developed as a web-based system there were certain risks associated with it like technical risk, scheduling risk and data loss risk during or after development.

Table 47: Risk Management Table

No	Risk Identification	Description	Risk Analysis	Mitigation Plan
1.	**Technical Risk**	Includes compatibility issues between the frontend (HTML/CSS/JavaScript) and backend (PHP/Firebase) components, as well as potential failures in SMTP email notification delivery during system operation.	Moderate	Perform early testing and debugging of all modules during the development phase. Configure a backup notification channel using in-system dashboard alerts in case SMTP service fails.
2.	**Scheduling Risk**	Includes time constraints or delays due to underestimating the complexity of the FIFO/FEFO algorithm and batch tracking module development. The system may not be completed within the expected time frame.	High	Initiate weekly consultation with the project supervisor. Break implementation into smaller incremental tasks and allocate time buffers for complex modules. Prioritise core features if schedule is at risk.

3.	**Data Loss Risk**	Can occur due to system crashing, accidental deletion of database records, or Firebase service disruption during the development or operational phase.	High	Keep a backup copy of all project files and database records. Perform regular system backups and use version control (Git/GitHub) throughout development. Enable automated daily backups on Firebase.

Despite careful planning, risks related to scheduling and technical integration were the most challenging during development. Thus, implementing the FIFO/FEFO expiry sorting algorithm and configuring SMTP email notification system took longer time than expected. These challenges were tackled by splitting tasks into smaller increments, prioritising core features and consulting regularly with the project supervisor. Using Git version tracking prevented the permanent loss of any important code or data during development.

9.4 Conclusion

The project management framework used for developing EXPI-STOCK is detailed in this chapter. The WBS artefact established a top-down approach to systematically break down project work into one of five phases. Also, the Gantt chart showed how tasks were allocated to weeks between 4 August 2025 and 8 April 2026 after which the project schedule timetable transformed the planning into a physical activity calendar with start dates and end dates.

The risk management element delineated the three main categories of risk: technical, timing and data loss, as well as plans for mitigation. Tools such as the WBS and Gantt chart clarified these tasks to ensure that project delivery was tracked according to the plan. With analysis of risk management, possible problems were evaluated and measures to prevent them from becoming a critical concern but the completion of the project on time.

10 CONCLUSION

10.1 Introduction

This final chapter is the conclusion of EXPI-STOCK Expiry Stock Inventory Management System development at Wardah Baiduri Health and Beauty Product Shop, which explains the overall result of this system Development, problems encountered during the system development process and improvements that can be made in the future. The system was built out to address three main issues identified in Chapter 1, namely a lack of an organized mechanized approach toward digital inventory tracking, consistent losses incurred by expired products with no relevant early warning systems and needless complexities around stock management as a consequence of the difficulty around manual implementations of First In, First Out (FIFO) methods (ToolsGroup, 2024; CYBRA, 2024; Lightspeed, 2025). In this chapter, this is shown by delivering web-based application functionality for automated batch tracking and expiry date monitoring with visual FIFO / FEFO marker prioritisation display as well as multi-channel alert notifications from EXPI-STOCK. It also outlines some of the challenges and limitations faced during development, as well as suggestions for improving the system's performance, usability, and scalability in future iterations.

10.2 Achievement

EXPI-STOCK has addressed its fundamental aims by supplying a web-based inventory tracking software program which automates expiry dates monitoring; encompasses colour-coded FIFO and FEFO precedence displays; can provide pre-emptive multi-stage alert notifications to the administrator and customers. It integrates product management, batch tracking, expiry tracking, notification notifications and reports to provide Wardah Baiduri with a single accessible system (MRPeasy, 2024). This allows them to replace their dated manual Excel-based practices which incur financial losses through expired stocks and have better visualization on the operations. The user response grades for functional testing, system testing and end-user surveys presented in previous chapters indicate that the users found the system to be intuitive, efficient and significantly reduced their manual workload which indicates that at a high level the project objectives have been met.

10.2.1 To Develop a Web-Based System That Tracks the Expiry Date of Each Product

(Achieved) This was done by building a responsive web-based inventory management application which allows users to systematically track their product expiry date in real time. The system then provides multiple layered level of expiry visibility across the platform as seen in figure 106, figure 107, and figure 108. As depicted in Figure 106, the Admin Dashboard provides a high-level view of total batches with summary cards reflecting the count of Expired, Critical and Warning batches at a glance and an Upcoming Expiry Priority table lists products requiring immediate action by rank order of urgency. An adjunct to this in Figure 107 is a pie chart of Batch Status Distribution and a Summary by Product table which together provide management with an analytical picture through all product categories as of inventory health at one glance level. On the day-to-day operational level, Figure 108 displays the Batch Management page containing a list of each batch with its unique batch number, product name, quantity received, received and expiry dates, days remaining until product expiration, and colour-coded status badges that easily differentiate Critical from Warning/Good/Expired conditions.

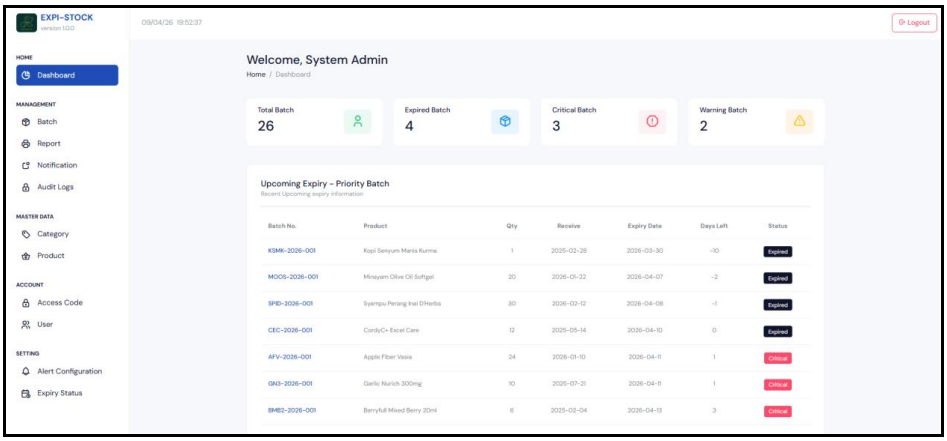


Figure 106: Admin Dashboard Part 1

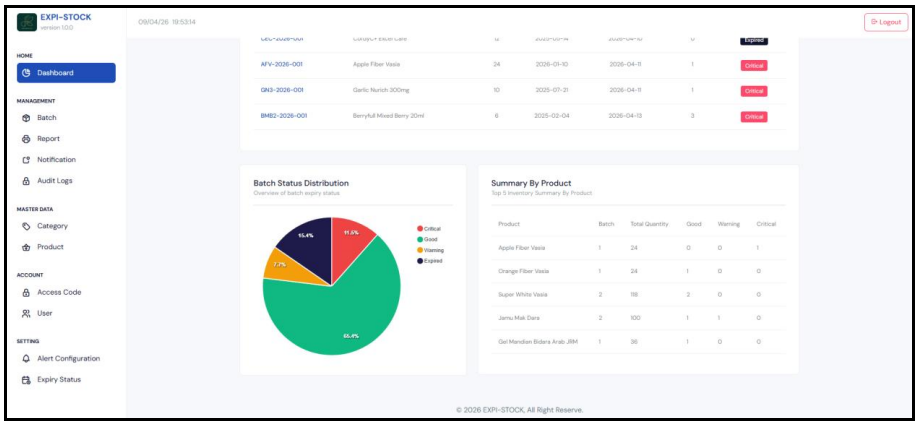


Figure 107: Admin Dashboard Part 2

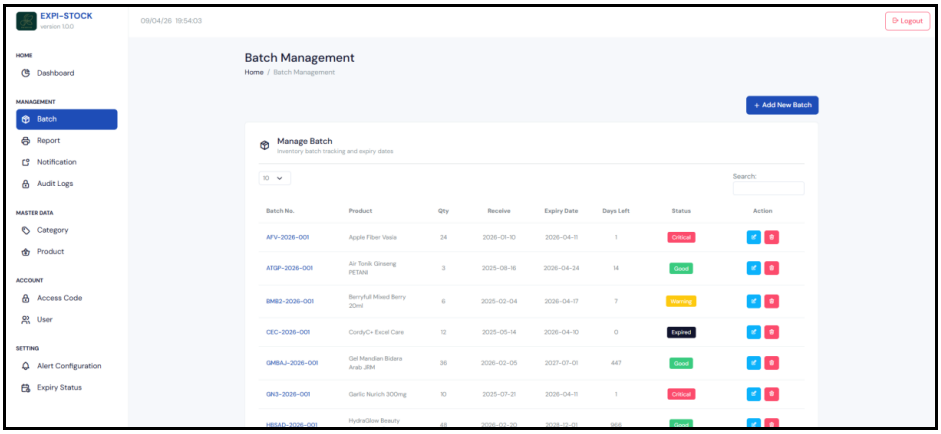


Figure 108: Batch Management Page

10.2.2 To Implement Automated Web-Based Reminders Before Products Expire

(Achieved) This objective was attained by the integration of a centralized multi-stage automated alert notification system into the web application. To illustrate the process, Figures 109 and 110 show how expiry warnings are pushed out through two different channels by the system when a product batch is at risk of expiring at 30-7-3 day intervals. We further developed the Notification and Expiry Alert page in system (Figure 109) listing all current alerts by product name, batch number, expiry date, and days-left-to-action such that once a user logs in staff immediately knows which products require action. As an added measure to make sure no alert is missed, each entry has a Mark as Read button. As shown by Figure 110, it automatically sends an email notification to the user inbox using Simple Mail Transfer Protocol (SMTP) integration with product name, batch number, expiry date and remaining days along with a direct Check Batch link for taking action immediately by redirecting to the respective batch record. Utilizing both channels simultaneously guarantees that personnel will always receive alerts, whether or not they are logged into the system at the time of alert (CYBRA, 2024), thus addressing the issue of expired products being found only during routine manual checks.

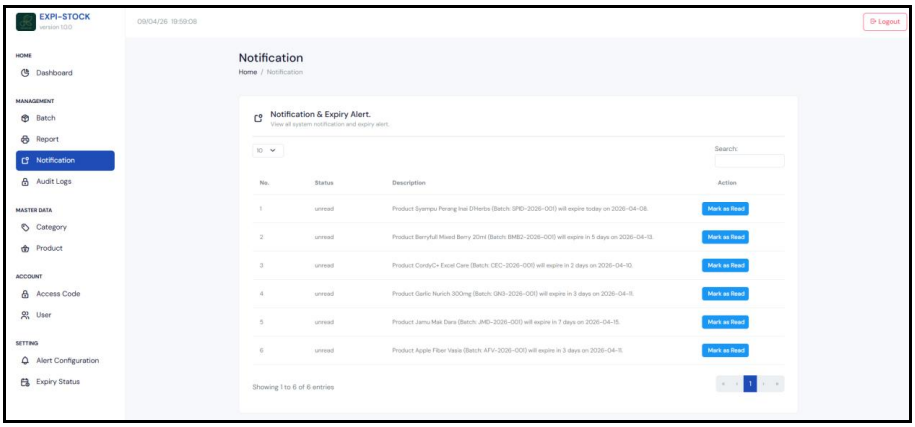


Figure 109: Notification Page

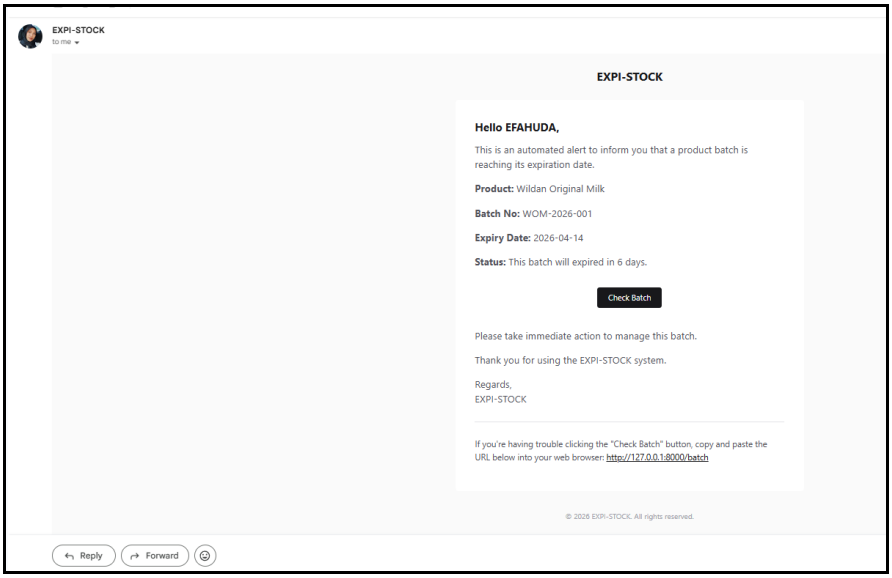


Figure 110: Automated Email Notification Sent by EXPI-STOCK for Approaching Product Expiry

10.2.3 To Identify FIFO Priority Products

(Achieved) The Batch Management interface allowed us to achieve this goal, as shown in Figure 111. The system captures the Receive date and Expiry Date for each batch, and based on FIFO and FEFO prioritisation methodologies, the platform automatically sets up to retrieve, at our stock rotation sequence. Batches are sorted and displayed (ascending expiry date), guaranteeing that the nearest-expiry products are always visible at the top of the list regardless of when received. The colour coded status badges: Critical in red, Warning in yellow, Good in green and Expired in black instantly give staff at a glance guidance on which products need to be sold first. This directly addresses the issue where junior and part-time staff often did not have the institutional knowledge to rotate stock appropriately during customer-facing roles, democratising this crucial operational information across all levels of experience (Lightspeed 2025; Addverb 2025).

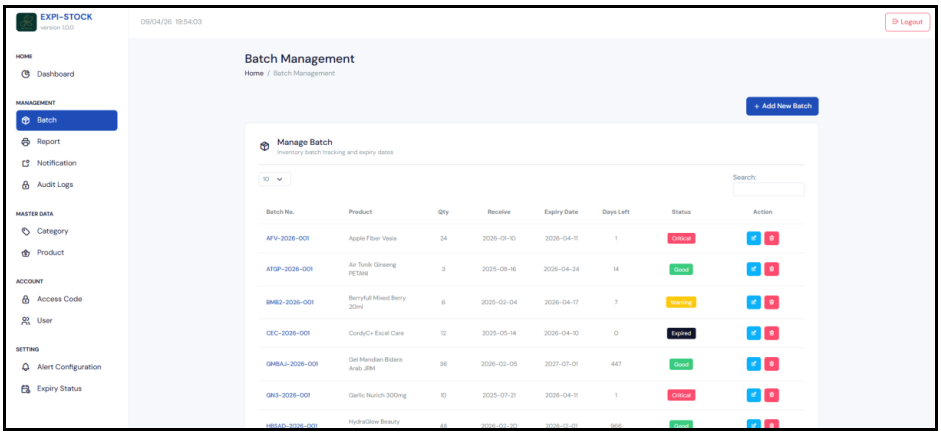


Figure 111: Batch Management Page

10.3 Constraint and Limitation

A few limitations were found during the development of EXPI-STOCK that impacted on system scope, depth and testing breadth. The major limitation was perhaps time, since this project needed to be complete within the two FYP semesters and in balance with the other academic responsibilities. This limited the amount of iterative tuning that could be done in terms of both user interface design and performance tuning, and how much real-world field testing could realistically be planned into the project timeline.

A second key limitation was choosing Firebase Cloud Infrastructure as the backend hosting environment. Though Firebase offered huge advantages regarding rapid deployment, built-in authentication and scalable hosting, Cloud Firestore NoSQL architecture lacked access to some advanced MySQL relational features that we have grown accustomed to over the years like complex stored procedures, cascading triggers and query optimisation tools. This meant that some business logic had to live in the application layer themselves, adding more complexity to backend controllers.

Also faced a few challenges configuring the SMTP email notification service and testing it out. Support for deliverability issues with email providers and spam filter configurations continued into development, requiring extra debugging as well as logs during testing which delayed the notifications phase. Additionally, the system presently depends on a single SMTP service provider which would cause users not to receive expiry notifications in the event of downtime or misconfiguration.

Lastly, the system evaluation was limited largely to performing tests of staff and administrators at Wardah Baiduri in controlled testing environments rather than across a fully operational multi-branch deployment. Therefore, the results may not account for every operational scenario that would occur in prolonged live usage across both shop branches. Equally, they also helped hold EXPI-STOCK to a stable and operating minimum viable system while highlighting obvious areas for refinement and opportunity.

10.4 Future Work and Recommendation

EXPI-STOCK serves as a strong operational platform that can be extended over time as the digital inventory management strategy of Wardah Baiduri develops. Below are the areas to improve in future versions of the system:

10.4.1 Multi-Branch Inventory Management

EXPI-STOCK is implemented to support the day-to-day operations of Wardah Baiduri with a single system of general ledger. The business, however, is bifurcated and each branch has separate inventories, stock levels, and expiry monitors. An idea for future improvements is to make a multi-branch inventory management functionality, which gives the administrators an option to create and manage inventory separately by branches (location), per location track expiry alerts, compare stock levels in two or more locations and transfer material between your own

10.4.2 Centralised Database Management

EXPI-STOCK currently uses MySQL as a primary database maintained with the phpMyAdmin client, which sufficiently meets functional requirements from its current scope. Although working on a single database instance may work in the initial stages, as the business grows and you start dealing with an exponential increase in product batches, transaction records, and activity logs generated by users interacting with your product; it would become critical to switch to an optimized and distributed architecture. Such use of advanced indexing and stored procedures would likely allow for better optimization of queries if new queries can be optimized with regards to time complexity. This approach would also allow collection of inventory data from different branches to be stored in a well-structured centralised database for complex queries, relational data integrity constraints and comprehensive reporting capabilities.

10.4.3 Sales and Expiry Loss Report

The existing reporting module in EXPI-STOCK allows to generate basic stock reports like product stock summary, expiry status summary and batch history report. Expanding the integration through building a Sales and Expiry Loss Report module that unifies sales transaction statistics with expiry tracking records can be a potential pivot for analytics. This improved reporting capability would allow management to measure monthly, quarterly and yearly financial impact from expired products, risk using the product categories most prone to expiry losses as a result of its different strategies such as discounting and finally evaluate the success/failure of discounting/promotion practices implemented in response to alerts generated for expiring stock. By employing data visualisation tools, like trend charts and loss heatmaps, raw inventory data would be transformed into valuable business intelligence to allow for better-informed procurement decisions and promotional planning.

10.5 Conclusion

In conclusion, the EXPI-STOCK - Expiry Stock Inventory Management System provides Wardah Baiduri Health and Beauty Product Shop with a dynamic web-based solution to extend its functionality in centralising expiry date tracking through automated alert notifications while prioritizing FIFO and FEFO. The system directly addresses each of the three problem statements identified at the beginning: automatic batch tracking removes dependence on human error and oversight typically involved in using paper or Excel spreadsheets (ToolsGroup, 2024); multi-staged notifications mean staff have proactive warnings that allow for foresight and promotional/return-to-supplier actions to avoid loss before it occurs (CYBRA, 2024); colour coded FEFO priority displays democratise critical stock rotation knowledge across all members of staff irrespective of experience (Lightspeed, 2025; Addverb, 2025).

Utilising the Waterfall methodology for the project ensured a structured approach to development, and given that inputs using questionnaires and expert opinion interviews based on systematic requirements analysis were received, efforts could be kept progressing throughout despite technical hurdles with Firebase configuration, SMTP deliverability and time constraints faced with an academic project period. In the end-user survey conducted during a site visit, Wardah Baiduri staff demonstrated high acceptance of this automated system; in fact, all respondents agreed that the new system would increase efficiency with regard to managing its inventory and [1] they calculated the savings due to fewer expired products returned was about RM15,000 per year, which should be considered a good return on development investment.

This has taught me a great deal of new skills such as full-stack web development, data-storage and retrieval using SQL databases, automated email notification systems and structured testing of each component. Tasks like configuring Firebase backend services, resolving SMTP email deliverability constraints and implementing real-time FIFO and FEFO sorting algorithms took a lot of patience, independent research as well as constant engagement with supervisory guidance. Much of the technical competency, time management, and problem-solving skills developed through these experiences translate well into future practices in software engineering.

With future enhancements such as multi-branch inventory management, a centralised database architecture, and a comprehensive Sales and Expiry Loss Report module; EXPI-STOCK will reach unparalleled heights in terms of scale, analytical depth, business value. These improvements will allow the systems to expand as Wardah Baiduri's business expands and provide long-term benefits by furthering efficiencies in inventory management, profits, and customer satisfaction.

Appendix A – Requirements Specification Document

Interview Questions (Expert Opinion)

EXPI-STOCK: Business Inventory Management System

Name of respondent:

Date:

Time:

1. Can you walk me through your current process for managing inventory and tracking product expiry dates at Wardah Baiduri?

   _____

2. What are the biggest challenges you face with the current inventory management system, particularly regarding product expiry tracking?

   _____

3. From a business perspective, can you describe the financial impact of expired products at Wardah Baiduri?

   _____

4. How do your staff currently identify which products should be prioritized for sale, especially those nearing expiry?

   _____

5. If you had an automated web-based inventory system like EXPI-STOCK, what specific features would be most critical for your business operations?

   _____

Interview Questions (Post-Development)

EXPI-STOCK: Business Inventory Management System

Name of respondent: Zurini Binti Ismail

Date: 27 March 2026

Time: 11.30 a.m. – 12.30 p.m.

1. Were you able to log in to the EXPI-STOCK system successfully using your given credentials?

   _____

2. Were you able to view the product list and search for products easily?

   _____

3. Were you able to view batch information including expiry dates and days remaining?

   _____

4. Were you able to update batch quantities after selling products?

   _____

5. Were you able to understand and use the colour-coded expiry status indicators (Expired, Critical, Warning, Good)?

   _____

6. Were you able to view expiry notifications and understand what action to take?

   _____

7. Were you able to access and use the system on a mobile device while working on the shop floor?

   _____

8. Does the system allow you to add new products and create product batches with ease? the system allow you to add new products and create product batches with ease?

_____

9. Does the system allow you to configure expiry alert thresholds according to your business needs?

_____

10. Does the system allow you to generate inventory reports?

_____

11. Does the system allow you to manage user accounts for your staff?

_____

12. Does the system manage inventory expiry more effectively compared to the previous manual method?

_____

13. Is the system interface easy to navigate and understand without extensive training?

_____

14. How would you rate the overall interface design of the EXPI-STOCK system?

_____

15. Does the system respond quickly when performing actions such as loading the dashboard, searching for products, or updating batch quantities?

_____

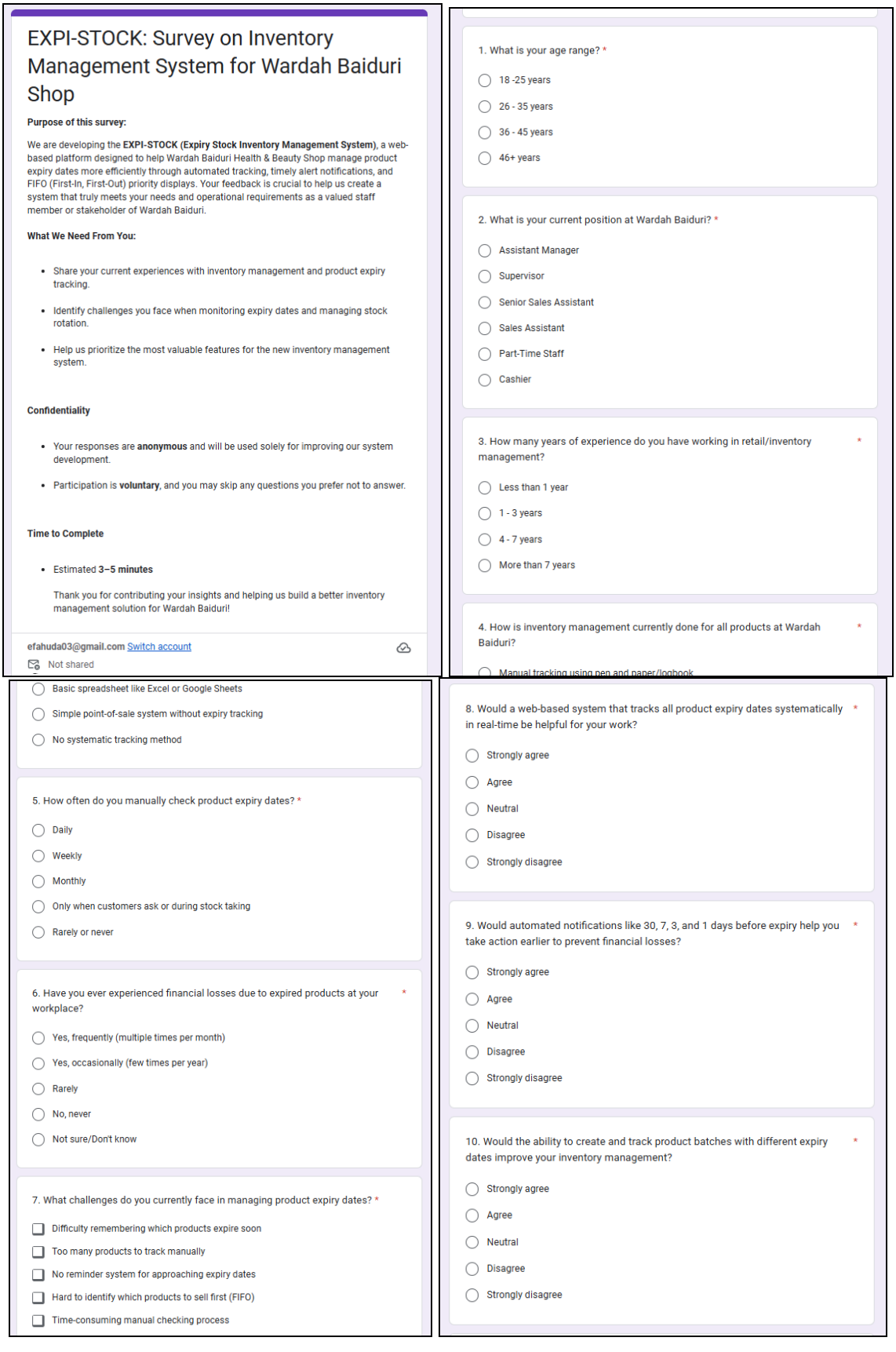
16. Did the system experience any crashes, errors, or delays during the testing session?

