UNIVERSITI POLY-TECH MALAYSIA

SMART INVENTORY AND TRACKING SYSTEM FOR SME CONTRACTOR

NURULAIN BINTI ROSDI

BACHELOR OF INFORMATION TECHNOLOGY (HONS) IN BUSINESS COMPUTING

UNIVERSITI POLY-TECH MALAYSIA
Faculty of Computing & Multimedia

SMART INVENTORY AND TRACKING SYSTEM FOR SME CONTRACTOR

NURULAIN BINTI ROSDI
AM2311015218

FYP4094

NOVEMBER 2025

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: 23/6/2025

Office Stamp

Abstract

In this era of globalization, a web and application are significant to all of us. We are using mobile phones every single day through web and application. In conjunction with that I decided to approach Avant Garde Solution(M) Sdn.Bhd inventory system to provide them with an application for their company as they are searching for a web and app developer. The application that is going to be developed is called Smart Inventory and Tracking System for SME Contractor. This application is going to be used by two users which are the field staff and the admin. This application is intended to record inventory Avant Garde Solution(M) Sdn.Bhd more convenient for field staff. As for the amin, it is designed to help them to manage and update all the stocks. Besides, admin can also download the report details as well.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my heartfelt gratitude to the several individuals and company for supporting me throughout my final year project. This experience has been nothing short of transformative, and I am deeply appreciative of the support and guidance I received throughout my time. First and foremost, I want to extend my gratitude to Madam Afra binti Taidin, my supervisor during the final year project. She provided me with enthusiasm, insightful comments, helpful information and unceasing ideas that have always helped me tremendously in my research and writing of this project. She patiently guided me and offering constructive feedback that significantly enhanced my skill. I am genuinely thankful for her mentorship.

I would also like to acknowledge the entire Avant Garde Solution team for their warm welcome and continuous support. Everyone I had the pleasure of working with made me feel like an integral part of the team and encouraged me to contribute my best. I am also grateful to Allah, my family and friends that always support and care for me until I finished my final year project. My heartfelt thank also go to my coordinator, Madam Nora for affording us, the degree students majoring in Business Computing has enabled us to complete this project successfully. Smart Inventory and Tracking System App at Avant Garde Solution(M) Sdn. Bhd. has been a significant milestone in my educational and professional journey. The knowledge, skills, and connections I gained during this time are priceless, and I will carry them with me throughout my career. Once again, thank you for giving me this remarkable opportunity.

1 INTRODUCTION

1.1 Introduction

Inventory management refers to the process of ordering, storing, using, and selling a company's inventory. This includes raw materials, components, and finished products, as well as the warehousing and processing of these items. There are different methods of inventory management, each with its pros and cons, depending on a company's needs (Hayes, 2024). In today's fast-paced and competitive environment, effective inventory management plays a crucial role in maintaining operational efficiency, especially for small and medium-sized enterprises (SMEs) in the construction and contracting sectors. However, many SMEs continue to rely on manual or spreadsheet-based methods, which are prone to errors, inefficiencies, and communication delays (Mourtzis et al., 2016). These limitations can lead to overstocking, stockouts, and project delays—directly affecting profitability and client satisfaction. The solution also enables professional reporting, secure user authentication, and automation features such as alerts for low stock levels.

An inventory management system can help the company to track their daily inventory throughout the days or months more effectively. To address these challenges, the Smart Inventory Tracking System has been developed as a centralized, cloud-based solution tailored to the specific needs of SME contractors. This system integrates a web-based admin dashboard for stock monitoring and reporting, along with a mobile application for team members to record stock-out activities on-site. By leveraging real-time data updates through Firebase and user-friendly interfaces built with React and Flutter, this system aims to enhance visibility, accuracy, and decision-making in inventory management. From this era, thousands of companies in Malaysia are entering the digital world by releasing a mobile app or their own internal inventory system. SME company have come to understand that spending a little money each month on such technology solutions would helped them to be more efficient in inventory management. Through this project, digital tools are applied not only to streamline business processes but also to support data-driven decisions that are vital for SME growth and sustainability (Christopher, 2016).