_____

17. How would you rate your overall experience using the EXPI-STOCK system?

_____

Questionnaire Questions



EXPI-STOCK: Survey on Inventory Management System for Wardah Baiduri Shop

Purpose of this survey:

We are developing the **EXPI-STOCK (Expiry Stock Inventory Management System)**, a web-based platform designed to help Wardah Baiduri Health & Beauty Shop manage product expiry dates more efficiently through automated tracking, timely alert notifications, and FIFO (First-In, First-Out) priority displays. Your feedback is crucial to help us create a system that truly meets your needs and operational requirements as a valued staff member or stakeholder of Wardah Baiduri.

What We Need From You:

- Share your current experiences with inventory management and product expiry tracking.
- Identify challenges you face when monitoring expiry dates and managing stock rotation.
- Help us prioritize the most valuable features for the new inventory management system.

Confidentiality

- Your responses are **anonymous** and will be used solely for improving our system development.
- Participation is **voluntary**, and you may skip any questions you prefer not to answer.

Time to Complete

- Estimated **3–5 minutes**

Thank you for contributing your insights and helping us build a better inventory management solution for Wardah Baiduri!

efahuda03@gmail.com Switch account

Not shared

1. What is your age range? *
- 18 -25 years
- 26 - 35 years
- 36 - 45 years
- 46+ years

2. What is your current position at Wardah Baiduri? *
- Assistant Manager
- Supervisor
- Senior Sales Assistant
- Sales Assistant
- Part-Time Staff
- Cashier

3. How many years of experience do you have working in retail/inventory management? *
- Less than 1 year
- 1 - 3 years
- 4 - 7 years
- More than 7 years

4. How is inventory management currently done for all products at Wardah Baiduri? *
- Manual tracking using pen and paper/logbook
- Basic spreadsheet like Excel or Google Sheets
- Simple point-of-sale system without expiry tracking
- No systematic tracking method

5. How often do you manually check product expiry dates? *
- Daily
- Weekly
- Monthly
- Only when customers ask or during stock taking
- Rarely or never

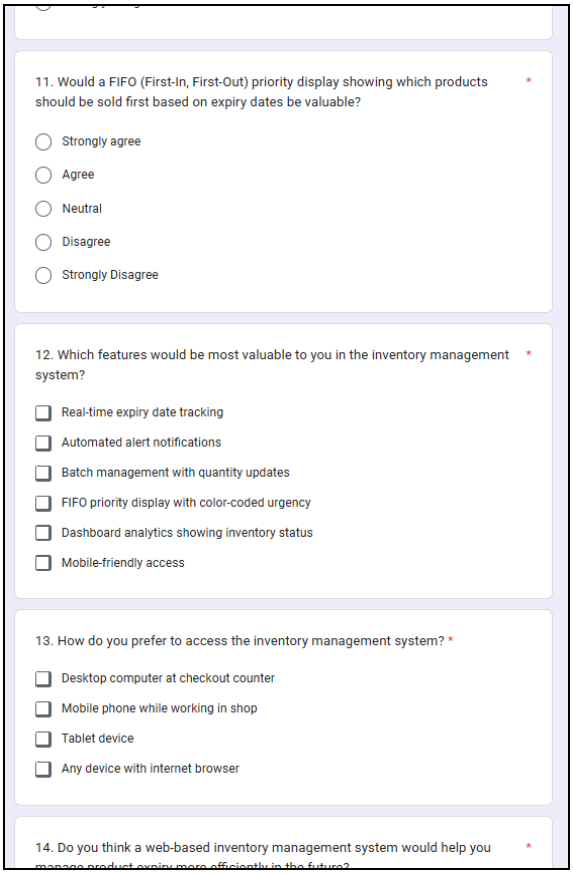
6. Have you ever experienced financial losses due to expired products at your workplace? *
- Yes, frequently (multiple times per month)
- Yes, occasionally (few times per year)
- Rarely
- No, never
- Not sure/Don't know

7. What challenges do you currently face in managing product expiry dates? *
- Difficulty remembering which products expire soon
- Too many products to track manually
- No reminder system for approaching expiry dates
- Hard to identify which products to sell first (FIFO)
- Time-consuming manual checking process

8. Would a web-based system that tracks all product expiry dates systematically in real-time be helpful for your work? *
- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

9. Would automated notifications like 30, 7, 3, and 1 days before expiry help you take action earlier to prevent financial losses? *
- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

10. Would the ability to create and track product batches with different expiry dates improve your inventory management? *
- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree



11. Would a FIFO (First-In, First-Out) priority display showing which products should be sold first based on expiry dates be valuable? *

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

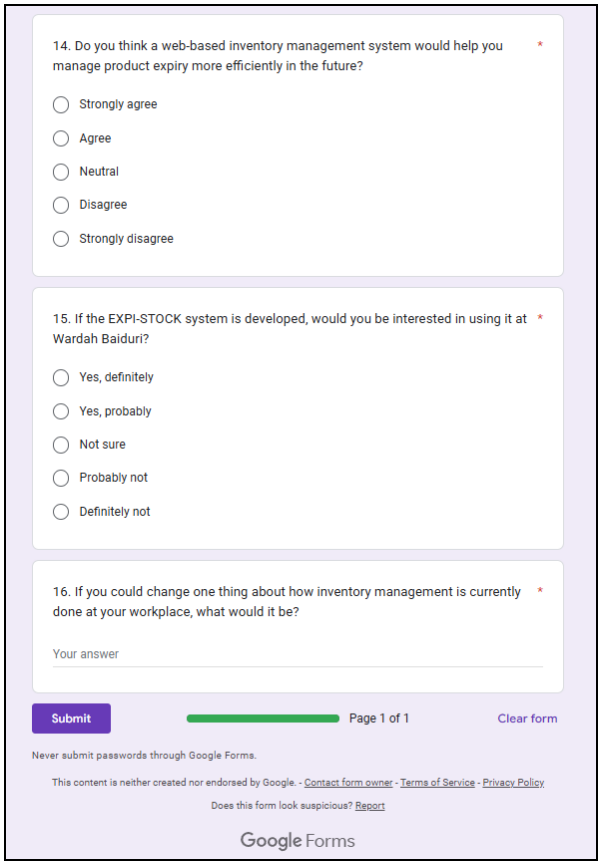
12. Which features would be most valuable to you in the inventory management system? *

- Real-time expiry date tracking
- Automated alert notifications
- Batch management with quantity updates
- FIFO priority display with color-coded urgency
- Dashboard analytics showing inventory status
- Mobile-friendly access

13. How do you prefer to access the inventory management system? *

- Desktop computer at checkout counter
- Mobile phone while working in shop
- Tablet device
- Any device with internet browser

14. Do you think a web-based inventory management system would help you manage product expiry more efficiently in the future? *



14. Do you think a web-based inventory management system would help you manage product expiry more efficiently in the future? *

- Strongly agree
- Agree
- Neutral
- Disagree
- Strongly disagree

15. If the EXPI-STOCK system is developed, would you be interested in using it at Wardah Baiduri? *

- Yes, definitely
- Yes, probably
- Not sure
- Probably not
- Definitely not

16. If you could change one thing about how inventory management is currently done at your workplace, what would it be? *

Your answer

Submit Page 1 of 1 Clear form

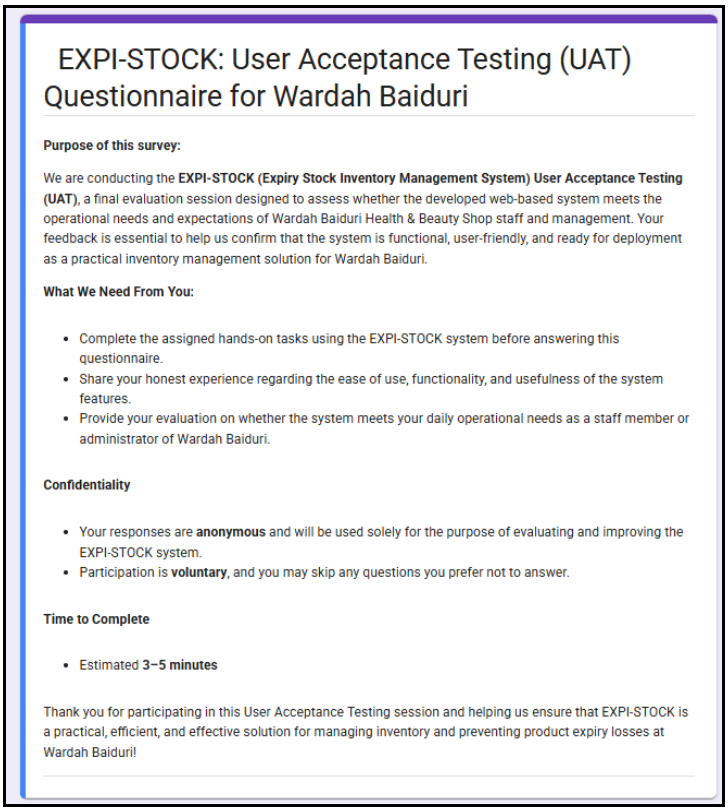
Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. - Contact form owner - Terms of Service - Privacy Policy

Does this form look suspicious? Report

Google Forms

Questionnaire (Post-Development)



EXPI-STOCK: User Acceptance Testing (UAT) Questionnaire for Wardah Baiduri

Purpose of this survey:

We are conducting the **EXPI-STOCK (Expiry Stock Inventory Management System) User Acceptance Testing (UAT)**, a final evaluation session designed to assess whether the developed web-based system meets the operational needs and expectations of Wardah Baiduri Health & Beauty Shop staff and management. Your feedback is essential to help us confirm that the system is functional, user-friendly, and ready for deployment as a practical inventory management solution for Wardah Baiduri.

What We Need From You:

- Complete the assigned hands-on tasks using the EXPI-STOCK system before answering this questionnaire.
- Share your honest experience regarding the ease of use, functionality, and usefulness of the system features.
- Provide your evaluation on whether the system meets your daily operational needs as a staff member or administrator of Wardah Baiduri.

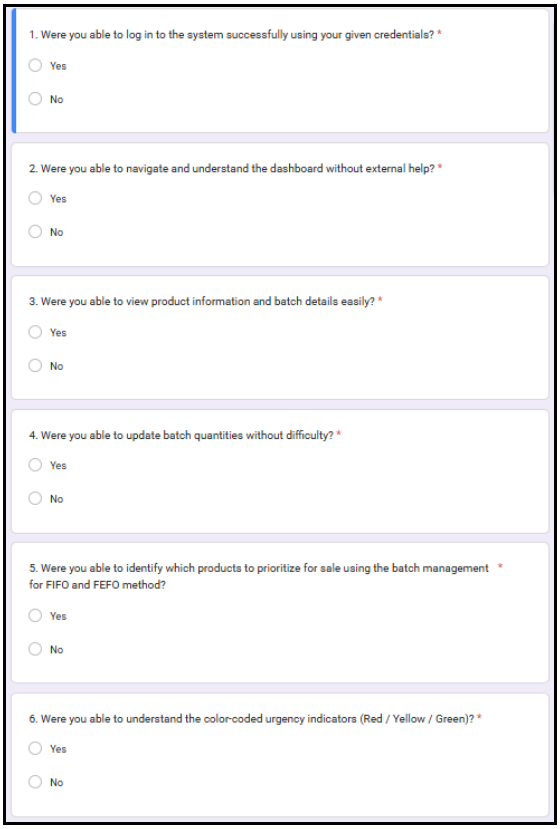
Confidentiality

- Your responses are **anonymous** and will be used solely for the purpose of evaluating and improving the EXPI-STOCK system.
- Participation is **voluntary**, and you may skip any questions you prefer not to answer.

Time to Complete

- Estimated **3–5 minutes**

Thank you for participating in this User Acceptance Testing session and helping us ensure that EXPI-STOCK is a practical, efficient, and effective solution for managing inventory and preventing product expiry losses at Wardah Baiduri!



1. Were you able to log in to the system successfully using your given credentials? *
- Yes
- No

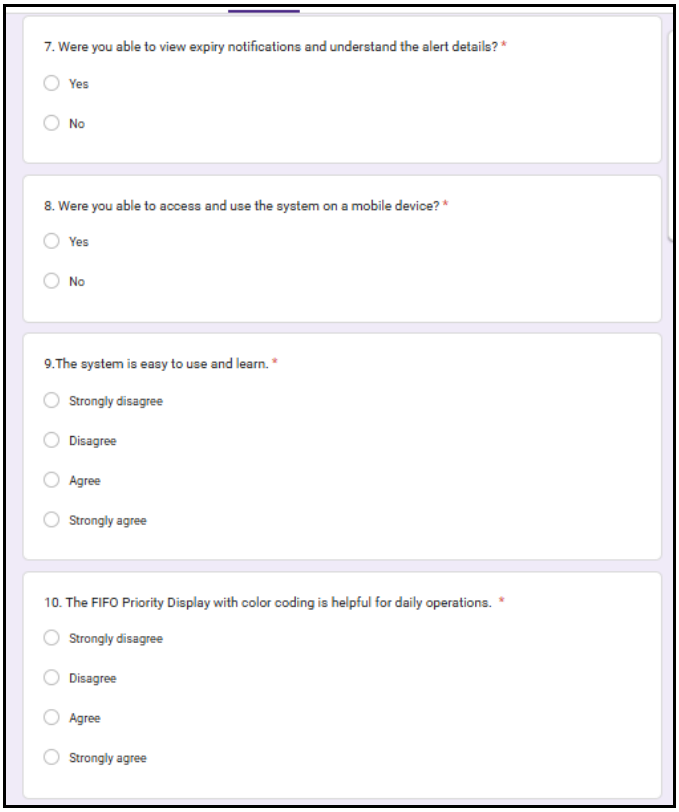
2. Were you able to navigate and understand the dashboard without external help? *
- Yes
- No

3. Were you able to view product information and batch details easily? *
- Yes
- No

4. Were you able to update batch quantities without difficulty? *
- Yes
- No

5. Were you able to identify which products to prioritize for sale using the batch management for FIFO and FEFO method? *
- Yes
- No

6. Were you able to understand the color-coded urgency indicators (Red / Yellow / Green)? *
- Yes
- No

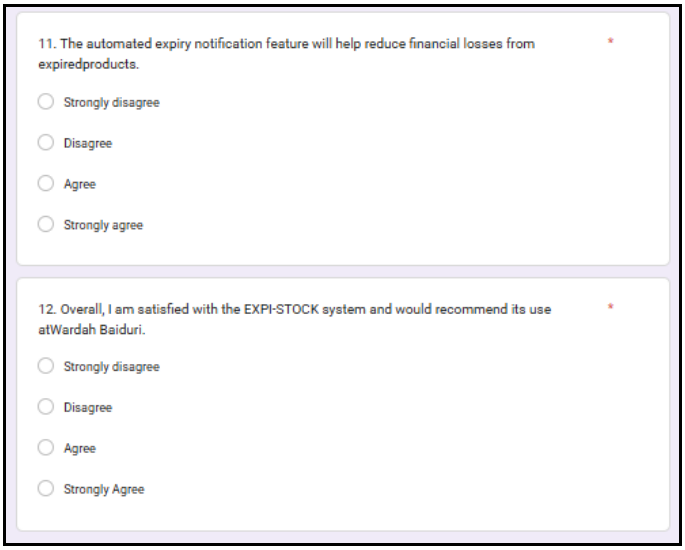


7. Were you able to view expiry notifications and understand the alert details? *
- Yes
- No

8. Were you able to access and use the system on a mobile device? *
- Yes
- No

9.The system is easy to use and learn. *
- Strongly disagree
- Disagree
- Agree
- Strongly agree

10. The FIFO Priority Display with color coding is helpful for daily operations. *
- Strongly disagree
- Disagree
- Agree
- Strongly agree



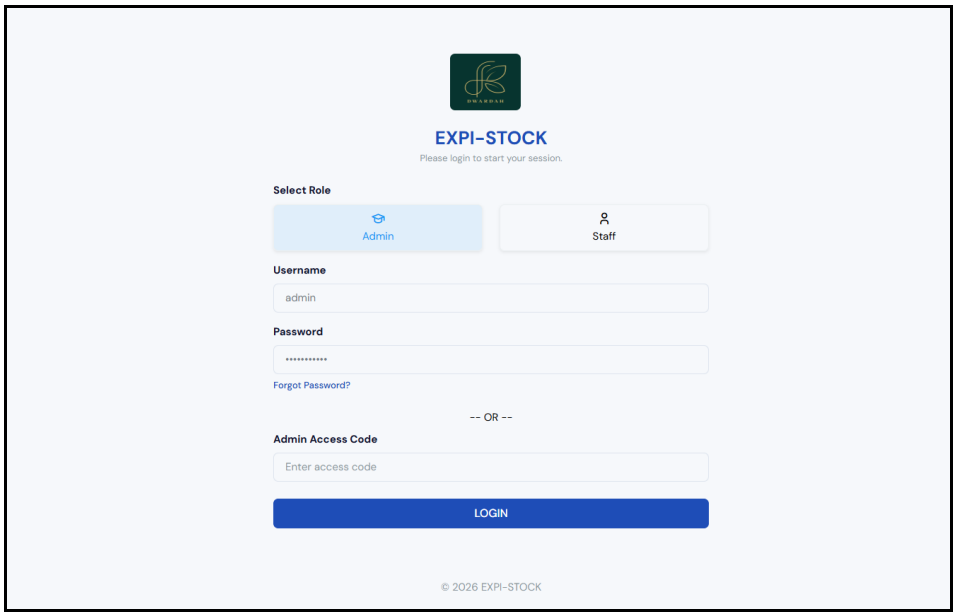
11. The automated expiry notification feature will help reduce financial losses from expiredproducts. *
- Strongly disagree
- Disagree
- Agree
- Strongly agree

12. Overall, I am satisfied with the EXPI-STOCK system and would recommend its use atWardah Baiduri. *
- Strongly disagree
- Disagree
- Agree
- Strongly Agree

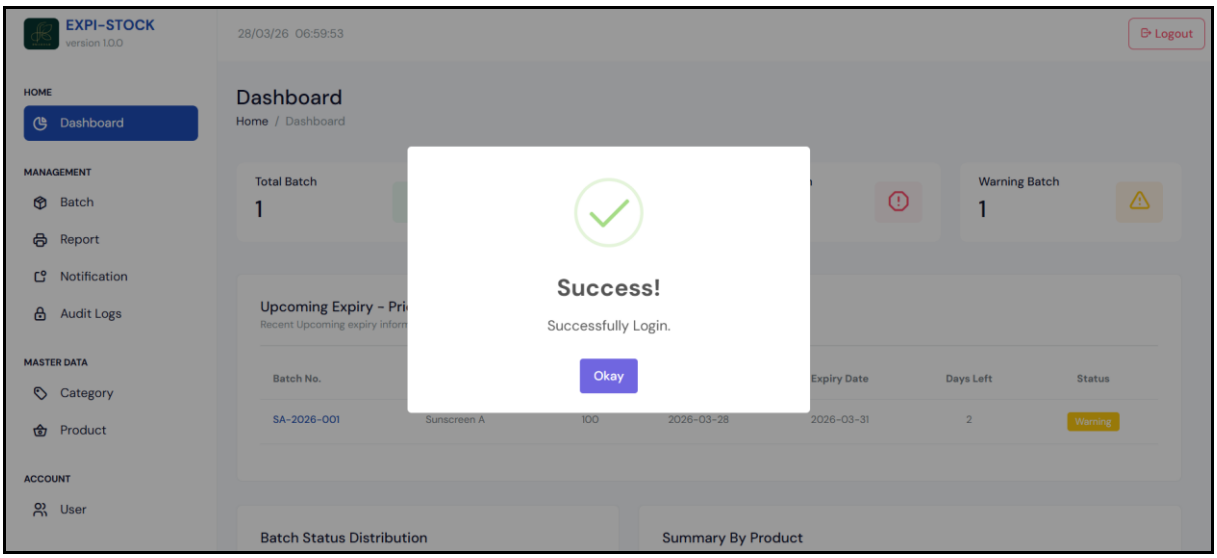
Appendix B – User Manual

Part A: Admin Module

Admin Login

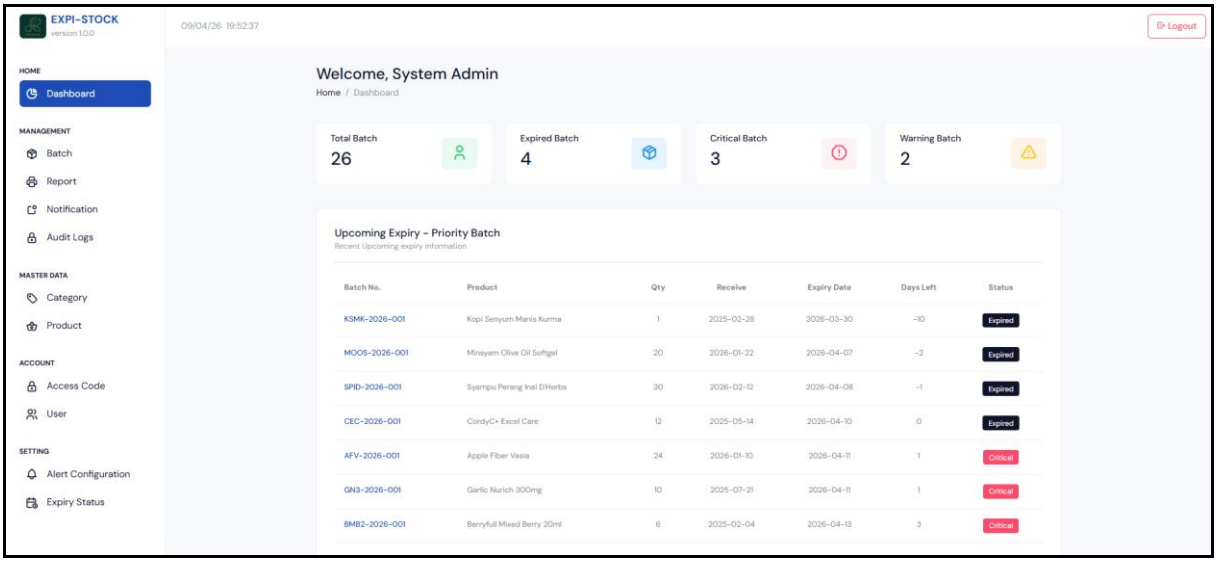


- Admin needs to insert username and password to login the system or login use access code.

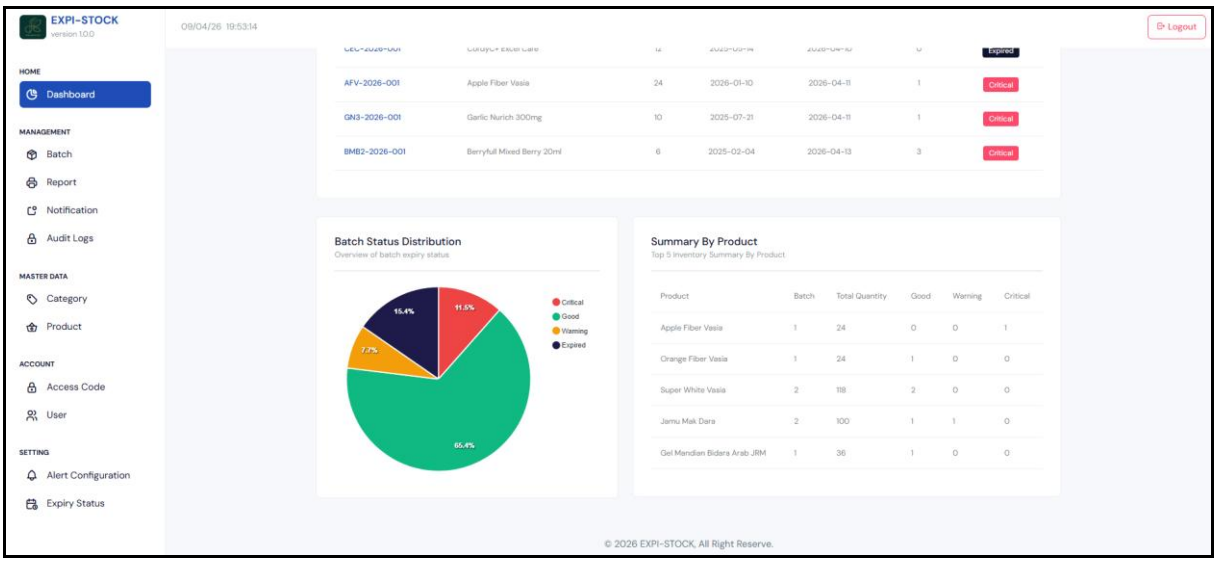


- A success popup will appear after successful login.

Admin Dashboard

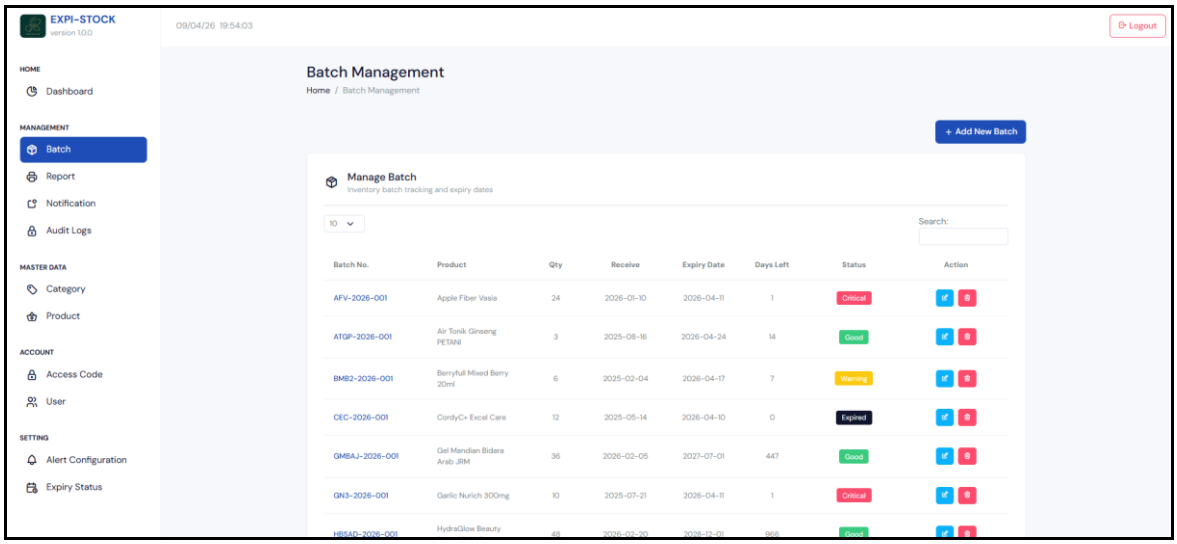


- Dashboard page part 1 - overview of system statistics.

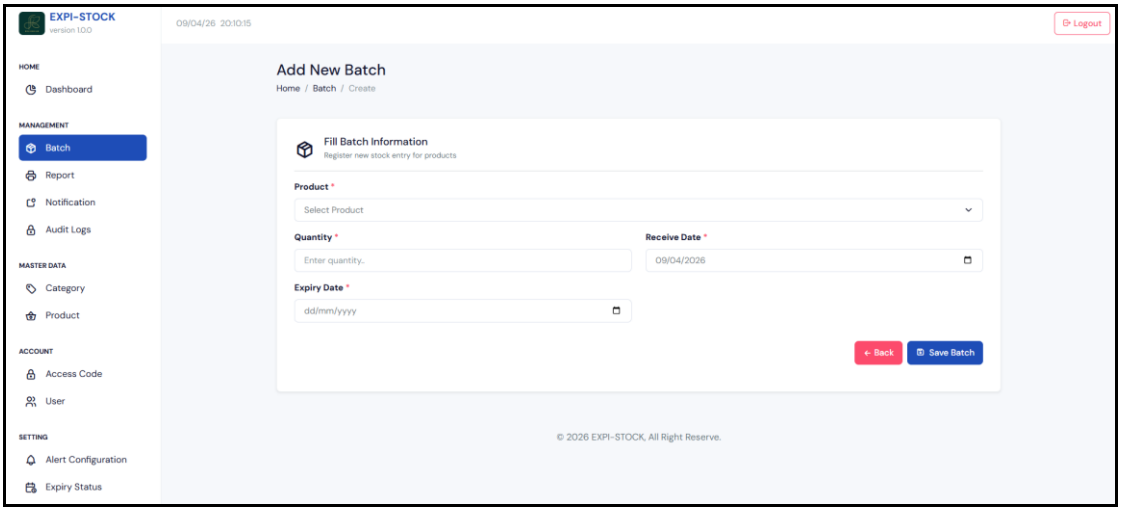


- Dashboard page part 2 - visible after scrolling down.

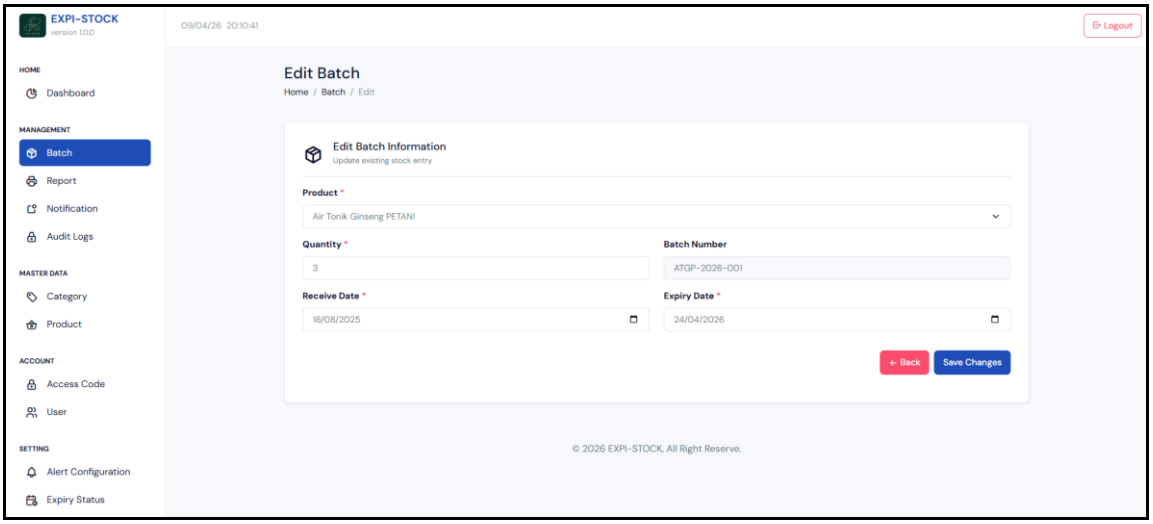
Batch Management



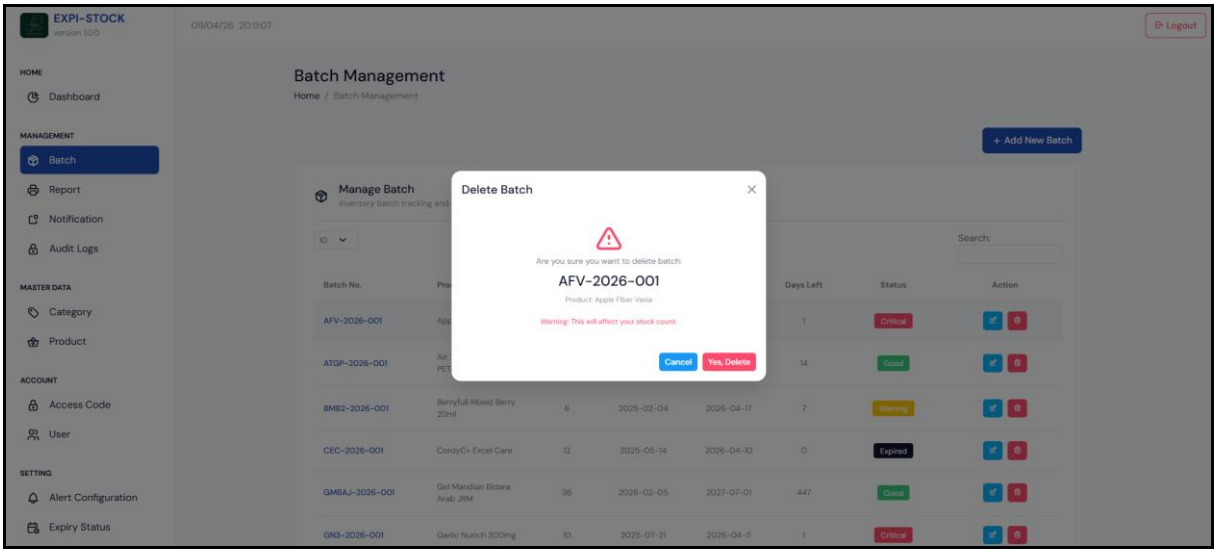
- Batch page - displays expiry status for each batch.



- Add New Batch page.

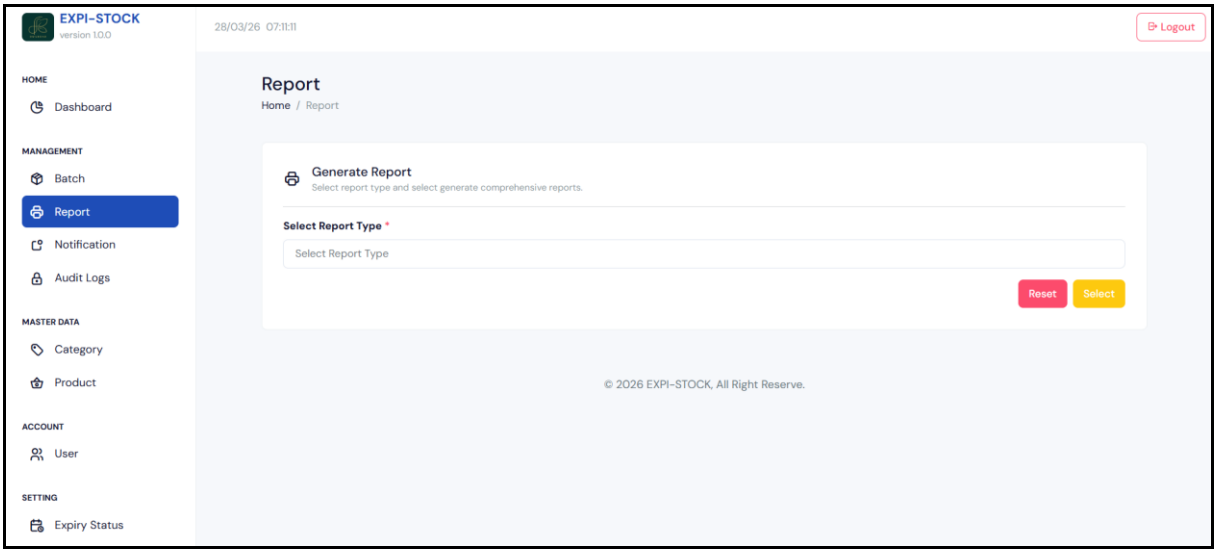


- Edit Batch page.

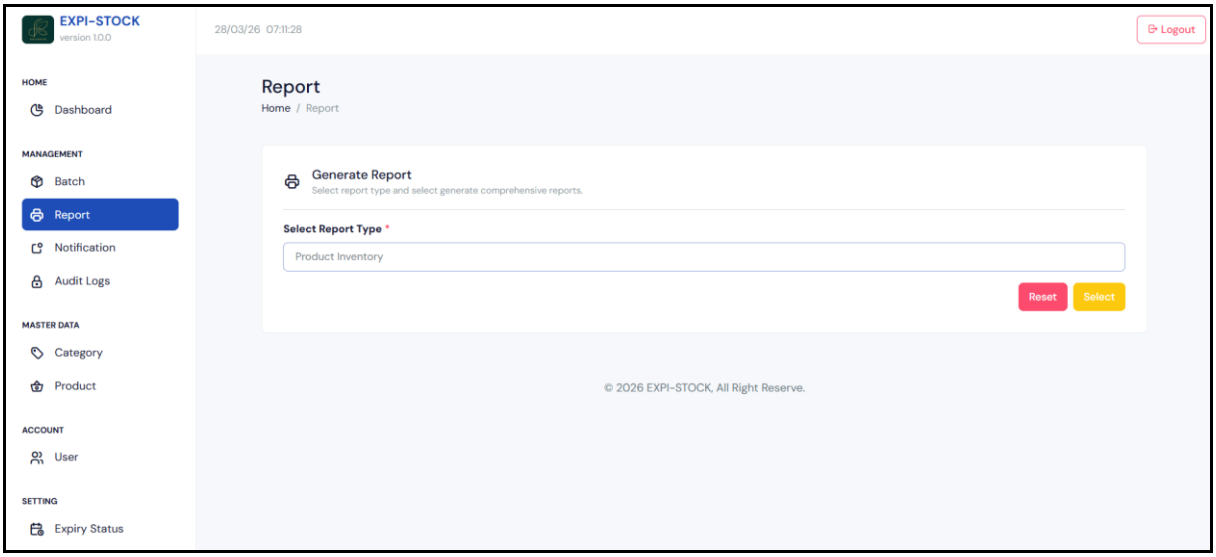


- Delete Batch confirmation popup.

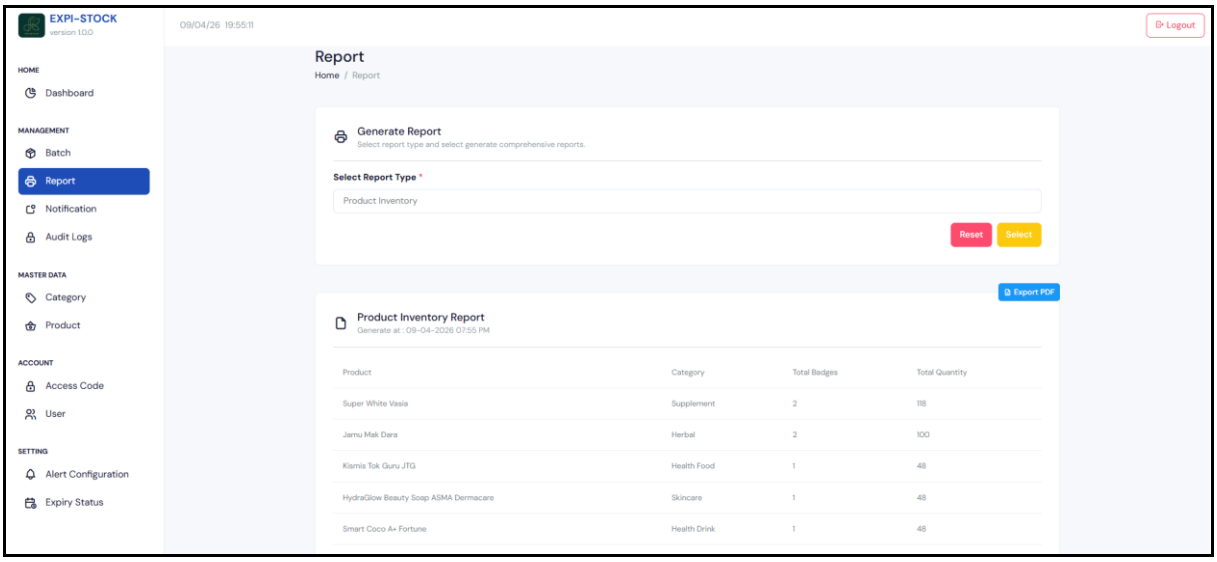
Report



- Report page overview.



◦ User selects Product Inventory report type. There are 2 report types: Product Inventory and Expiry Status - helping staff/admin identify which products need to be sold first or restocked urgently.



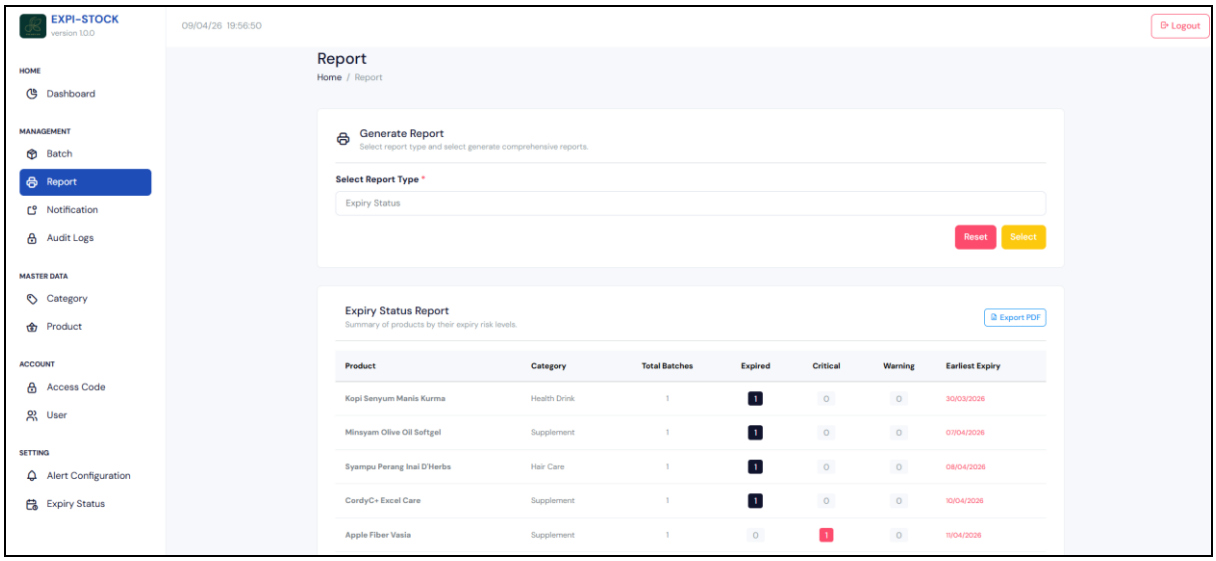
◦ Report page after user clicks Select.

Product Inventory

Generated at: 09/04/2026 07:55 PM

No	Product	Category	Total Batches	Total Quantity
1	Apple Fiber Vasia	Supplement	1	24
2	Orange Fiber Vasia	Supplement	1	24
3	Super White Vasia	Supplement	2	118
4	Jamu Mak Dara	Herbal	2	100
5	Gel Mandian Bidara Arab JRM	Self-Care	1	36
6	Lotion Mustajab Extreme Hot Dunia Herbs	Self-Care	1	24
7	Sacha Inchi Mandian Shifa Herb	Self-Care	1	30
8	Kismis Tok Guru JTG	Health Food	1	48
9	Serbuk Halia Madu AMRAN	Health Drink	1	36
10	Minsyam Olive Oil Softgel	Supplement	1	20
11	Minsyam Black Seed Oil Capsule	Supplement	1	20
12	Garlic Nurich 300mg	Supplement	1	10
13	CordyC+ Excel Care	Supplement	1	12
14	Berryfull Mixed Berry 20ml	Health Drink	1	6
15	Air Tonik Ginseng PETANI	Health Drink	1	3

- PDF report generated after clicking "Export PDF".



- Report page after user selects Expiry Status report type.

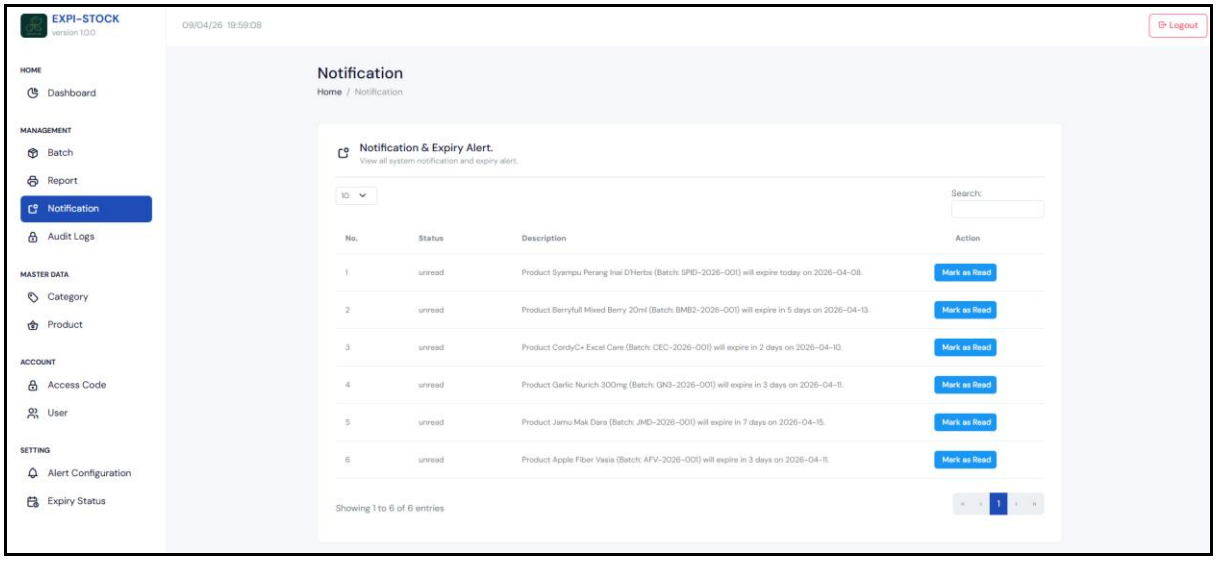
EXPIRY STATUS

Generated Date: 09/04/2026 07:57 PM

NO	PRODUCT NAME	CATEGORY	TOTAL BATCHES	EXPIRED	CRITICAL	WARNING	EARLIEST EXPIRY
1	Apple Fiber Vasia	Supplement	1	0	1	0	11/04/2026
2	Orange Fiber Vasia	Supplement	1	0	0	0	24/12/2027
3	Super White Vasia	Supplement	2	0	0	0	09/07/2027
4	Jamu Mak Dara	Herbal	2	0	0	1	15/04/2026
5	Gel Mandian Bidara Arab JRM	Self-Care	1	0	0	0	01/07/2027
6	Lotion Mustajab Extreme Hot Dunia Herbs	Self-Care	1	0	0	0	01/04/2028
7	Sacha Inchi Mandian Shifa Herb	Self-Care	1	0	0	0	01/01/2027
8	Kismis Tok Guru JTG	Health Food	1	0	0	0	30/12/2026
9	Serbuk Halia Madu AMRAN	Health Drink	1	0	0	0	30/05/2027
10	Minsyam Olive Oil Softgel	Supplement	1	1	0	0	07/04/2026
11	Minsyam Black Seed Oil Capsule	Supplement	1	0	0	0	01/09/2027
12	Garlic Nurich	Supplement	1	0	1	0	11/04/2026

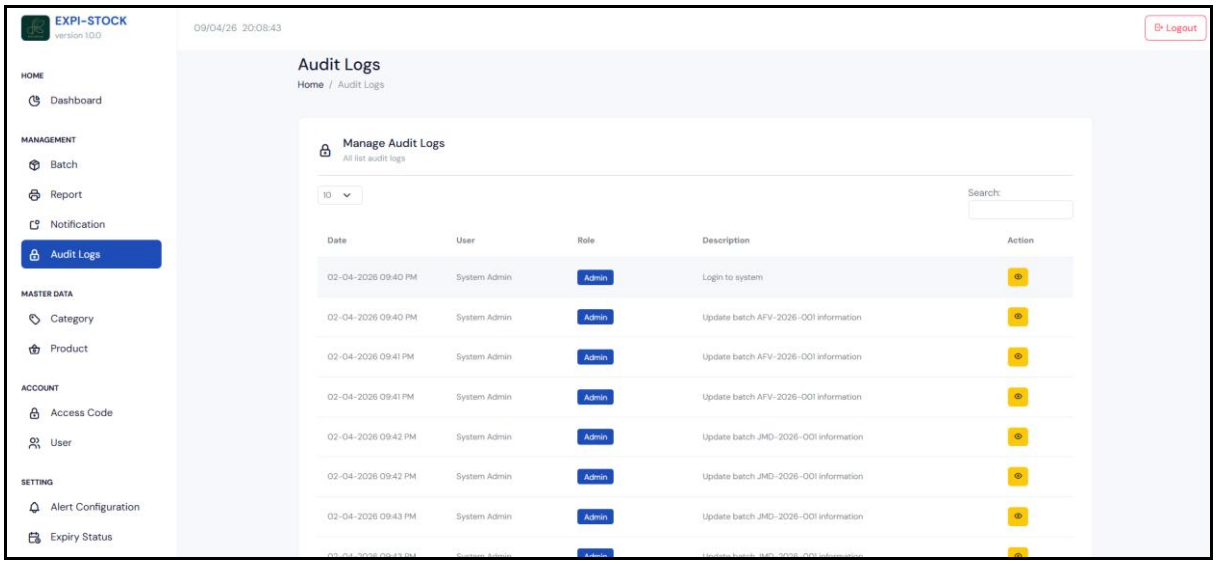
- Expiry Status PDF report after clicking Export PDF.