1.2 Project Background

Small and medium-sized enterprises (SMEs) in the contracting and construction industry often operate in fast-paced, resource-sensitive environments where efficient inventory management is crucial to project success. However, many of these SMEs lack access to affordable, tailored digital systems that support real-time inventory tracking. Instead, they rely heavily on manual documentation, spreadsheets, or informal communication methods, which can lead to inaccurate stock records, material shortages, and project delays (Gunasekaran et al., 2017).

Smart Inventory and Tracking System for SME Contractor is a business website and application that is provided for an online system that allows field staff to update the inventory throughout the day by update the total in and out of the inventory. Besides, the admin can manage inventory by adding new stock and generate report such as excel and file in pdf format. Avant Garde Solution (M) Sdn. Bhd. is a small contractor home-based company that located in Bandar Tun Razak, Cheras Kuala Lumpur. Avant Garde Solution (M) specializes in total solutions for Building Automation System (BAS/BCS/BMS) and ELV System that offer a wide range services including consultancy, design, supply, installation and even after-sales maintenance service contracts. The admin wanted to develop an inventory system by website and application that called Smart Inventory and Tracking System for SME Contractor that can ease their field staff to update their inventory by apps. Then, the field staff does not need to use the traditional method which they need to write inventory management or using Microsoft words or Excel. The field staff can simply use the apps that provide features of inventory management. Admin can update the new inventory and edit the quantity based on their purchases of the stocks.

1.3 Problem Statement

A problem statement is a clear, concise description of an issue or challenge that needs to be addressed. It outlines the gap between the current state and the desired future state and often explains why the problem matters. It's a crucial step in problem-solving, helping to define the scope and focus of a project or research. An excellent problem statement not only explains the relevance and importance of the research but also helps readers quickly determine if the article aligns with their interests by clearly defining the topic (Elsevier, 2022).

1.3.1 Inefficient Stock Management

Small and medium enterprises (SMEs), especially in the construction or contractor sectors, often face challenges in maintaining efficient stock control. Manual methods such as paper logs or spreadsheets are still widely used, resulting in errors like miscounts, stock duplication, or unrecorded

withdrawals. These inefficiencies may lead to production delays, increased operational costs, and inventory shrinkage (Muller, 2011). Furthermore, manual systems lack automation, requiring significant time and effort for routine stock-taking and reconciliation, thus reducing overall productivity.

1.3.2 Lack the Tools to Generate Professional Inventory Reports

Most SMEs do not utilize software tools that can generate accurate and professional inventory reports in real-time. Reporting is often done manually or using basic spreadsheet templates, which are not optimized for analytics or formal presentation. This limitation affects decision-making processes, particularly in identifying trends, forecasting stock needs, or preparing reports for stakeholders. Without automated reporting tools, companies also struggle with consistency in data formatting and compliance reporting (Rai & Sahoo, 2012).

1.3.3 Lack of Real-Time Inventory Visibility

Real-time visibility refers to the ability to monitor stock levels and movement as they occur. Many SMEs lack systems that provide this functionality, leading to delays in identifying low-stock situations or overstocking. Without real-time updates, inventory records may not reflect actual usage by on-site team members, especially in fast-moving environments like construction or logistics. This disconnect can result in project delays, emergency procurement, and unplanned downtime (Waller & Fawcett, 2013).

1.4 Project Objectives

Project objectives are specific, measurable outcomes that a project aims to achieve within a defined timeframe. They clarify the project's purpose and guide the team toward successful completion. Articulating project objectives is crucial in project management, as they significantly influence decision-making and resource planning throughout the entire project life cycle (Atlassian, 2024).

1.4.1 To Develop an Application That Can Monitor Inventory Management

The primary objective is to design and implement a digital application that streamlines inventory monitoring. By shifting from manual methods to a centralized, app-based solution, businesses can achieve more accurate tracking of stock levels, movement, and usage. Such a system reduces the chances of overstocking or stockouts and enhances data accuracy. Mobile and web-based inventory applications enable staff to update records in real time, improving operational efficiency (Mourtzis et al., 2016). Modern inventory systems contribute to faster decision-making and better resource planning, especially in fast-paced environments like SME construction and logistics.