Notification

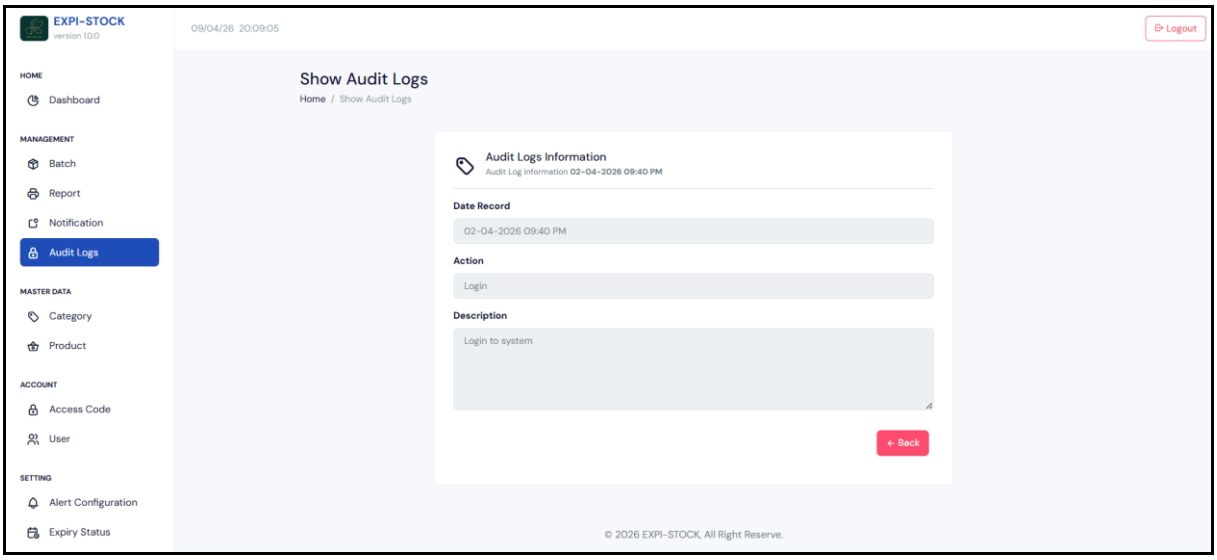


- Notification panel - alerts user to critical, warning, or expired products.
- Users also receive these alerts via their registered email.

Audit Logs

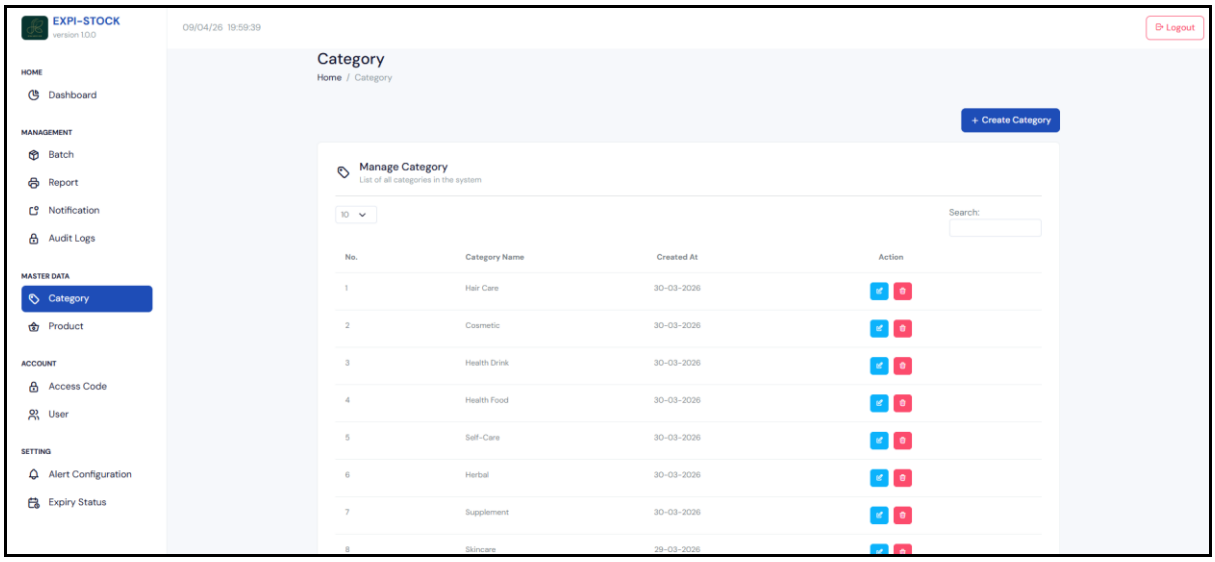


- Audit Logs page - accessible by Admin only. Used to monitor user activity and enhance system security.

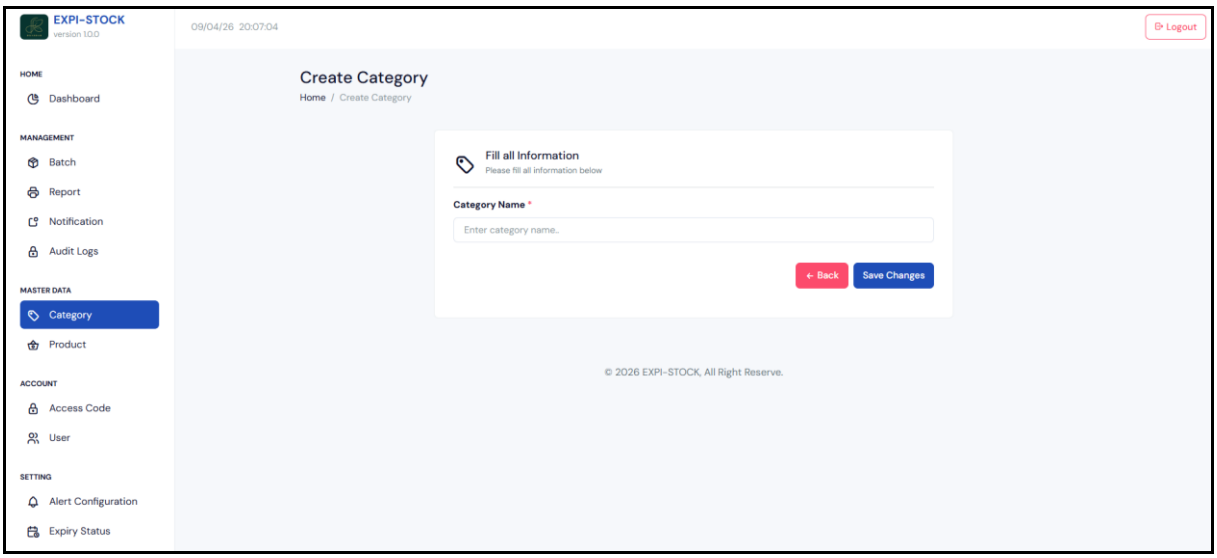


- Audit Log detail view after clicking the View button.

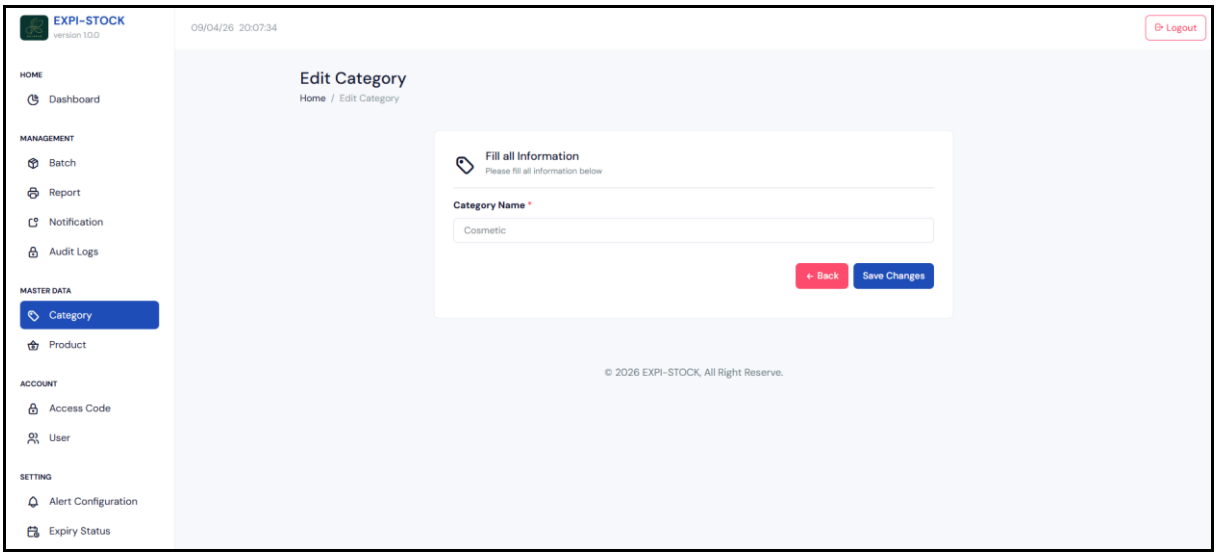
Category Management



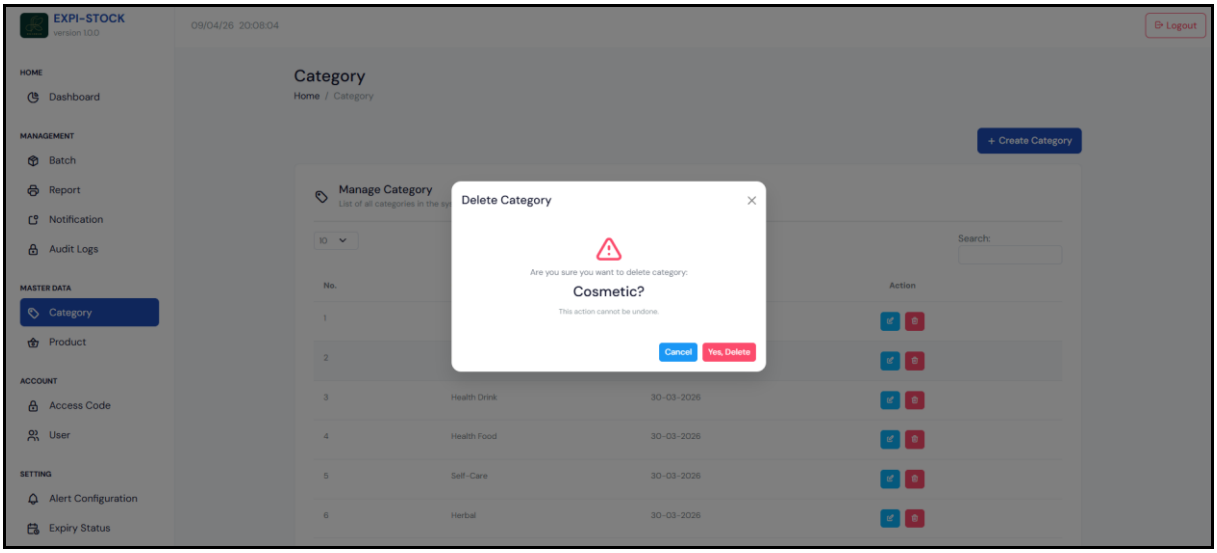
- Category page - Admin can add, edit, or delete categories (e.g. Cosmetic, Skincare).



- Add Category page.

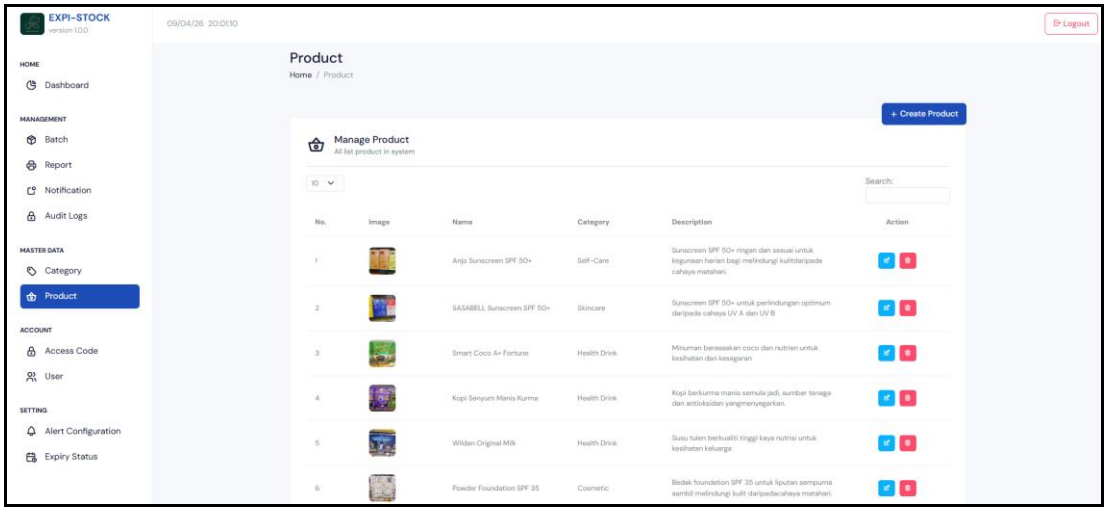


- Edit Category page.

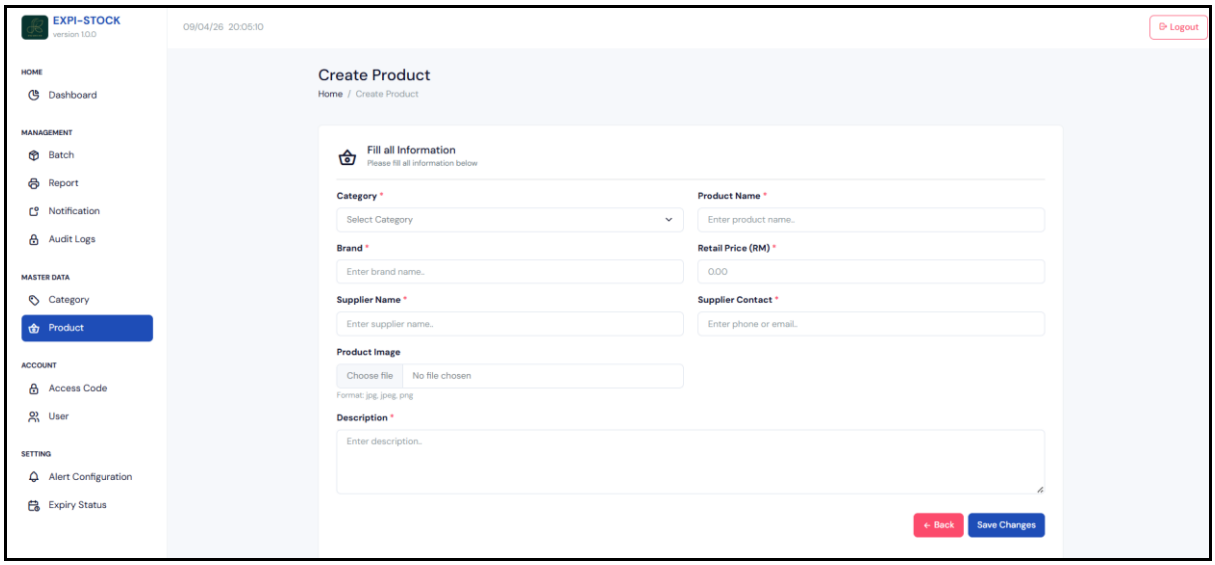


- Delete Category confirmation popup.

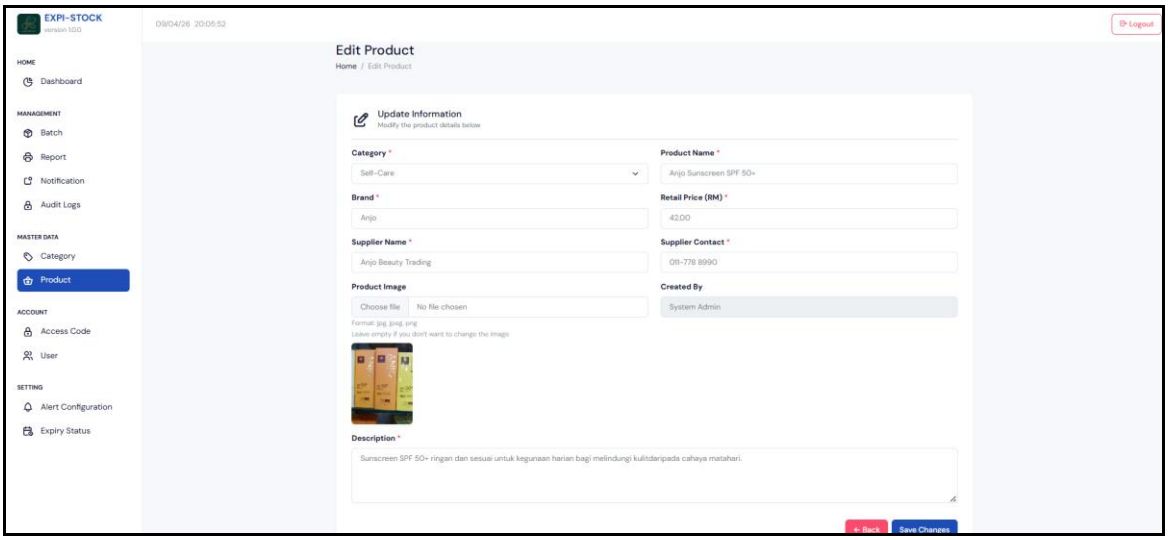
Product Management



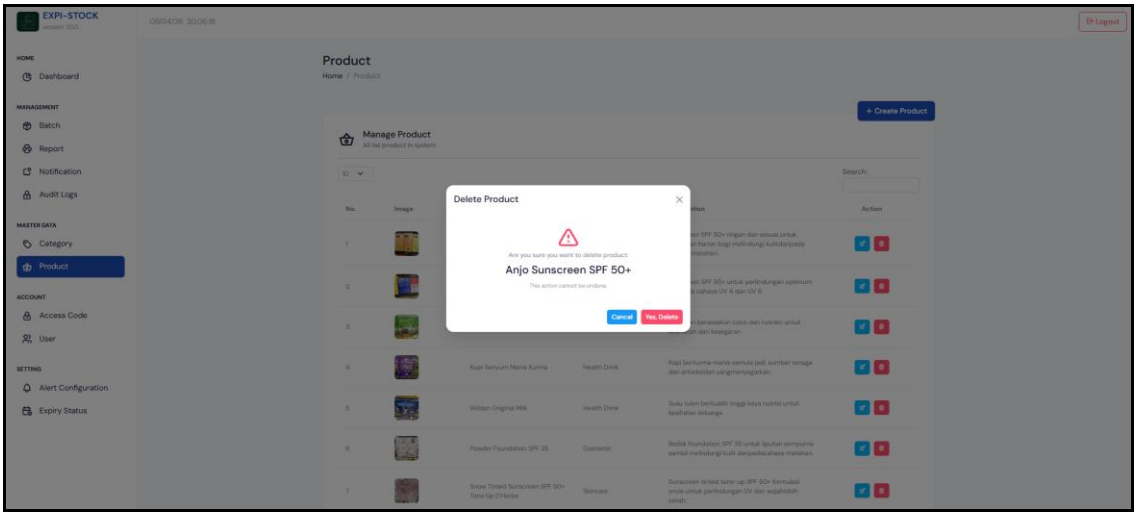
- Product page - Admin can add, edit, and delete products.



- Add Product page.

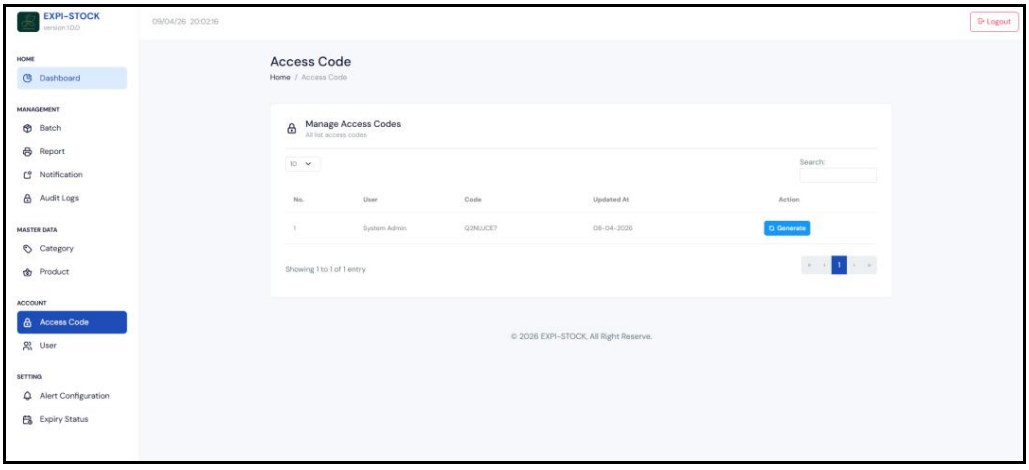


- Edit Product page.



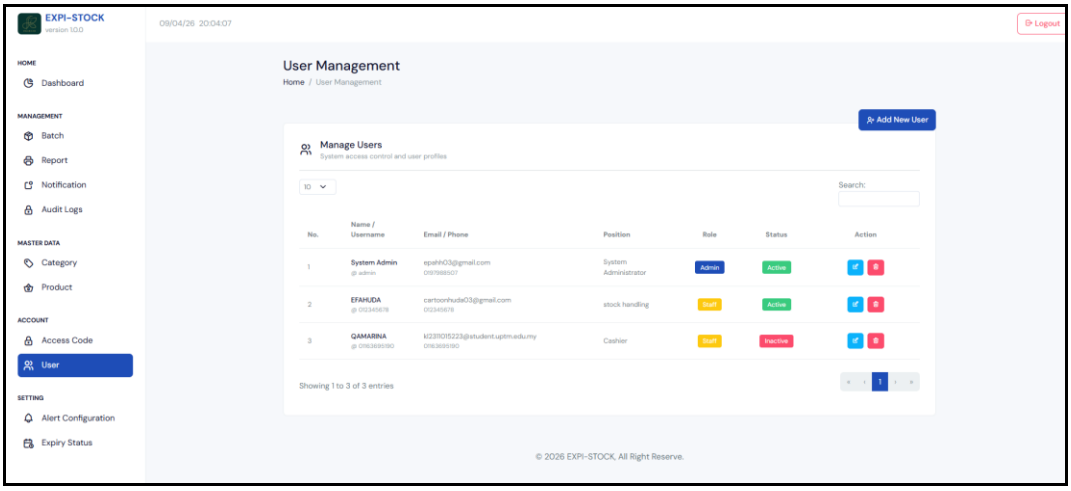
- Delete Product confirmation popup.

Access Code

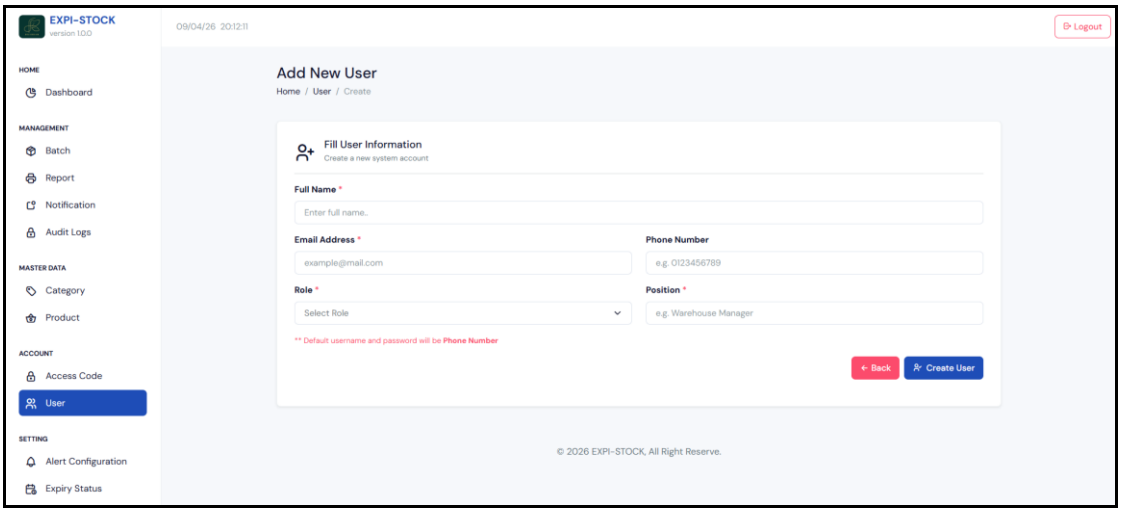


- Access Code page - Admin can generate a new access code for login.

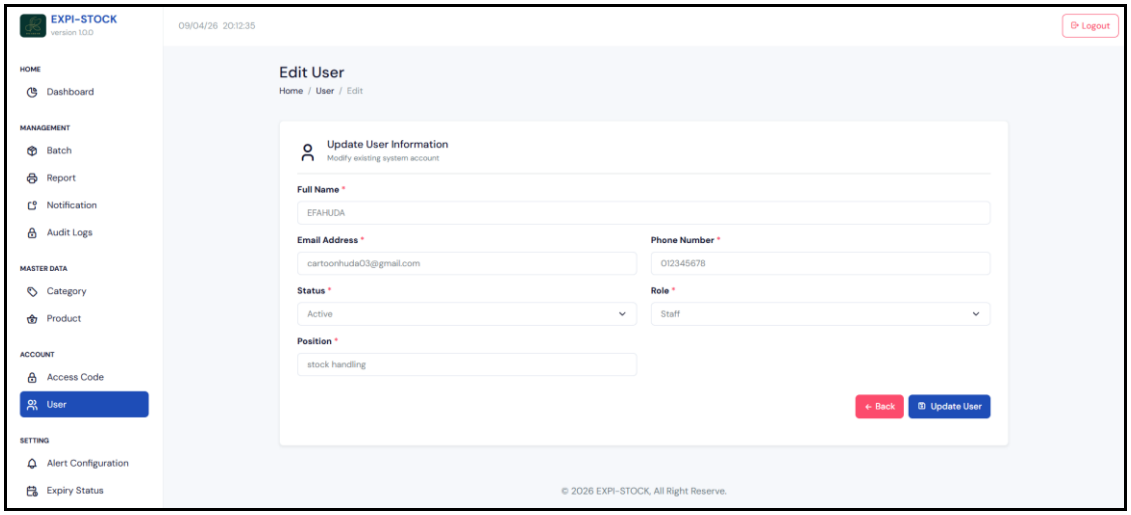
User Management



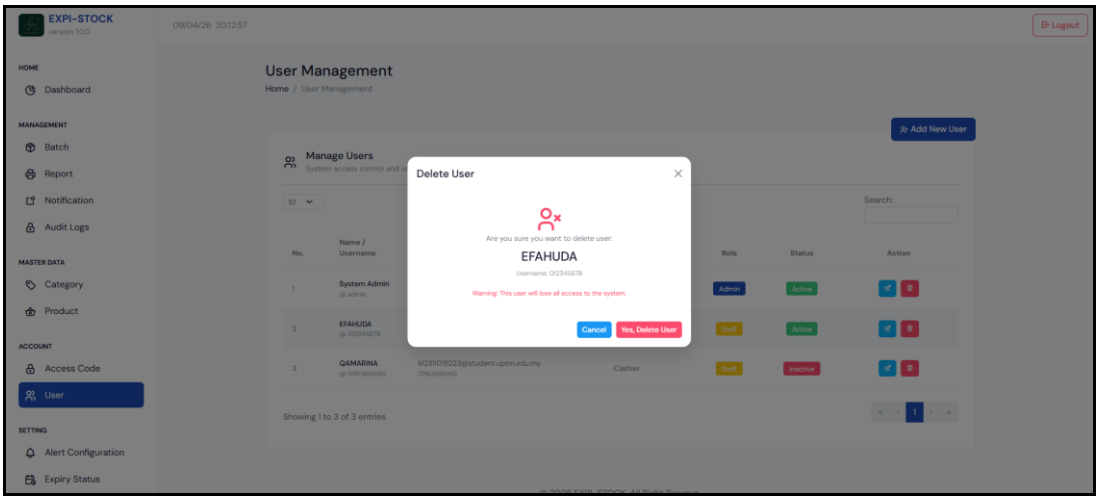
- User Management page - Admin-only. Admin can add staff accounts and new admin accounts.
-



- Add User page.

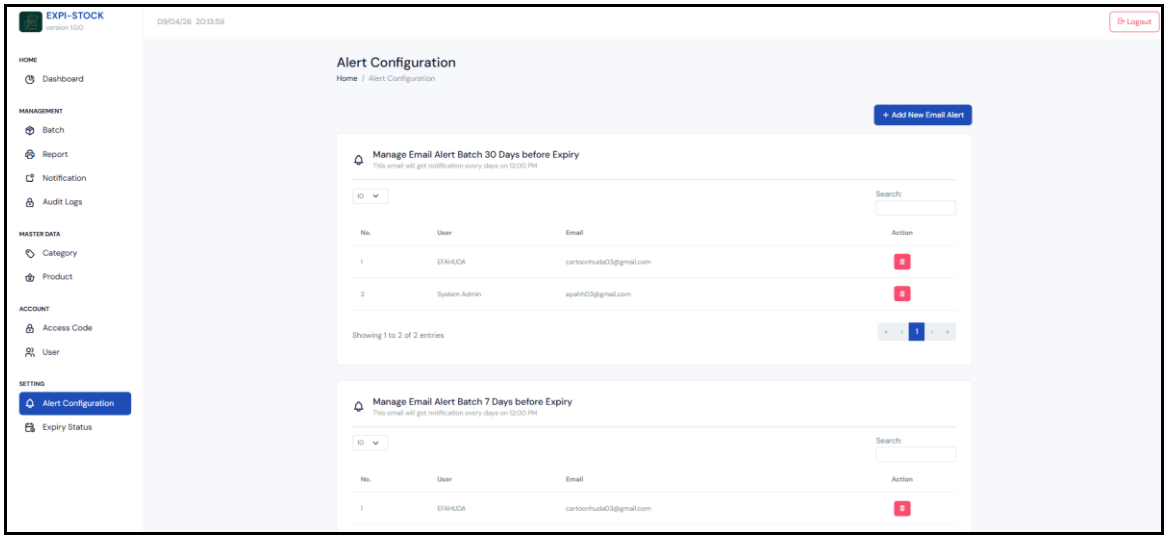


- Edit User page.

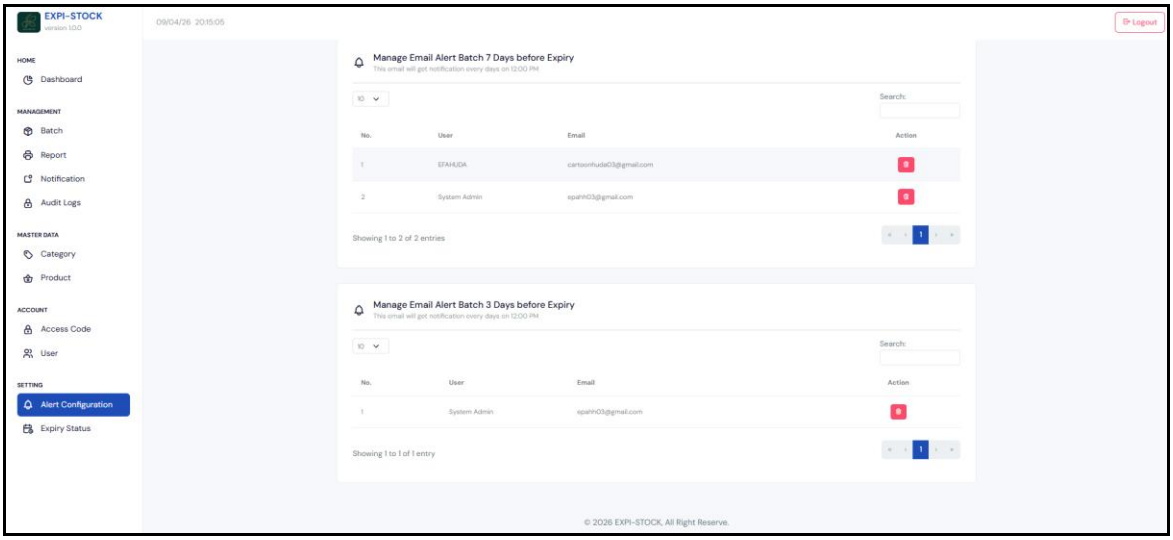


- Delete User confirmation popup.

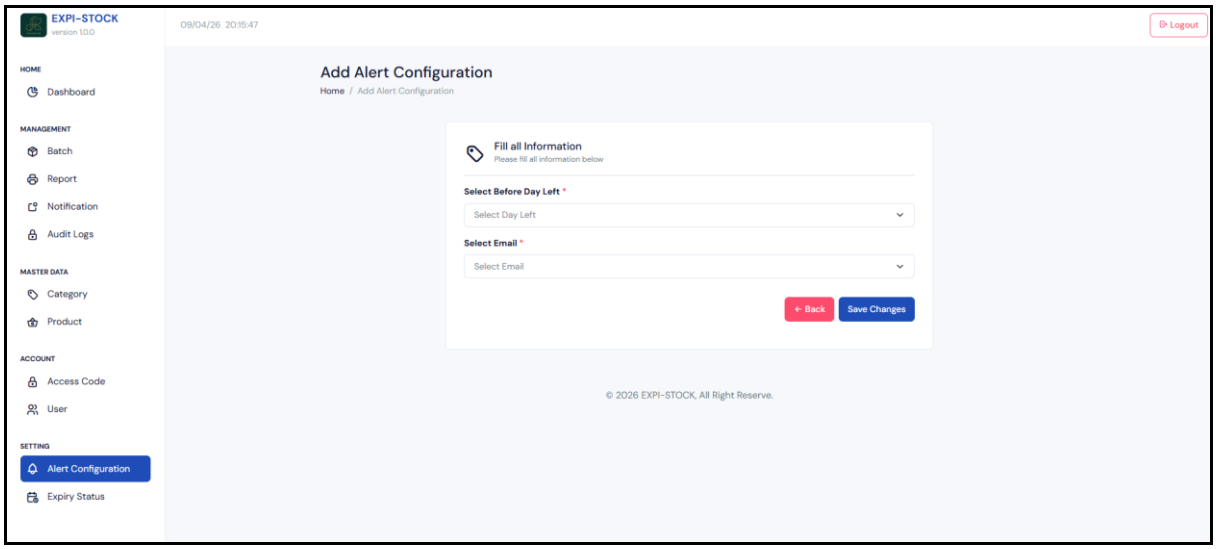
Alert Configuration



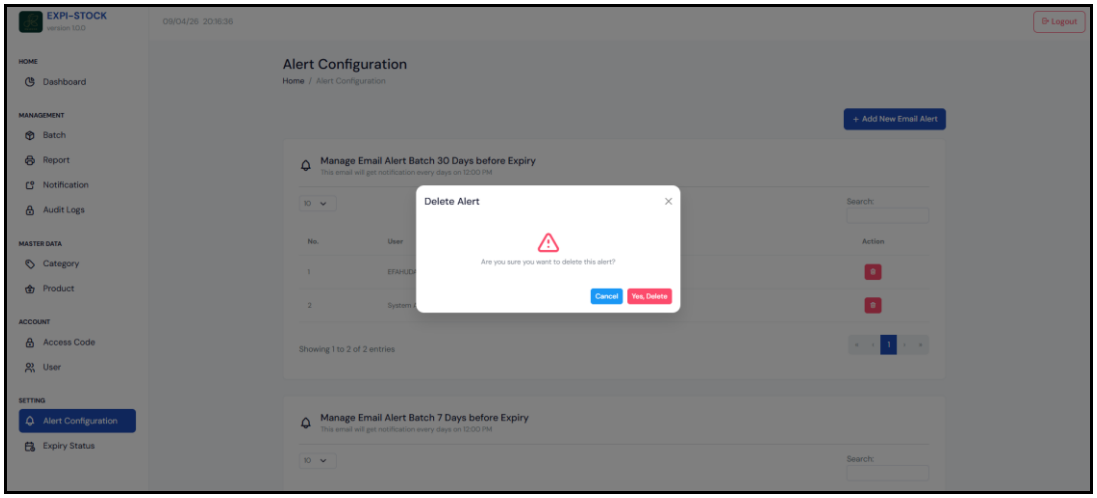
- Alert Configuration page part 1 - set which email addresses receive expiry alert notifications.



- Alert Configuration page part 2.

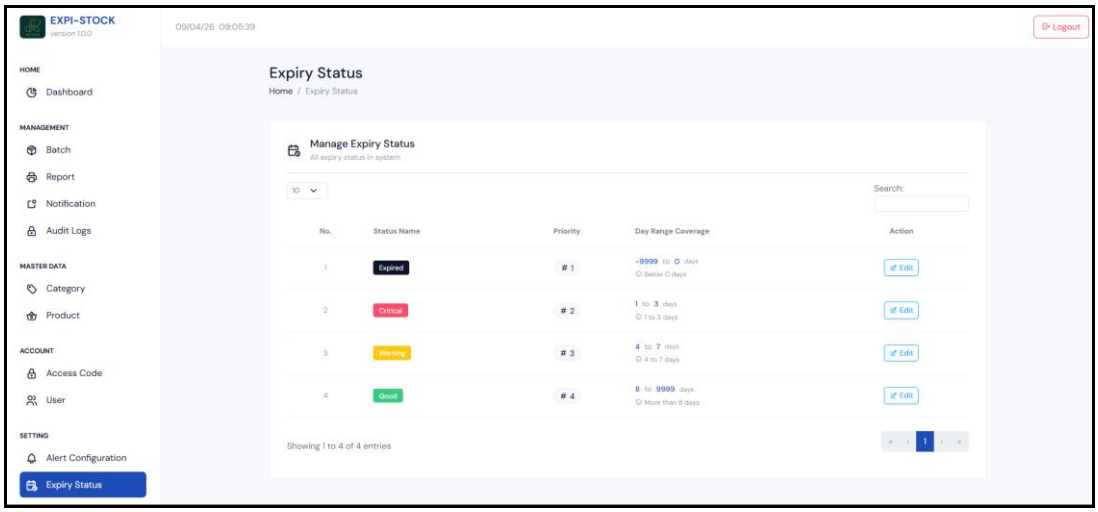


- Add New Email Alert page.

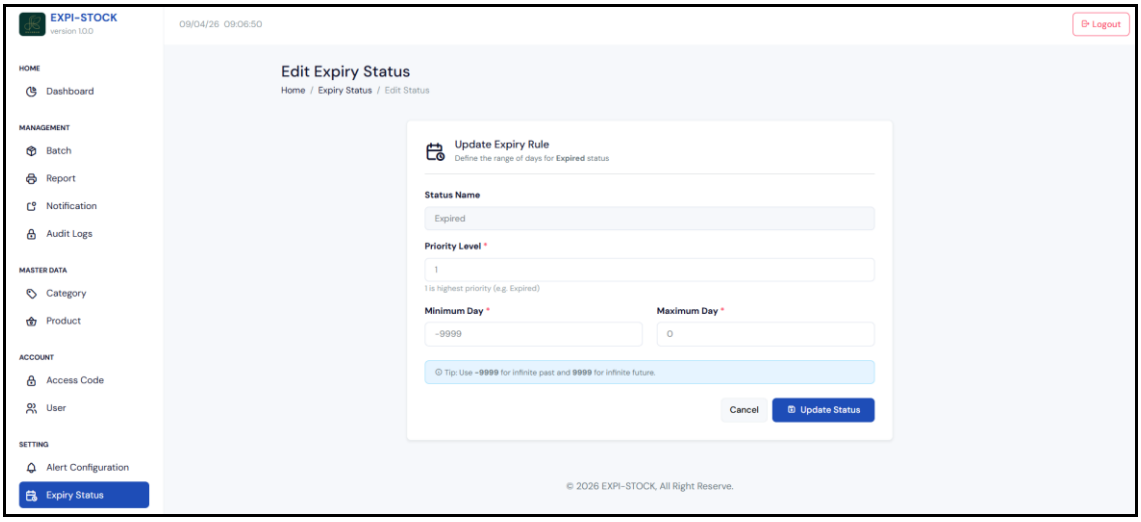


- Delete Email Alert confirmation popup.

Expiry Status Settings

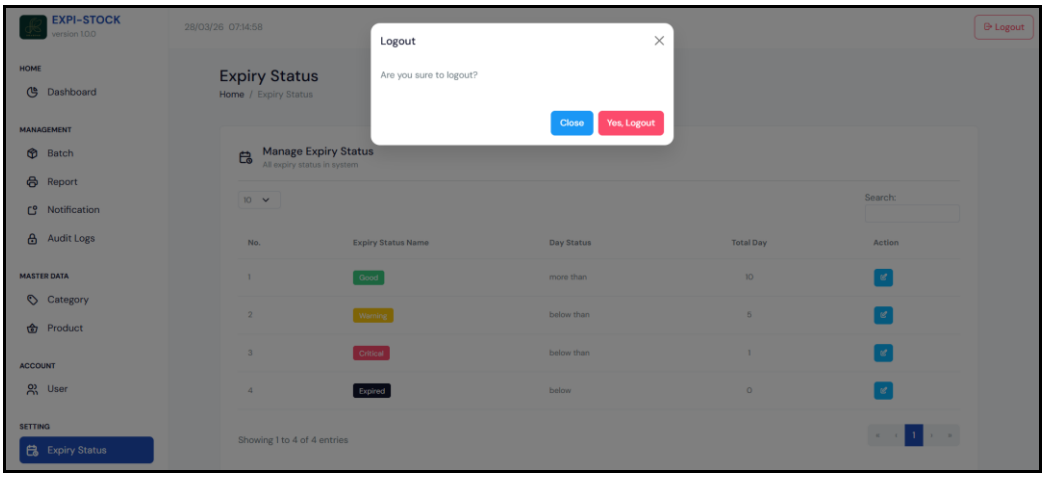


- Expiry Status page - Admin-only. Set the day-range thresholds for Good, Warning, Critical, or Expired product status.



- Edit Expiry Status page.

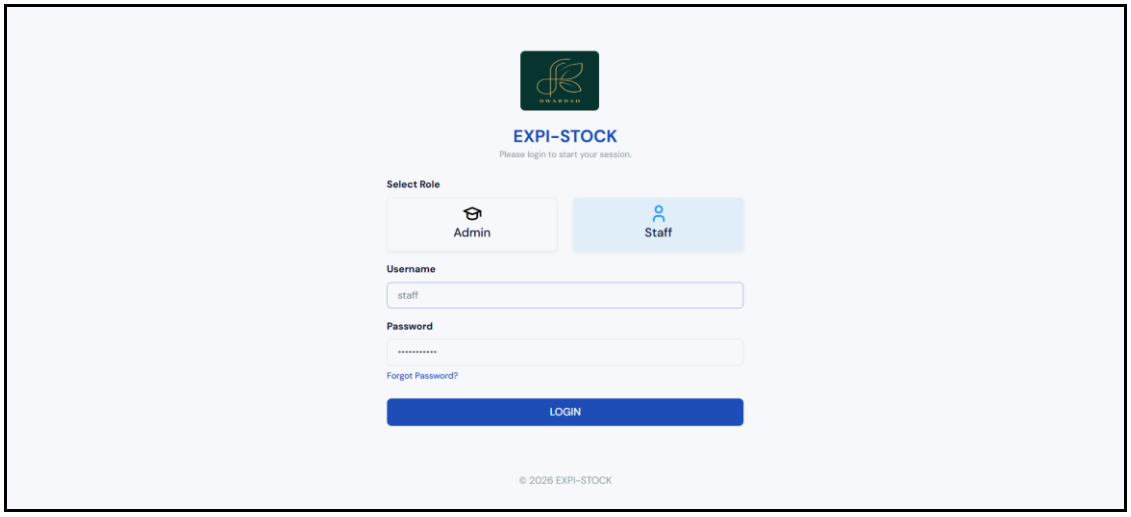
Logout



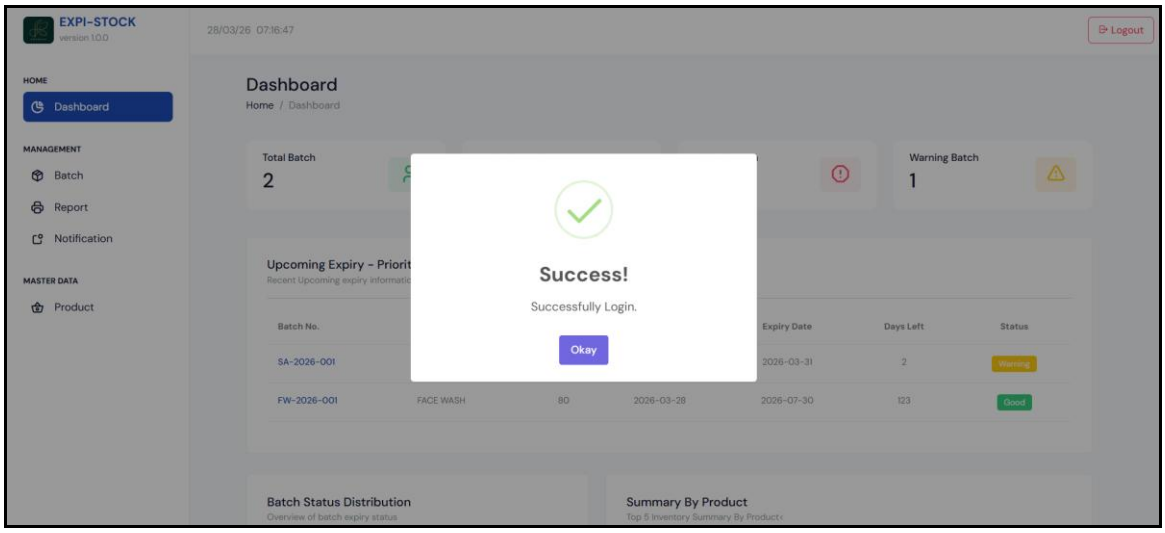
- Logout confirmation popup. After confirming, the login page is displayed.

Part B: Staff Module

Staff Login

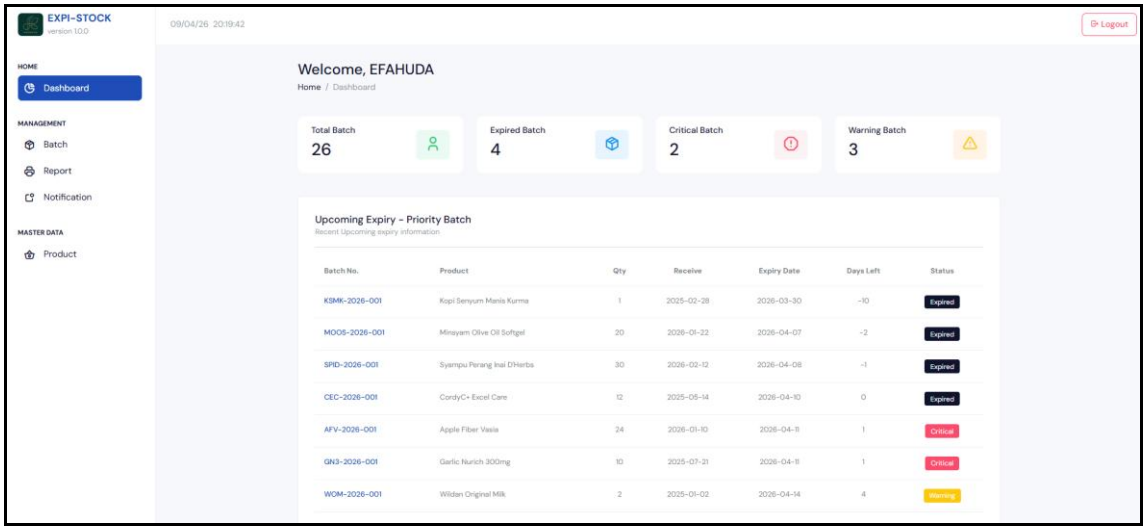


- Staff login page - no access code required.