1.4.2 To Generate Professional Inventory Reports

Another objective is to incorporate automated reporting capabilities within the inventory application. These reports should include stock inflow/outflow summaries, low-stock alerts, and usage analytics, formatted in a presentable and exportable format (e.g., PDF or Excel). Professionally generated reports aid in strategic planning, stakeholder communication, and compliance reporting. They help businesses evaluate inventory turnover, waste, and purchasing patterns (Christopher, 2016). Furthermore, report automation reduces human error and administrative workload.

1.4.3 To Provide Real-Time Visibility of Inventory Data to Users and Management

This objective emphasizes implementing real-time data synchronization using technologies like Firebase or cloud databases. Real-time inventory visibility ensures that field staff and management can access up-to-date stock information at any time, reducing miscommunication and downtime. It allows proactive stock reordering and immediate identification of stock irregularities (Waller & Fawcett, 2013). In dynamic environments, such as SMEs with distributed teams or job sites, this visibility is crucial for operational continuity and performance optimization.

1.5 Scope and Target User

Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, tasks, costs and deadlines. The documentation of a project's scope is called a scope statement or terms of reference. It explains the boundaries of the project, establishes responsibilities for each team member and sets up procedures for how completed work will be verified and approved (Lutkevich, 2021). A target user refers to a specific group of people who are most likely to be interested in a particular product or service. These individuals share similar characteristics, behaviours, and demographics, making them a valuable focus for businesses aiming to tailor their marketing efforts effectively (Breakcold.com, 2023).

1.5.1 Project Scope

The scope of this project is the design, development, and deployment of a two-part inventory management solution. The first component is a web-based platform that is built using React and Firebase, which serves as the central control panel for administrators. This interface allows for the monitoring of stock in and out records, editing of inventory data, and the generation of professional reports in PDF or Excel formats. The second component is a Flutter-based mobile application intended for use by team members on-site. This application enables authenticated users to record stock-out transactions in real-time, ensuring that all material usage is promptly logged and synced with the central database.

The system’s development also includes setting up Firebase Firestore for real-time cloud storage, implementing Firebase Authentication for secure role-based access, and ensuring responsiveness and usability across different device types. While this system focuses primarily on inventory tracking and visibility, it does not extend to other enterprise modules such as finance, procurement, or human resources making it lightweight, affordable, and specifically tailored to SME contractor operations.

1.5.2 Product Scope

The Smart Inventory Tracking System offers distinct features for admin and field staff. On the web-based admin platform, users can authenticate securely, dashboard admin can view an overview of inventory movements, manually input new stock data, edit existing records, delete outdated entries, and generate downloadable reports. This allows for better planning, audit readiness, and decision-making.

In parallel, the mobile application empowers field staff to log stock-out on-site events such as when materials are used on a construction project that can directly from their phones. These inputs are sent instantly to the central system, ensuring real-time visibility for the management team. The

app includes form validation, edit functionality, and feedback messages for successful submissions or errors. These features collectively ensure the application not only streamlines inventory tracking but also supports traceability, accountability, and proactive stock replenishment (Christopher, 2016).

1.5.3 Target User

The primary users of the system are field staff that can update stock in or out while in or outside of company by their own phone. Smart Inventory Tracking System for field staff is using the mobile apps that they need to log in before editing the stocks and they cannot change any details of stock. If there any wrong spelling or invalid information, they need to contact admin by feedback messages in the staff dashboard. The new staff needs to get approval email from admin because the inventory are private and confidential. Basically, Smart Inventory Tracking System apps cannot be used by public because they need to request permission from administrators.

Next user is administrator who require accurate and up-to-date records to make restocking and reporting decisions. Smart Inventory Tracking System for admin is using the website system that only one person that can log in and editing the stock also adding a new stock in the system. Everything on his or her behalf of the inventory management such as the details, stock information and deleting any inventory information. The admin can generate report in this system by day, month or year without any delay and they need to be alert about any feedback messages from field staff.