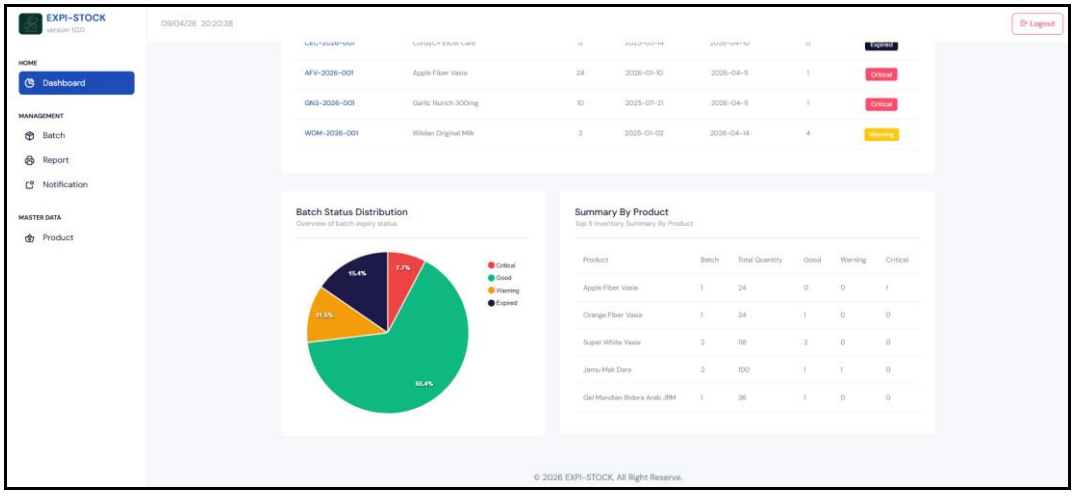


- Success popup displayed after successful login.

Staff Dashboard

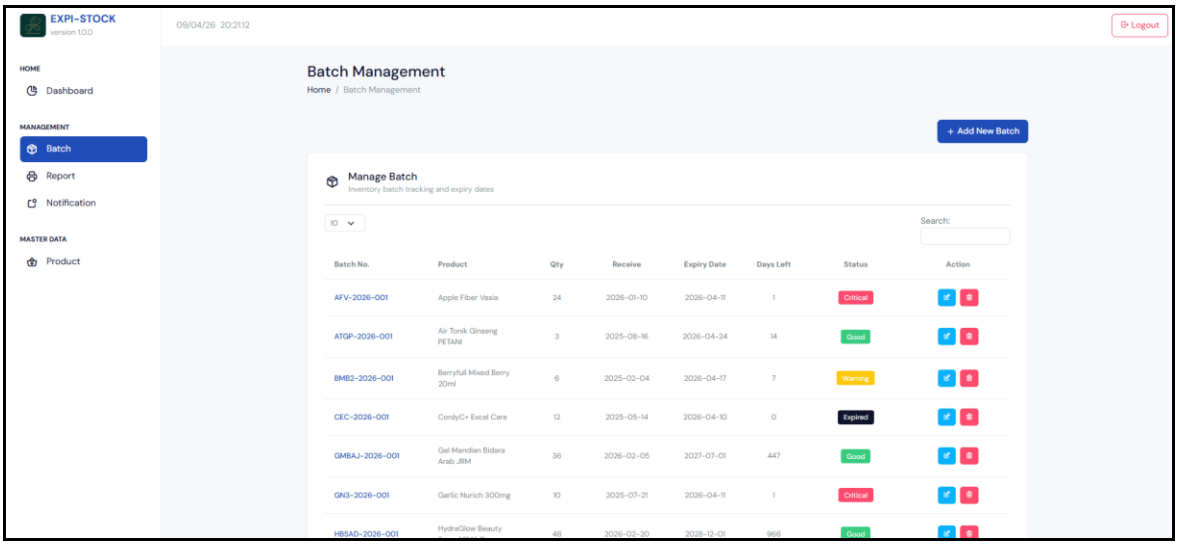


- Dashboard page part 1.

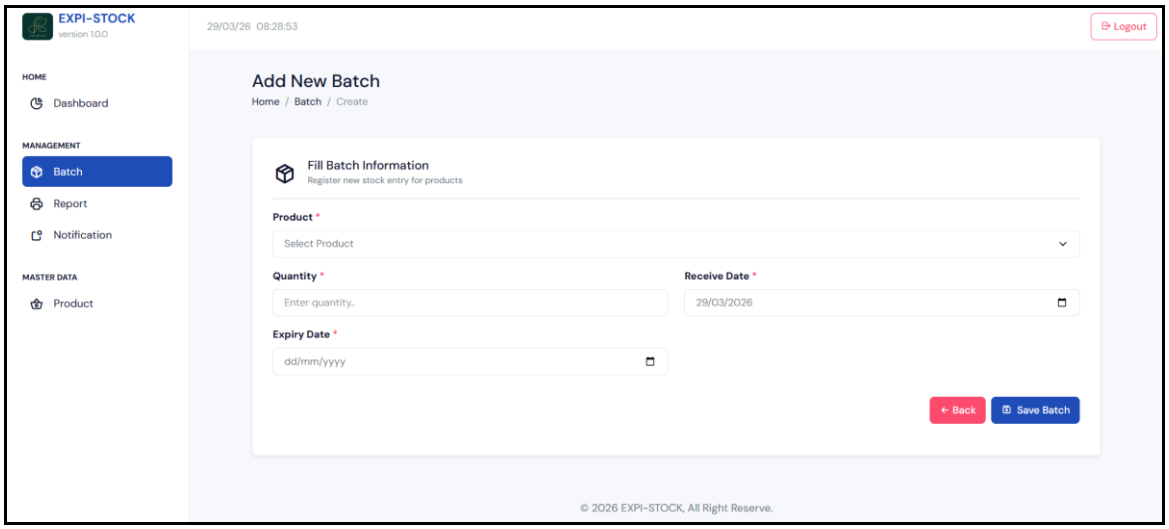


- Dashboard page part 2 - visible after scrolling down.

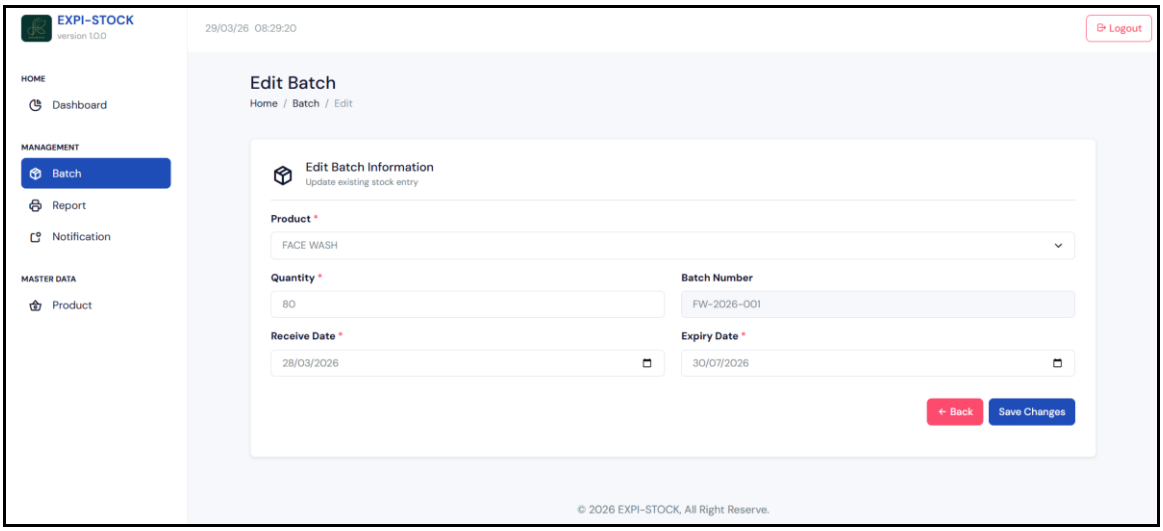
Batch Management



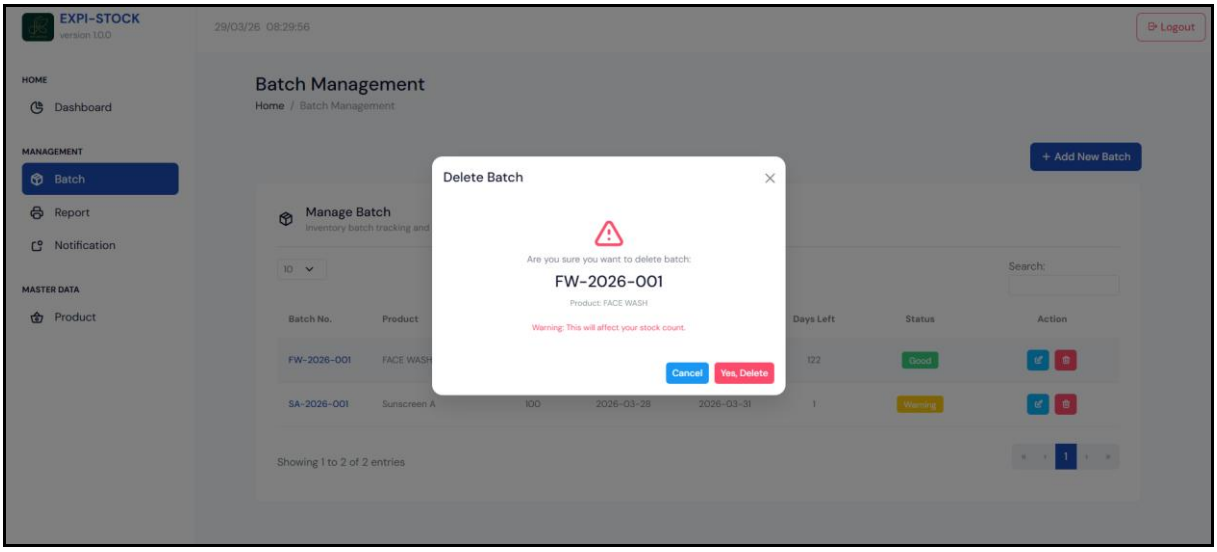
- Batch page - displays expiry status for each batch.



- Add Batch page.

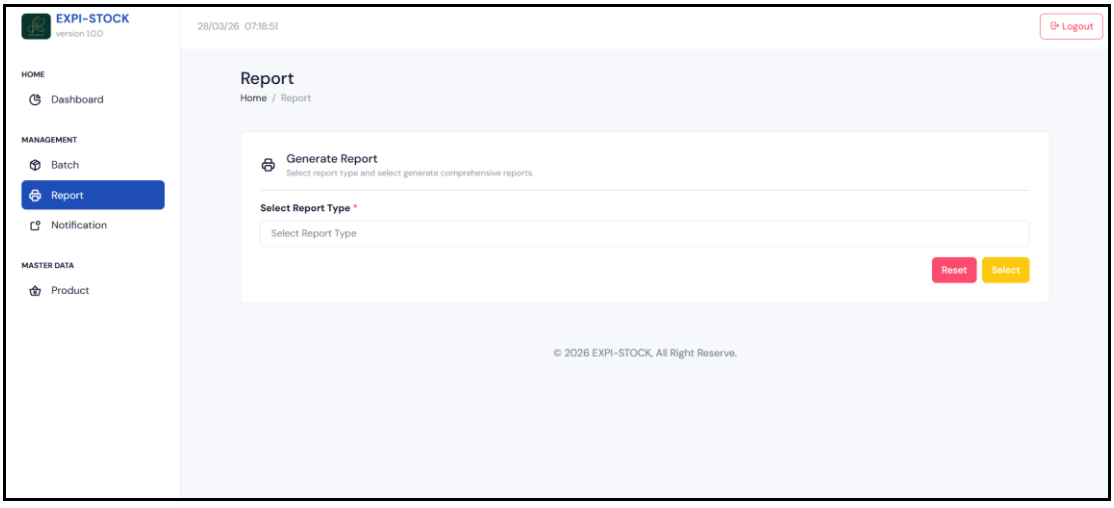


- Edit Batch page.

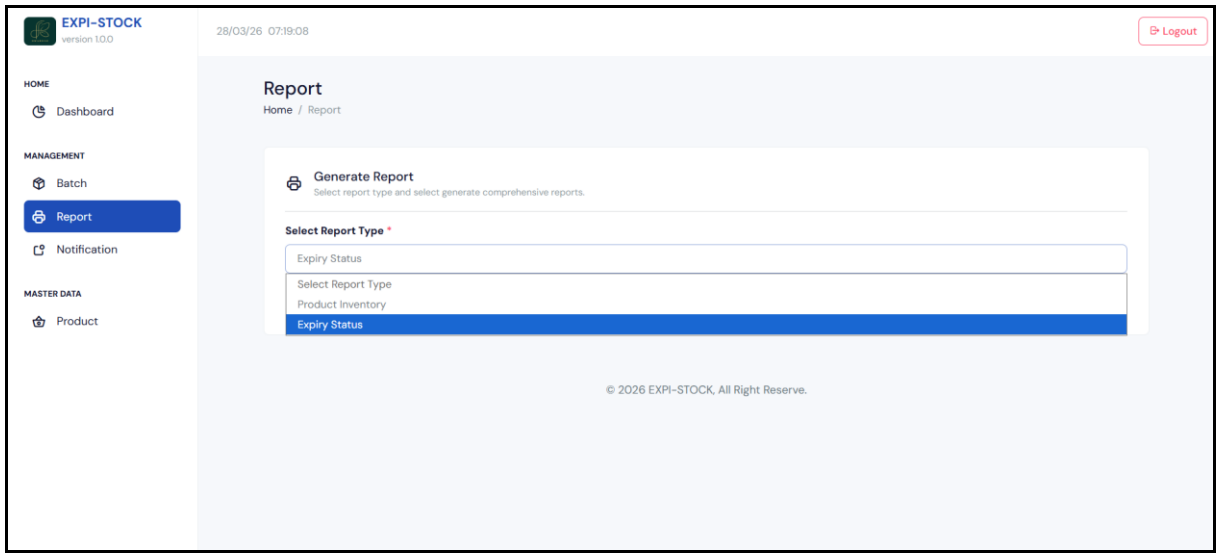


- Delete Batch confirmation popup.

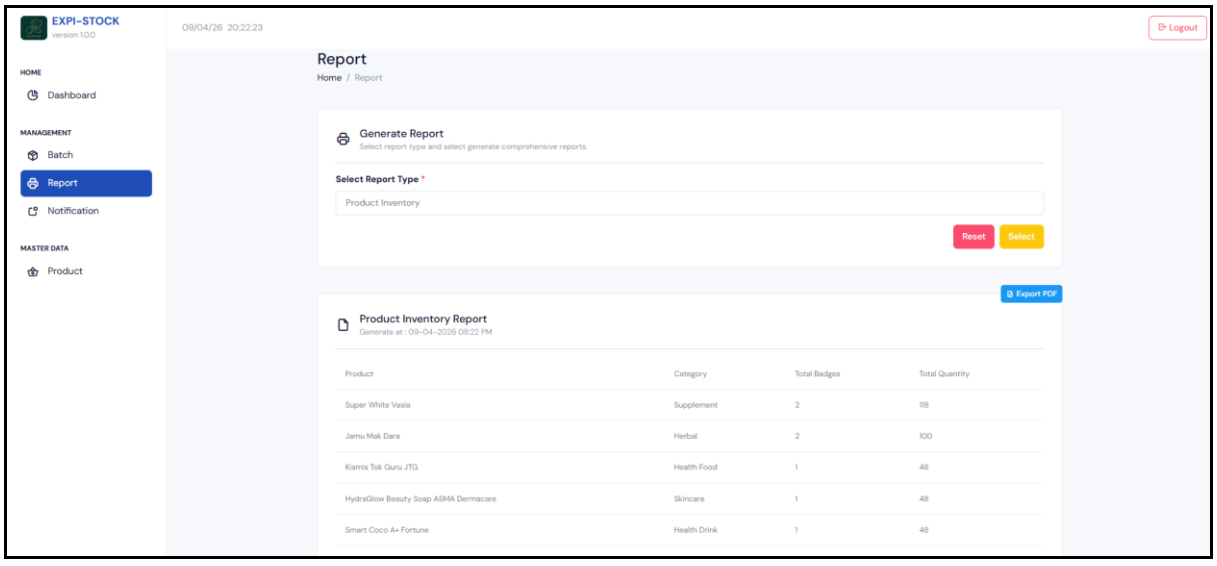
Report



- Report page overview.



- Report page after selecting report type.



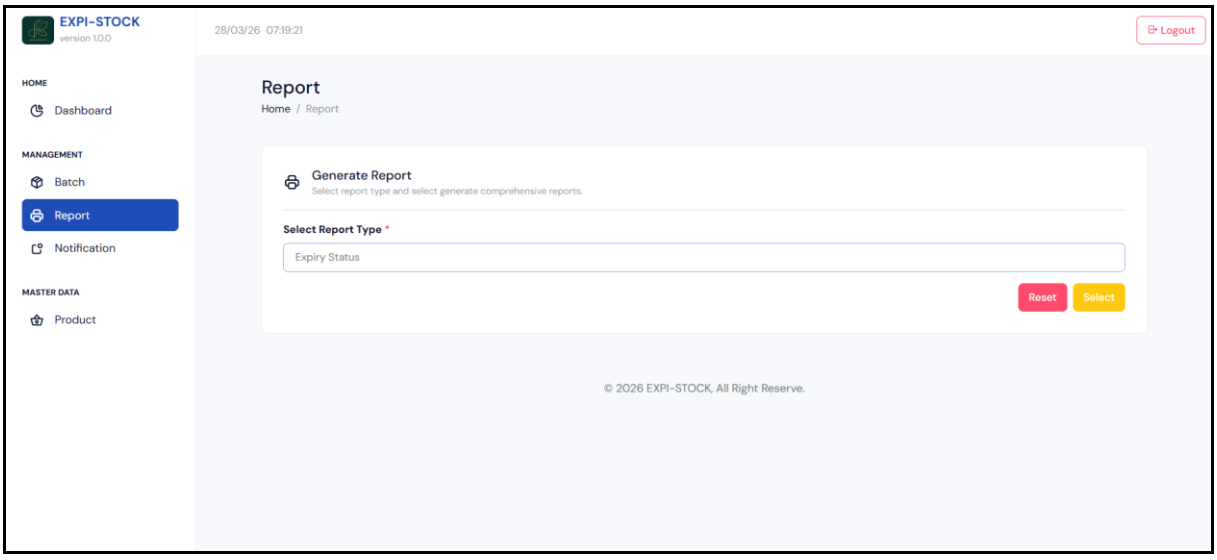
- Report page after selecting Product Inventory report.

Product Inventory

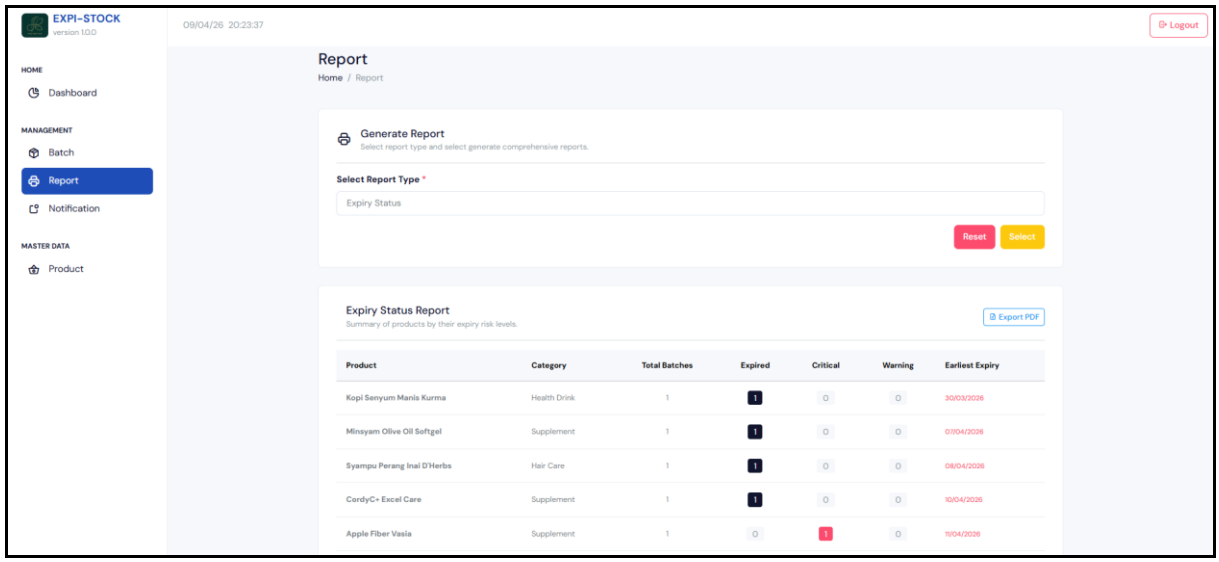
Generated at: 09/04/2026 07:55 PM

No	Product	Category	Total Batches	Total Quantity
1	Apple Fiber Vasia	Supplement	1	24
2	Orange Fiber Vasia	Supplement	1	24
3	Super White Vasia	Supplement	2	118
4	Jamu Mak Dara	Herbal	2	100
5	Gel Mandian Bidara Arab JRM	Self-Care	1	36
6	Lotion Mustajab Extreme Hot Dunia Herbs	Self-Care	1	24
7	Sacha Inchi Mandian Shifa Herb	Self-Care	1	30
8	Kismis Tok Guru JTG	Health Food	1	48
9	Serbuk Halia Madu AMRAN	Health Drink	1	36
10	Minsyam Olive Oil Softgel	Supplement	1	20
11	Minsyam Black Seed Oil Capsule	Supplement	1	20
12	Garlic Nurich 300mg	Supplement	1	10
13	CordyC+ Excel Care	Supplement	1	12
14	Berryfull Mixed Berry 20ml	Health Drink	1	6
15	Air Tonik Ginseng PETANI	Health Drink	1	3

- PDF report after clicking Export PDF.
-



- Report page after selecting Expiry Status report type.



◦ Report page after clicking Select for Expiry Status.

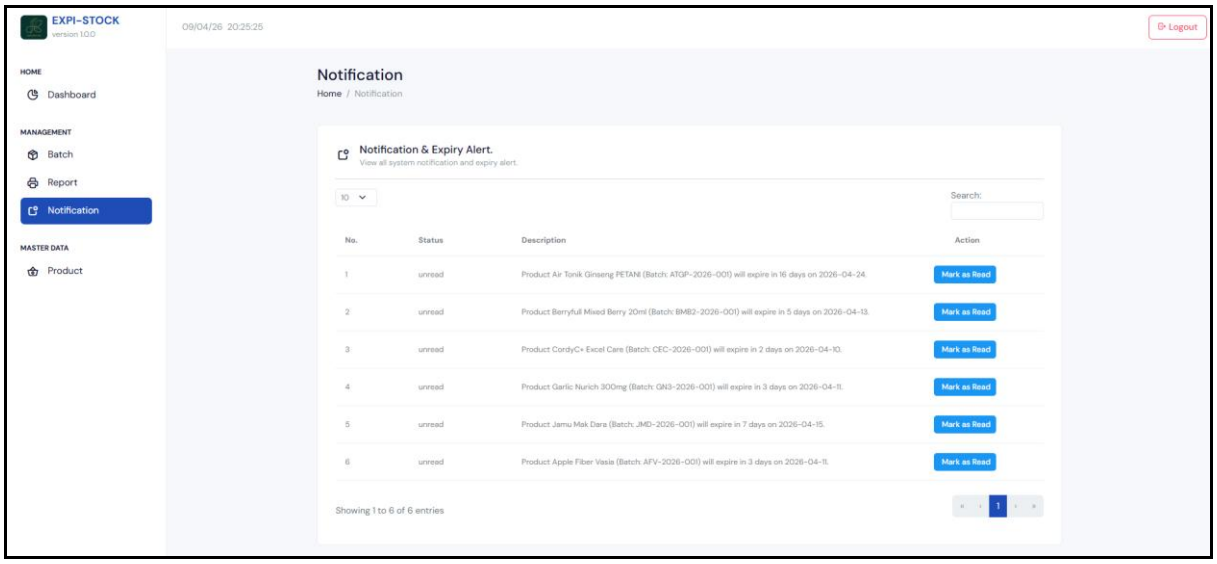
EXPIRY STATUS

Generated Date: 09/04/2026 07:57 PM

NO	PRODUCT NAME	CATEGORY	TOTAL BATCHES	EXPIRED	CRITICAL	WARNING	EARLIEST EXPIRY
1	Apple Fiber Vasia	Supplement	1	0	1	0	11/04/2026
2	Orange Fiber Vasia	Supplement	1	0	0	0	24/12/2027
3	Super White Vasia	Supplement	2	0	0	0	09/07/2027
4	Jamu Mak Dara	Herbal	2	0	0	1	15/04/2026
5	Gel Mandian Bidara Arab JRM	Self-Care	1	0	0	0	01/07/2027
6	Lotion Mustajab Extreme Hot Dunia Herbs	Self-Care	1	0	0	0	01/04/2028
7	Sacha Inchi Mandian Shifa Herb	Self-Care	1	0	0	0	01/01/2027
8	Kismis Tok Guru JTG	Health Food	1	0	0	0	30/12/2026
9	Serbuk Halia Madu AMRAN	Health Drink	1	0	0	0	30/05/2027
10	Minsyam Olive Oil Softgel	Supplement	1	1	0	0	07/04/2026
11	Minsyam Black Seed Oil Capsule	Supplement	1	0	0	0	01/09/2027
12	Garlic Nurich	Supplement	1	0	1	0	11/04/2026

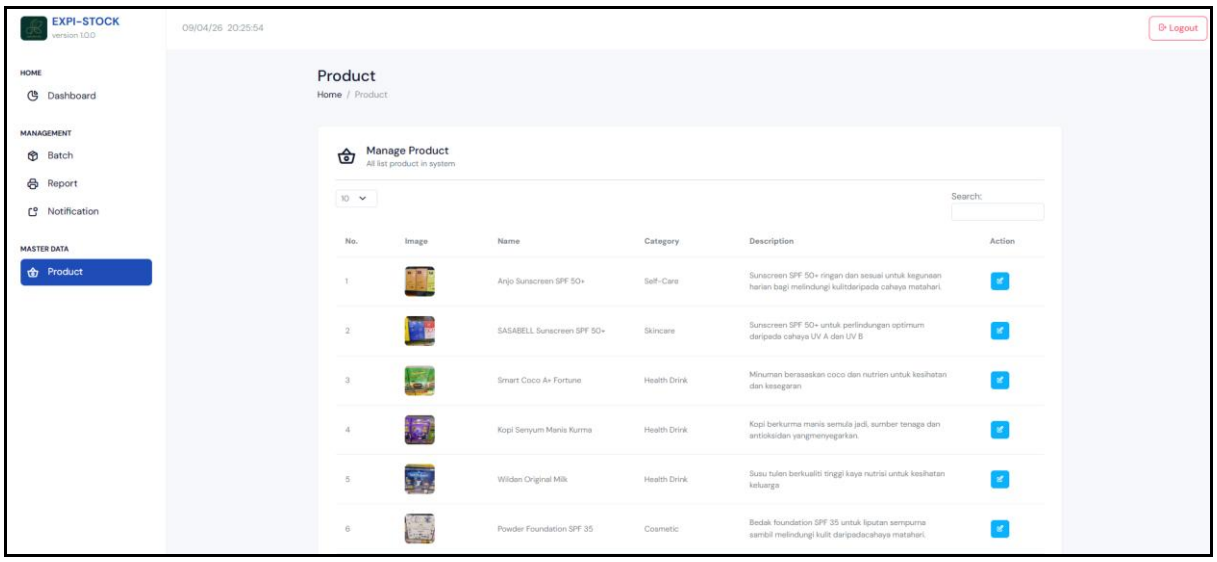
◦ Expiry Status PDF report after clicking Export PDF.

Notification

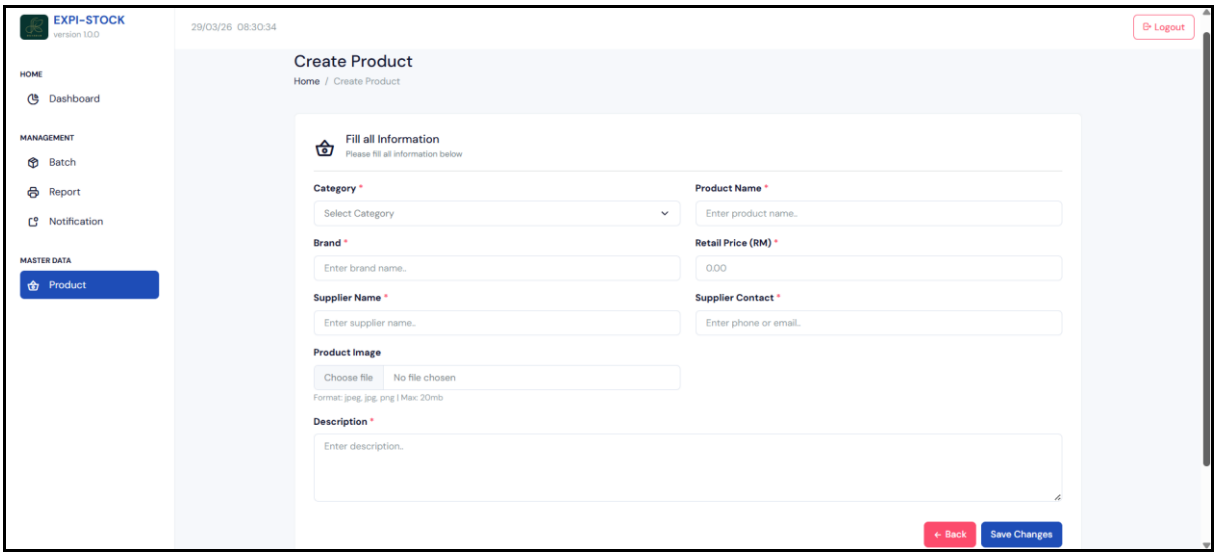


- Notification panel - alerts user to critical, warning, or expired products.
- Users also receive these alerts via their registered email.

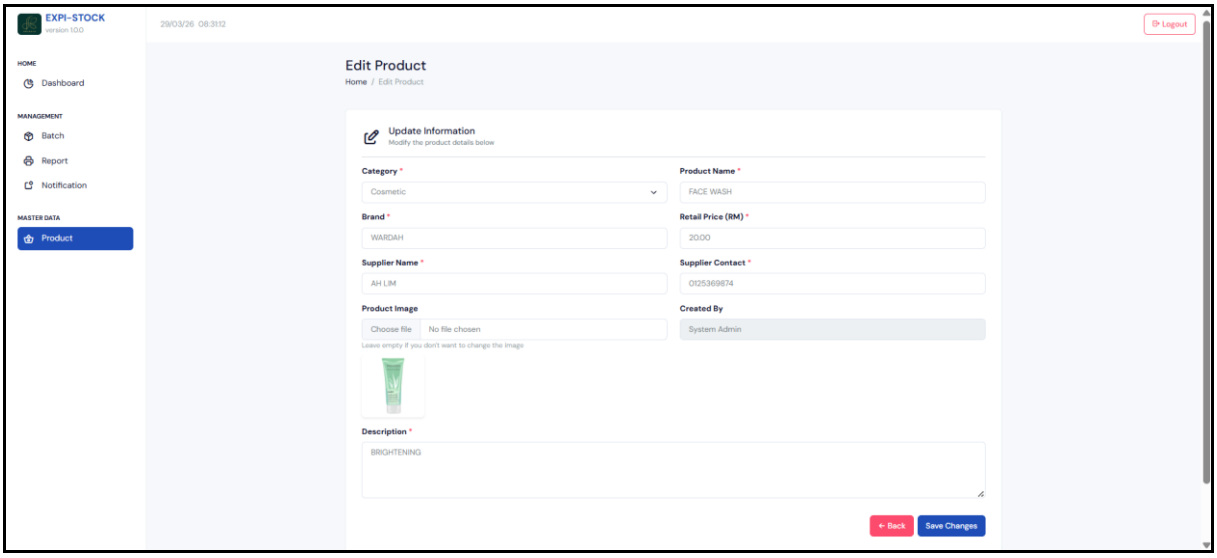
Product Management



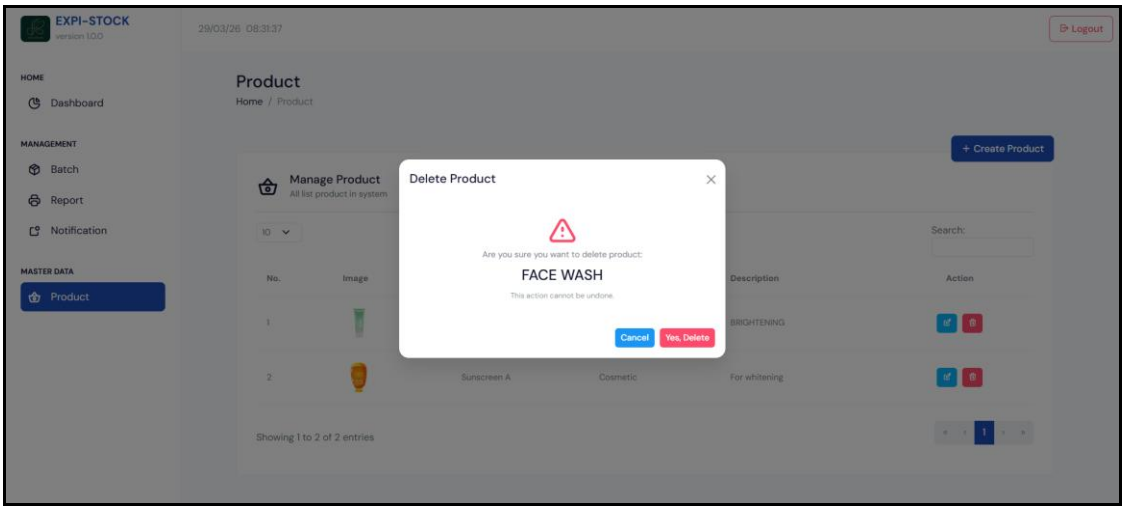
- Product page - Staff can add, edit, and delete products.



- Add Product page.

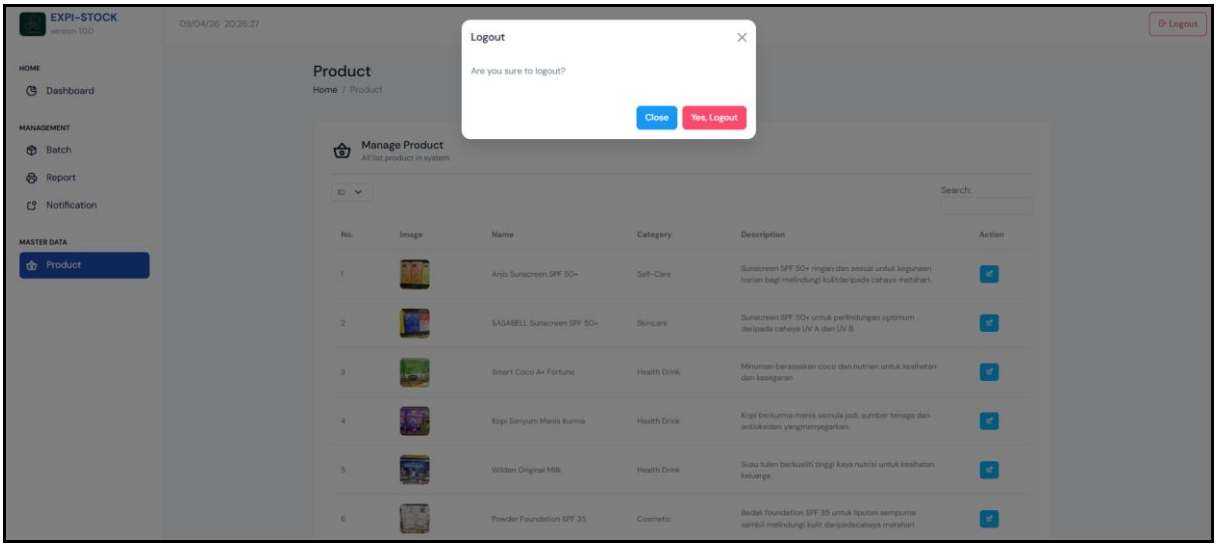


- Edit Product page.



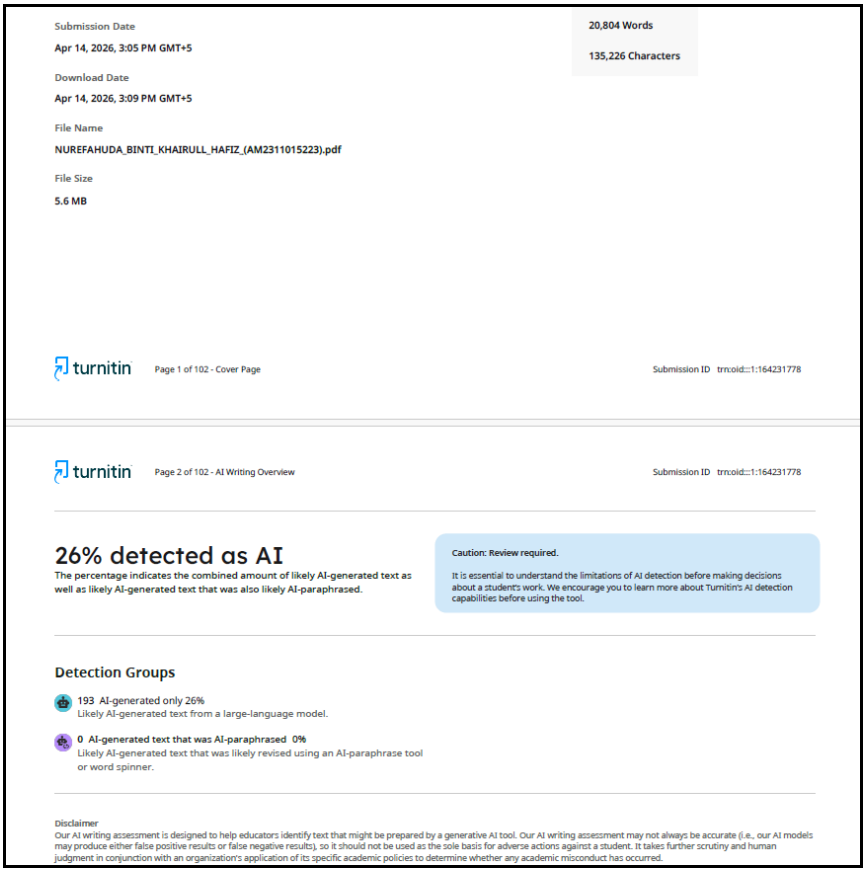
- Delete Product confirmation popup.

Logout



- Logout confirmation popup. After confirming, the login page is displayed.

Appendix C – Turnitin Result



Submission Date
Apr 14, 2026, 3:05 PM GMT+5

Download Date
Apr 14, 2026, 3:09 PM GMT+5

File Name
NUREFAHUDA_BINTI_KHAIRULL_HAFIZ_(AM2311015223).pdf

File Size
5.6 MB

20,804 Words

135,226 Characters

turnitin Page 1 of 102 - Cover Page Submission ID trn:oid:::1:164231778

turnitin Page 2 of 102 - AI Writing Overview Submission ID trn:oid:::1:164231778

26% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

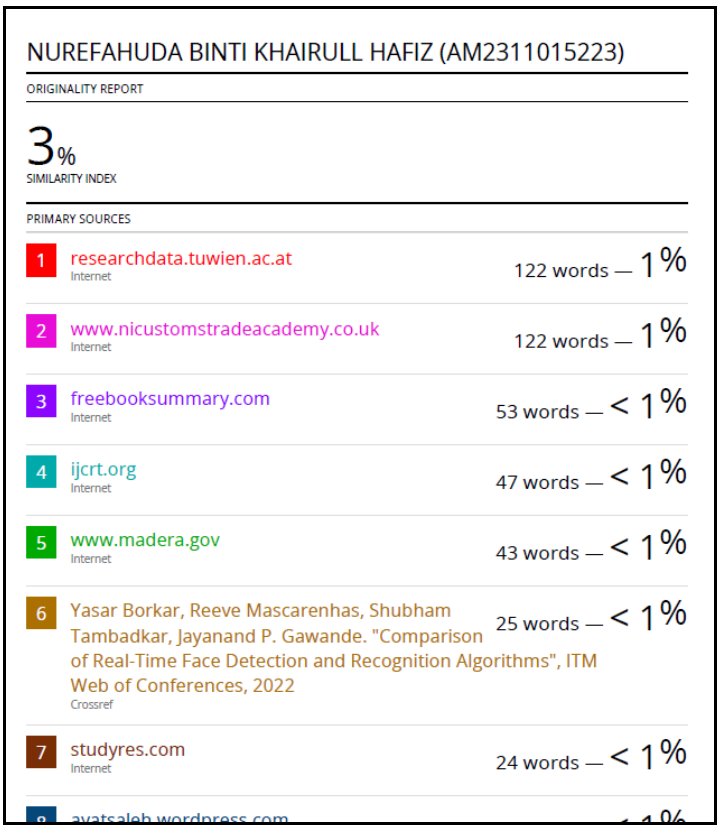
It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

193 AI-generated only 26%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.



NUREFAHUDA BINTI KHAIRULL HAFIZ (AM2311015223)

ORIGINALITY REPORT

3%
SIMILARITY INDEX

PRIMARY SOURCES

1	researchdata.tuwien.ac.at Internet	122 words — 1%
2	www.nicustomstradeacademy.co.uk Internet	122 words — 1%
3	freebooksummary.com Internet	53 words — < 1%
4	ijcrt.org Internet	47 words — < 1%
5	www.madera.gov Internet	43 words — < 1%
6	Yasar Borkar, Reeve Mascarenhas, Shubham Tambadkar, Jayanand P. Gawande. "Comparison of Real-Time Face Detection and Recognition Algorithms", ITM Web of Conferences, 2022 Crossref	25 words — < 1%
7	studyres.com Internet	24 words — < 1%
8	ayatsaleh.wordpress.com	< 1%

Appendix D – Log Book

CT203/BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN BUSINESS COMPUTING

FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

BUSINESS COMPUTING PROJECT 1

(FYP4094)

LOG BOOK

STUDENT'S NAME : NUREFAHUDA BINTI KHAIRULL HAFIZ

ID NO. : AM2311015223

SUPERVISOR : DR. MOHD NOOR AFIQ BIN RAMLEE

PROJECT TITLE : EXPI-STOCK: BUSINESS INVENTORY MANAGEMENT SYSTEM

CT203/BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN BUSINESS COMPUTING

Date/ Week		Agenda	Next Agenda	Signature (Supervisor)
7/8/2025	1	Find potential Supervisors	Finalize topic and title with SV	MOHD NOOR AFIQ BIN RAMLEE Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
11/8/2025	2	Discuss the topic and title with SV	Certify the system's functions with SV	MOHD NOOR AFIQ BIN RAMLEE Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
19/8/2025	3	Consult with SV regarding the system's functions.	Review Project Brief Proposal with SV	MOHD NOOR AFIQ BIN RAMLEE Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
29/8/2025	4	Discussion and feedback on Project Brief Proposal with SV.	Evaluate Chapter 1 with SV	MOHD NOOR AFIQ BIN RAMLEE Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
4/9/2025	5	Revise Chapter 1 based on SV's feedback	Discuss Chapter 2 content with SV	MOHD NOOR AFIQ BIN RAMLEE Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
9/9/2025	6	Incorporate SV's comments into Chapter 2	Finalize Chapter 2 with SV	MOHD NOOR AFIQ BIN RAMLEE Lecturer Faculty of Computing & Multimedia Universiti Poly-Tech Malaysia
24/9/2025	7	Discussion on Feedback and Corrections	Review Chapter 3 with SV	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
1/10/2025	8	Discusson and feedback on Chapter 3	Review chapter 3 that has been corrected.	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
8/10/2025	9	Incorporate SV's comments into Chapter 3	Discuss questionnaire content with SV	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
16/10/2025	10	Consult with SV regarding the questionnaire.	Evaluate Chapter 4 with SV	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

CT203/BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN BUSINESS COMPUTING

24/10/2025	11	Revise chapter 4 based on SV's Feedback	Review chapter 5 with SV	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
31/10/2025	12	Discussion ~~on feedba~~ and feedback on chapter 5	Discuss about Presentation with SV	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
7/11/2025	13	Consult with SV regarding Slide Presentation.	Make final touch-ups of slide presentation	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
10/11/2025	14	Presentation	make final touch-ups to report and compile all documents for submissio	DR. MOHD NOOR AFIQ BIN RAMLEE Pensyarah Kanan Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

CT203 / BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN BUSINESS COMPUTING

FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

BUSINESS COMPUTING PROJECT 2

FYP4105

LOG BOOK

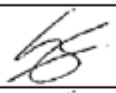
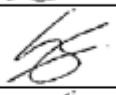
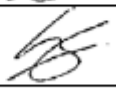
PROJECT TITLE : EXPI-STOCK: BUSINESS INVENTORY MANAGEMENT SYSTEM

STUDENT'S NAME : NUREFAHUDA BINTI KHAIRULL HAFIZ

ID NO. : AM2311015223

SUPERVISOR : NOR AZURA BINTI SALLEH @ OMAR

CT203 / BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN BUSINESS COMPUTING

Week		Agenda	Next Agenda	Signature (Supervisor / Coordinator)
25/12/2025	1	Consultation with sv to evaluate & verify the FYP 1.	Submission flowchart to sv.	
1/1/2026	2	System flowchart discussion with sv.	Wireframe design submission to sv.	
8/1/2026	3	Wireframe design discussion with sv.	Data dictionary submission to sv.	
12/1/2026	4	Data dictionary discussion with sv.	ERD submission to sv.	
21/1/2026	5	ERD discussion with sv.	Data Flow Diagram submission to sv.	
28/1/2026	6	Data Flow Dagram discussion with sv.	Flow of the System Diagram submission to sv.	
5/2/2026	7	Flow of the System Diagram discussion with sv.	Check on backend of the system with sv.	
13/2/2026	8	Discuss about backend of the system with sv.	Show to sv the testing of the backend.	
19/2/2026	9	Discuss about the result of testing of the backend with sv.	Show to sv the backend that fixed.	
25/2/2026	10	Check on backend that fixed with sv.	Show to sv the frontend of the sytem.	
5/3/2026	11	Discuss with sv the frontend of the system.	Check on function on the frontend with sv.	
13/3/2026	12	Discuss with sv the functional of the front end.	Show to sv the the frontend that fixed.	
20/3/2026	13	Check on frontend that fixed with sv.	Presentation full system.	
MID-TERM BREAK				
30/3/2026	14	Presentation full system.		

References

1. Addverb (2025) FIFO inventory management: Definition, benefits, and implementation. Available at: https://www.addverb.com
2. Agans, D.J. (2021) Debugging: The 9 indispensable rules for finding even the most elusive software and hardware problems. 2nd edn. New York: AMACOM.
3. Ahmed, T. and Ahmad, S. (2023) 'Risk management in waterfall software development: A systematic review', Journal of Software Engineering Research and Development, 11(2), pp. 145-162. doi: 10.1186/s40411-023-00134-5.
4. Apache Software Foundation (2024) Apache JMeter. Available at: https://jmeter.apache.org/
5. Arirms (2024) Retail industry in Malaysia: Stats, challenges & solutions. Available at: https://arirms.com/malaysia-retail-industry
6. Ash, J.S., Sittig, D.F., Poon, E.G., Guappone, K., Campbell, E. and Dykstra, R.H. (2020) 'The extent and importance of unintended consequences related to computerized provider order entry', Journal of the American Medical Informatics Association, 27(1), pp. 114-123. doi: 10.1093/jamia/ocz157.
7. A2Z Cloud (2019) Zoho security: GDPR, SOC 2, and compliance. Available at: https://www.zoho.com/compliance.html
8. Balaji, S. and Murugaiyan, M.S. (2021) 'Waterfall vs agile: A comparative study on software development methodologies', International Journal of Emerging Technology and Advanced Engineering, 11(5), pp. 29-35. Available at: https://www.ijetae.com/Volume11Issue5.html
9. Barth, A., Jackson, C. and Mitchell, J.C. (2020) 'Robust defenses for cross-site request forgery', Proceedings of the ACM Conference on Computer and Communications Security, pp. 75-88. doi: 10.1145/1455770.1455782.
10. Bass, L., Clements, P. and Kazman, R. (2021) Software architecture in practice. 4th edn. Upper Saddle River, NJ: Addison-Wesley.
11. Bassil, Y. (2022) 'A simulation model for the waterfall software development life cycle', International Journal of Engineering & Technology, 11(4), pp. 742-749. doi: 10.14419/ijet.v11i4.32145.
12. Beck, K. (2022) Test-driven development: By example. 2nd edn. Boston, MA: Addison-Wesley.
13. Bergmann, S. (2024) PHPUnit documentation. Available at: https://phpunit.de/documentation.html
14. Bonnardel, N., Piolat, A. and Le Bigot, L. (2020) 'The impact of colour on website appeal and users' cognitive processes', Displays, 62, 101946. doi: 10.1016/j.displa.2020.101946.
15. Braun, V. and Clarke, V. (2022) Thematic analysis: A practical guide. London: SAGE Publications.
16. Brooke, J. (2020) 'SUS: A quick and dirty usability scale', Usability Evaluation in Industry, pp. 189-194. London: CRC Press.
17. Bryman, A. (2024) Social research methods. 6th edn. Oxford: Oxford University Press.
18. Buxton, B. (2020) Sketching user experiences: Getting the design right and the right design. 2nd edn. San Francisco, CA: Morgan Kaufmann.

19. Campanelli, A.S. and Parreiras, F.S. (2020) 'Agile methods tailoring: A systematic literature review', Journal of Systems and Software, 167, 110621. doi: 10.1016/j.jss.2020.110621.
20. Candidroot (2025) Odoo integrations and customization. Available at: https://www.candidroot.com/odoo-customization
21. Castillo-Montoya, M. (2021) 'Preparing for interview research: The interview protocol refinement framework', The Qualitative Report, 26(5), pp. 1551-1563. doi: 10.46743/2160-3715/2021.4763.
22. Chacon, S. and Straub, B. (2022) Pro Git. 3rd edn. Berkeley, CA: Apress. Available at: https://git-scm.com/book/en/v2
23. Chain Store Age (2024) Study: Global retail losses due to inventory 'distortion' hit $1.77 trillion. Available at: https://chainstoreage.com/study-global-retail-losses-due-inventory-distortion-hit-177-trillion
24. Chen, P.P. (2020) 'The entity-relationship model: Toward a unified view of data', ACM Transactions on Database Systems, 45(1), pp. 1-23. doi: 10.1145/3380786.
25. Clark, J. (2020) Designing for touch. 2nd edn. New York: A Book Apart.
26. Codd, E.F. (2020) 'A relational model of data for large shared data banks', Communications of the ACM, 63(1), pp. 64-69. doi: 10.1145/3360467.
27. Cohn, M. (2020) Succeeding with agile: Software development using Scrum. 2nd edn. Upper Saddle River, NJ: Addison-Wesley.
28. Conventional Commits (2024) Conventional commits specification. Version 1.0.0. Available at: https://www.conventionalcommits.org/
29. Creswell, J.W. and Clark, V.L.P. (2023) Designing and conducting mixed methods research. 4th edn. Thousand Oaks, CA: SAGE Publications.
30. Creswell, J.W. and Creswell, J.D. (2023) Research design: Qualitative, quantitative, and mixed methods approaches. 6th edn. Thousand Oaks, CA: SAGE Publications.
31. Crockford, D. (2020) 'The application/json media type for JavaScript Object Notation (JSON)', Internet Engineering Task Force (IETF) RFC 8259. Available at: https://datatracker.ietf.org/doc/html/rfc8259
32. CYBRA (2024) Obsolete inventory management and financial impact. Available at: https://www.cybra.com
33. Ded9 (2022) Why choose PHP language? Available at: https://ded9.com/why-choose-php-language/
34. Date, C.J. (2020) Database design and relational theory: Normal forms and all that jazz. 2nd edn. Sebastopol, CA: O'Reilly Media.
35. DeJonckheere, M. and Vaughn, L.M. (2020) 'Semistructured interviewing in primary care research: A balance of relationship and rigour', Family Medicine and Community Health, 7(2), e000057. doi: 10.1136/fmch-2018-000057.
36. Denzin, N.K. (2020) The research act: A theoretical introduction to sociological methods. New York: Routledge.

37. Design Council (2020) The double diamond: A universally accepted depiction of the design process. Available at: https://www.designcouncil.org.uk/our-work/skills-learning/tools-frameworks/framework-for-innovation-design-councils-evolved-double-diamond/
38. Driessen, V. (2020) 'A successful Git branching model', nvie.com. Available at: https://nvie.com/posts/a-successful-git-branching-model/
39. Dumas, M., La Rosa, M., Mendling, J. and Reijers, H.A. (2023) Fundamentals of business process management. 3rd edn. Berlin: Springer.
40. ECMA International (2023) ECMAScript 2023 language specification. ECMA-262, 14th edn. Geneva: ECMA International. Available at: https://www.ecma-international.org/publications-and-standards/standards/ecma-262/
41. Elliot, A.J. and Maier, M.A. (2021) 'Color psychology: Effects of perceiving color on psychological functioning in humans', Annual Review of Psychology, 72, pp. 95-120. doi: 10.1146/annurev-psych-010419-051052.
42. Elmasri, R. and Navathe, S.B. (2023) Fundamentals of database systems. 8th edn. Harlow: Pearson Education.
43. Envertis (2024) Reasons why your business needs Odoo ERP software in 2024. Available at: https://www.envertis.com/18-compelling-reasons-why-your-business-needs-odoo-erp-software-in-2024/
44. Erickson, L. (2024) Introduction to SQL and MySQL for web applications. Available at: https://www.mysqltutorial.org/getting-started-with-mysql/
45. Few, S. (2021) Information dashboard design: Displaying data for at-a-glance monitoring. 3rd edn. El Dorado Hills, CA: Analytics Press.
46. Field, A. (2024) Discovering statistics using IBM SPSS Statistics. 6th edn. London: SAGE Publications.
47. Fielding, R.T. (2020) 'Architectural styles and the design of network-based software architectures', Doctoral dissertation, University of California, Irvine. Available at: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
48. Fit Small Business (2023) Sortly review. Available at: https://fitsmallbusiness.com/sortly-review/
49. Forsberg, K. and Mooz, H. (2020) 'The relationship of system engineering to the project cycle', Engineering Management Journal, 32(3), pp. 156-167. doi: 10.1080/10429247.2020.1756792.
50. Fortune (2024) Why retail's $100 billion 'shrink' crisis may not be all about shoplifting. Available at: https://fortune.com/2024/01/09/why-retail-100-billion-shrink-crisis-may-shoplifting/
51. Fowler, F.J. (2022) Survey research methods. 6th edn. Thousand Oaks, CA: SAGE Publications.
52. Fowler, M. (2020) Refactoring: Improving the design of existing code. 2nd edn. Boston, MA: Addison-Wesley.
53. Fowler, M. (2022a) Patterns of enterprise application architecture. 2nd edn. Boston, MA: Addison-Wesley.

54. Fowler, M. (2022b) 'Technical debt', Martin Fowler's Blog. Available at: https://martinfowler.com/bliki/TechnicalDebt.html
55. Fowler, M. and Foemmel, M. (2020) 'Continuous integration', ThoughtWorks. Available at: https://martinfowler.com/articles/continuousIntegration.html
56. Future Market Insights (2025) Inventory management software market growth 2025-2035. Available at: https://www.futuremarketinsights.com/reports/inventory-management-software-market
57. Garrett, J.J. (2021) The elements of user experience: User-centered design for the web and beyond. 3rd edn. Berkeley, CA: New Riders.
58. GeeksforGeeks (2025) Requirements gathering in software development. Available at: https://www.geeksforgeeks.org/software-engineering/requirements-gathering-introduction-processes-benefits-and-tools/
59. George, D. and Mallery, P. (2021) IBM SPSS statistics 27 step by step: A simple guide and reference. 17th edn. New York: Routledge. doi: 10.4324/9781003205333.
60. GetApp (2025) Sortly inventory management software. Available at: https://www.getapp.com/operations-management-software/inventory-management/f/mobile-integration/
61. Google (2024) Firebase documentation. Available at: https://firebase.google.com/docs
62. Google (no date) Firebase. Available at: https://firebase.google.com
63. Haerder, T. and Reuter, A. (2021) 'Principles of transaction-oriented database recovery', ACM Computing Surveys, 53(1), pp. 1-40. doi: 10.1145/3442355.
64. Halfond, W.G., Viegas, J. and Orso, A. (2020) 'A classification of SQL injection attacks and countermeasures', IEEE Symposium on Security and Privacy, pp. 65-81. doi: 10.1109/ISSREW.2020.00014.
65. Hambling, B. and Goethem, P.V. (2020) User acceptance testing: A step-by-step guide. 2nd edn. Swindon: BCS, The Chartered Institute for IT.
66. HashMicro (2024) 16 best warehouse management system in Malaysia (2025). HashMicro Blog. Available at: https://www.hashmicro.com/my/blog/best-wms-warehouse-management-system/
67. Havi Technology (2024) Odoo inventory management: fundamentals and features. Available at: https://havi.com.au/blog/odoo-inventory-management
68. Henry, S.L., Abou-Zahra, S. and Brewer, J. (2021) 'The role of accessibility in a universal web', Communications of the ACM, 64(2), pp. 44-51. doi: 10.1145/3433539.
69. Hoffman, P. (2021) 'SMTP service extension for secure SMTP over Transport Layer Security', Internet Engineering Task Force (IETF) RFC 3207. Available at: https://datatracker.ietf.org/doc/html/rfc3207
70. Humble, J. and Farley, D. (2020) Continuous delivery: Reliable software releases through build, test, and deployment automation. 2nd edn. Upper Saddle River, NJ: Addison-Wesley.
71. Hunt, A. and Thomas, D. (2020) The pragmatic programmer: Your journey to mastery. 20th anniversary edn. Boston, MA: Addison-Wesley.

72. IEEE (2020) IEEE standard for software requirements specifications. IEEE Std 830-2020. New York: Institute of Electrical and Electronics Engineers.
73. IHL Group (2023a) 2023 out-of-stocks and overstocks matrix. Inside Retail Asia. Available at: https://insideretail.asia/2025/09/05/overstock-is-a-us-554-billion-problem-for-retail-these-platforms-aim-to-solve-it/
74. IHL Group (2023b) Inventory distortion: the good, the bad, the ugly. Available at: https://www.retailtouchpoints.com/features/industry-insights/ihl-study-inventory-distortion-will-cost-retailers-1-77-trillion-in-2023
75. IHL Group (2024) Fixing inventory distortion – Are we there yet? IHL Services. Available at: https://www.ihlservices.com/product/fixing-inventory-distortion-are-we-there-yet/
76. Inside Retail Asia (2025) Overstock is a US $554 billion problem for retail: these platforms aim to solve it. Available at: https://insideretail.asia/2025/09/05/overstock-is-a-us-554-billion-problem-for-retail-these-platforms-aim-to-solve-it/
77. Institute of Project Management. (2020). Work breakdown structure: A project management essential. Institute of Project Management.
78. Institute of Electrical and Electronics Engineers (2023) IEEE standard for system, software, and hardware verification and validation. IEEE Std 1012-2023. New York: IEEE.
79. InVue (2024) 6 retail shrinkage statistics and what they mean for your business. Available at: https://invue.com/resource-center/blog/6-retail-shrinkage-statistics
80. ISO (2022) Information processing: Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts. ISO 5807:2022. Geneva: International Organization for Standardization.
81. Joshi, A., Kale, S., Chandel, S. and Pal, D.K. (2021) 'Likert scale: Explored and explained', British Journal of Applied Science & Technology, 7(4), pp. 396-403. doi: 10.9734/BJAST/2021/v7i430396.
82. Kalbach, J. (2021) Mapping experiences: A complete guide to creating value through journeys, blueprints, and diagrams. 2nd edn. Sebastopol, CA: O'Reilly Media.
83. Kallio, H., Pietilä, A.M., Johnson, M. and Kangasniemi, M. (2020) 'Systematic methodological review: Developing a framework for a qualitative semi-structured interview guide', Journal of Advanced Nursing, 76(12), pp. 3410-3420. doi: 10.1111/jan.14569.
84. Kirda, E., Kruegel, C., Vigna, G. and Jovanovic, N. (2020) 'Noxes: A client-side solution for mitigating cross-site scripting attacks', Proceedings of the ACM Symposium on Applied Computing, pp. 330-337. doi: 10.1145/1363686.1363762.
85. Klensin, J. (2021) 'Simple Mail Transfer Protocol', Internet Engineering Task Force (IETF) RFC 5321. Available at: https://datatracker.ietf.org/doc/html/rfc5321
86. Krasner, G.E. and Pope, S.T. (2020) 'A description of the model-view-controller user interface paradigm in the Smalltalk-80 system', Journal of Object Technology, 19(3), pp. 1-11. doi: 10.5381/jot.2020.19.3.a1.

87. Kumar, R. (2022) Research methodology: A step-by-step guide for beginners. 5th edn. London: SAGE Publications.
88. Labrecque, L.I. and Milne, G.R. (2021) 'To be or not to be different: Exploration of norms and benefits of color differentiation in the marketplace', Marketing Letters, 32, pp. 165-176. doi: 10.1007/s11002-020-09541-9.
89. Leff, A. and Rayfield, J.T. (2021) 'Web-application development using the Model/View/Controller design pattern', IEEE Enterprise Distributed Object Computing Conference, pp. 118-127. doi: 10.1109/EDOC.2021.00024.
90. Lientz, B.P. and Swanson, E.B. (2020) Software maintenance management. 2nd edn. Reading, MA: Addison-Wesley.
91. Lidwell, W., Holden, K. and Butler, J. (2022) Universal principles of design. 3rd edn. Beverly, MA: Rockport Publishers.
92. Lightspeed (2025) FIFO stock rotation methods for retail: Best practices and benefits. Available at: https://www.lightspeedhq.com
93. Marcotte, E. (2020) Responsive web design. 2nd edn. New York: A Book Apart.
94. McCabe, T.J. (2020) 'A complexity measure', IEEE Transactions on Software Engineering, 46(4), pp. 308-320. doi: 10.1109/TSE.1976.233837.
95. McGuire, D. (2024). Risk management in software development projects. Available at: https://www.projectmanager.com/blog/risk-management-process-steps
96. Mehta, R. and Zhu, R. (2020) 'Blue or red? Exploring the effect of color on cognitive task performances', Science, 323(5918), pp. 1226-1229. doi: 10.1126/science.1169144.
97. Meszaros, G. (2020) xUnit test patterns: Refactoring test code. 2nd edn. Upper Saddle River, NJ: Addison-Wesley.
98. Meta Open Source (2024) Jest: Delightful JavaScript testing. Available at: https://jestjs.io/
99. Microsoft (2022) Visual Studio Code – Code editing redefined. Available at: https://code.visualstudio.com/
100. Microsoft (2025) Visual Studio Code - Code editing. Redefined. Available at: https://code.visualstudio.com/
101. Mordor Intelligence (2024) Malaysia retail market size & share analysis - Growth trends & forecasts (2024-2029). Available at: https://www.mordorintelligence.com
102. Mordor Intelligence (2025) Retail industry in Malaysia - market report, size & outlook. Available at: https://www.mordorintelligence.com/industry-reports/malaysian-retail-industry/market-size
103. MRPeasy (2024a) Automated inventory management – a quick guide. Available at: https://www.mrpeasy.com/blog/automated-inventory-management/
104. MRPeasy (2024b) Inventory shrinkage – Causes, consequences, and tips. Available at: https://www.mrpeasy.com/blog/inventory-shrinkage/

105. Munassar, N.M.A. and Govardhan, A. (2020) 'A comparison between five models of software engineering', International Journal of Computer Science Issues, 17(5), pp. 94-101. Available at: https://www.ijcsi.org/papers/IJCSI-17-5-94-101.pdf
106. Myers, G.J., Sandler, C. and Badgett, T. (2024) The art of software testing. 4th edn. Hoboken, NJ: John Wiley & Sons.
107. Netstock (2024) 2024 inventory management benchmark report. Available at: https://www.netstock.com/research/inventory-management-report/
108. Nielsen, J. (2020) 'Usability 101: Introduction to usability', Nielsen Norman Group. Available at: https://www.nngroup.com/articles/usability-101-introduction-to-usability/
109. Nielsen, J. and Molich, R. (2020) 'Heuristic evaluation of user interfaces', Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 249-256. doi: 10.1145/97243.97281.
110. Norman, D.A. (2024) The design of everyday things. Revised and expanded edn. New York: Basic Books.
111. Nuseibeh, B. and Easterbrook, S. (2020) 'Requirements engineering: A roadmap', Future of Software Engineering, pp. 35-46. doi: 10.1145/336512.336523.
112. Nunnally, J.C. and Bernstein, I.H. (2020) Psychometric theory. 4th edn. New York: McGraw-Hill.
113. Odoo (2025) Odoo inventory management system. Available at: https://www.odoo.com
114. Ohno, T. (2020) Toyota production system: Beyond large-scale production. 2nd edn. New York: Productivity Press.
115. Open-Source Integrators (2025) Odoo implementation benefits. Available at: https://www.opensourceintegrators.com/
116. Open Web Application Security Project (2024) OWASP top ten web application security risks. Available at: https://owasp.org/www-project-top-ten/
117. Oracle (2025) MySQL - The world's most popular open-source database. Available at: https://www.mysql.com/
118. Osmani, A. (2020) Learning JavaScript design patterns. 2nd edn. Sebastopol, CA: O'Reilly Media.
119. OWASP (2024) OWASP testing guide v5. Open Web Application Security Project. Available at: https://owasp.org/www-project-web-security-testing-guide/
120. OWS (2025) What is HTML? Learn to create a website with HTML. Available at: https://166tech.az/en/what-is-html-learn-to-create-a-website-with-html
121. Oxford Web Studio (2023) What is CSS and why is it important for web design? Available at: https://www.oxfordwebstudio.com/en/did-you-know/what-is-css
122. Patton, M.Q. (2023) Qualitative research and evaluation methods. 5th edn. Thousand Oaks, CA: SAGE Publications.

123. Petersen, K., Wohlin, C. and Baca, D. (2020) 'The waterfall model in large-scale development', Lecture Notes in Business Information Processing, 408, pp. 386-400. doi: 10.1007/978-3-030-64148-1_24.
124. PHP Documentation (2024) PHP manual. The PHP Group. Available at: https://www.php.net/manual/en/
125. Pigoski, T.M. (2020) Practical software maintenance: Best practices for managing your software investment. 2nd edn. New York: John Wiley & Sons.
126. Pohl, K. (2022) Requirements engineering: Fundamentals, principles, and techniques. 2nd edn. Berlin: Springer.
127. Portcities (2025) Odoo 19 features roadmap: the ultimate guide. Available at: https://portcities.net/blog/erp-and-odoo-insights-2/odoo-19-features-roadmap-205
128. Pravossoudovitch, K., Cury, F., Young, S.G. and Elliot, A.J. (2020) 'Is red the colour of danger? Testing an implicit red-danger association', Ergonomics, 63(5), pp. 623-628. doi: 10.1080/00140139.2020.1728398.
129. Pressman, R.S. and Maxim, B.R. (2024) Software engineering: A practitioner's approach. 10th edn. New York: McGraw-Hill Education.
130. Project Management Institute. (n.d.). What is project management? Available at: https://www.pmi.org/about/learn-about-pmi/what-is-project-management
131. Project Management Institute. (2021). A guide to the project management body of knowledge (PMBOK guide) (7th ed.). Project Management Institute.
132. ProjectManager.com. (2022). Work breakdown structure (WBS). Available at: https://www.projectmanager.com/guides/work-breakdown-structure
133. ProjectManager.com. (2025). What is a project schedule? Available at: https://www.projectmanager.com/blog/project-schedule
134. Provos, N. and Mazières, D. (2020) 'A future-adaptable password scheme', USENIX Annual Technical Conference, pp. 81-92. Available at: https://www.usenix.org/conference/usenixsecurity20/presentation/provos
135. Questudio (2025) Odoo's inventory management module: a comprehensive guide. Available at: https://questudio.com/blog/odoos-inventory-management-module-a-comprehensive-guide/
136. Rescorla, E. (2021) 'HTTP over TLS', Internet Engineering Task Force (IETF) RFC 2818. Available at: https://datatracker.ietf.org/doc/html/rfc2818
137. Rodriguez, F. (2020) Responsive email design: A complete guide to creating mobile-optimized emails. Available at: https://www.emailonacid.com/blog/article/email-development/responsive-email-design/
138. Ruparelia, N.B. (2021) 'Software development lifecycle models', ACM SIGSOFT Software Engineering Notes, 46(3), pp. 8-13. doi: 10.1145/3464984.3464987.
139. SaaS Tools Lab (2024) Sortly inventory management review. Available at: https://saastoolslab.com/sortly-review/

140. Saltzer, J.H. and Schroeder, M.D. (2020) 'The protection of information in computer systems', Proceedings of the IEEE, 108(1), pp. 8-35. doi: 10.1109/JPROC.2019.2957723.
141. Sandhu, R.S. and Samarati, P. (2021) 'Access control: Principles and practice', IEEE Communications Magazine, 59(9), pp. 40-48. doi: 10.1109/MCOM.001.94026.
142. Schwaber, K. and Sutherland, J. (2020) The Scrum guide: The definitive guide to Scrum. Available at: https://scrumguides.org/scrum-guide.html
143. SeleniumHQ (2024) Selenium WebDriver. Available at: https://www.selenium.dev/documentation/webdriver/
144. Smarttek Solutions (2023) Odoo security and compliance. Available at: https://smarttek.solutions/portfolio/odoo-inventory-accounting-manufacturing-case-study/
145. Software Connect (2024) Sortly inventory management system review. Available at: https://softwareconnect.com/reviews/sortly/
146. Sommerville, I. (2024) Software engineering. 11th edn. Harlow: Pearson Education.
147. Sortly (2025) Sortly pricing and features. Available at: https://www.sortly.com/
148. Souders, S. (2020) High performance web sites: Essential knowledge for front-end engineers. 2nd edn. Sebastopol, CA: O'Reilly Media.
149. Spaceo Technologies (2024) Waterfall methodology diagram. Available at: https://www.spaceo.ca/waterfall-model/
150. Stoica, M., Mircea, M. and Ghilic-Micu, B. (2020) 'Software development: Agile vs traditional', Informatica Economică, 24(2), pp. 37-48. doi: 10.12948/issn14531305/24.2.2020.04.
151. Taber, K.S. (2020) 'The use of Cronbach's alpha when developing and reporting research instruments in science education', Research in Science Education, 50(6), pp. 1273-1296. doi: 10.1007/s11165-016-9602-2.
152. Tavakol, M. and Dennick, R. (2020) 'Making sense of Cronbach's alpha', International Journal of Medical Education, 11, pp. 53-55. doi: 10.5116/ijme.5f96.442f.
153. TeamGantt. (2025). What is a Gantt chart? Available at: https://www.teamgantt.com/what-is-a-gantt-chart
154. The Retail Exec (2025) Sortly inventory management for retail. Available at: https://theretailexec.com/tools/sortly-review/
155. The Workflow Academy (2024) Zoho Inventory automation features. Available at: https://www.theworkflowacademy.com/zoho-inventory/
156. ToolsGroup (2024) The hidden costs of poor inventory management: How much are you really losing? Available at: https://www.toolsgroup.com/blog/the-hidden-costs-of-poor-inventory-management-how-much-are-you-really-losing/
157. Tufte, E.R. (2020) The visual display of quantitative information. 2nd edn. Cheshire, CT: Graphics Press.
158. Unicommerce (2024) Malaysian retail inventory management challenges and solutions. Available at: https://www.unicommerce.com

159. W3C (2021) CSS media queries level 4. W3C Candidate Recommendation. Available at: https://www.w3.org/TR/mediaqueries-4/
160. W3C (2022a) CSS flexible box layout module level 1. W3C Candidate Recommendation. Available at: https://www.w3.org/TR/css-flexbox-1/
161. W3C (2022b) CSS values and units module level 4. W3C Working Draft. Available at: https://www.w3.org/TR/css-values-4/
162. W3C (2023a) CSS grid layout module level 2. W3C Candidate Recommendation. Available at: https://www.w3.org/TR/css-grid-2/
163. W3C (2023b) HTML5: A vocabulary and associated APIs for HTML and XHTML. W3C Recommendation. Available at: https://www.w3.org/TR/html52/
164. W3C (2023c) Notifications API. W3C Working Draft. Available at: https://www.w3.org/TR/notifications/
165. Ware, C. (2021) Information visualization: Perception for design. 4th edn. Cambridge, MA: Morgan Kaufmann.
166. Web Hypertext Application Technology Working Group (2024) Fetch standard. WHATWG Living Standard. Available at: https://fetch.spec.whatwg.org/
167. Wertheimer, M. (2020) 'Laws of organization in perceptual forms', Psychological Research, 84, pp. 1-11. doi: 10.1007/s00426-019-01256-9.
168. Wharton, C., Rieman, J., Lewis, C. and Polson, P. (2020) 'The cognitive walkthrough method: A practitioner's guide', Usability Inspection Methods, pp. 105-140. New York: John Wiley & Sons.
169. World Wide Web Consortium (2023) Web Content Accessibility Guidelines (WCAG) 2.1. W3C Recommendation. Available at: https://www.w3.org/TR/WCAG21/
170. Wroblewski, L. (2020) Web form design: Filling in the blanks. 2nd edn. Brooklyn, NY: Rosenfeld Media.
171. Wright, G. (2021) Windows 11 – What's new and what's changed. Available at: https://news.microsoft.com/june-24-2021/
172. Yandex (2020) BEM methodology. Available at: https://en.bem.info/methodology/
173. Zeller, A. (2020) Why programs fail: A guide to systematic debugging. 3rd edn. San Francisco, CA: Morgan Kaufmann.
174. Zenatta Consulting (2025) Zoho Inventory implementation guide. Available at: https://zenatta.com/the-ultimate-guide-to-configuring-optimizing-zoho-inventory/
175. Zoho Corporation (2025) Zoho Inventory - batch tracking and expiry management. Available at: https://www.zoho.com/inventory/

Links

1. Project Source Code (GitHub Link):

- https://github.com/efahuda03/EXPI-STOCK---INVENTORY-SYSTEM.git

2. Demonstration Video (YouTube Link):

- Admin Section - https://youtu.be/Bcn3L40Yzys?si=ezVW8-8CYAHa78Ot
- Staff Section - https://youtu.be/IWAQeVzs6mE?si=qfdmYMTnnvi4N6Wv