1.6 Overview of This Report

Inventory management refers to the systematic approach of ordering, storing, using, and tracking a company's inventory, including raw materials, components, and finished goods. It also encompasses warehousing and the processing of items to ensure smooth operations. Depending on a company's operational scale and industry, various inventory management techniques can be applied to maintain balance between supply and demand (Hayes, 2024).

In the construction and contracting sectors, especially among small and medium-sized enterprises (SMEs), inventory management plays a critical role in project continuity and cost control. However, many SMEs continue to rely on outdated manual methods—such as paper records or spreadsheets—which are prone to human error, inefficiencies, and delays in communication. These limitations often result in stock discrepancies, overstocking, or material shortages that can delay projects and reduce overall client satisfaction (Mourtzis et al., 2016).

To address these challenges, the Smart Inventory Tracking System for SME Contractors has been developed as a cloud-based digital solution. The system includes two primary components: a web-based dashboard for administrators and a mobile application for field staff. The admin platform, developed using React and Firebase, allows for monitoring stock movements, editing inventory records, and generating formal reports in both PDF and Excel formats. Simultaneously, the Flutter-based mobile app enables team members to log stock-out events directly from construction sites or remote locations.
With real-time data synchronization via Firebase, this system empowers users with up-to-date inventory visibility and decision-making capabilities. Features such as automated alerts for low stock levels, secure authentication, and user-specific access further enhance usability and data integrity. According to Christopher (2016), the ability to act on real-time data can significantly improve supply chain responsiveness and planning accuracy.

This innovation reflects a growing trend in Malaysia, where SMEs are embracing digital transformation by adopting internal systems and mobile apps to improve operations. As many businesses realize the cost-efficiency and long-term value of digital tools, this system aims to provide a scalable and user-friendly inventory tracking solution tailored to their unique needs. Ultimately, the project contributes to the modernization of SME practices, enhancing transparency, accountability, and operational sustainability.

2 LITERATURE REVIEW

2.1 Introduction

To develop a system, we need some references so that it can give us ideas on how the system will look. The reference of the system is based on the research that has been done in this chapter. References that we get from this chapter can help us to develop a more attractive interface as well as the functionality of Smart Inventory Tracking System. We will discuss the other mobile applications in the market that have the same concept with our Smart Inventory Tracking System concept which are BoxHero, ManageEngine and ZohoInventory.

From a technological standpoint, the integration of React (for web development) and Flutter (for cross-platform mobile apps) with Firebase as the backend is increasingly popular in modern system development. These tools provide rapid development capabilities, scalability, and built-in user authentication. Studies have shown that Flutter enables faster mobile deployment with a single codebase while offering native performance (Google Developers, 2023).

We will investigate several important components in detail that relate with our project. After the research has been done, we will compare each of them and discuss which have the most suitable component for our project. From there, we will make the web and mobile application as our reference to complete our project.

2.2 Investigation

This section explores the key concepts, tools, and technologies used in the development of the Smart Inventory Tracking System for SME Contractors. It includes an investigation into the importance of real-time inventory systems, the use of Firebase as a cloud backend, and the application of role-based access control (RBAC) to ensure secure data handling.

2.2.1 Real-Time Inventory Management System

Real-time inventory management refers to the process of tracking inventory levels, stock movement, and usage as it happens, without delays (Waller & Fawcett, 2013). This system allows businesses to make immediate decisions based on current data, helping to reduce stockouts, overstocking, and manual errors. For SMEs in the construction sector, where materials are often

consumed on-site, real-time inventory tracking ensures accurate updates from remote teams. Studies have shown that implementing such systems improves supply chain efficiency and supports lean inventory practices (Mourtzis et al., 2016).

2.2.2 Firebase as a Backend Platform

Firebase is a Backend-as-a-Service (BaaS) platform developed by Google that provides various services such as real-time NoSQL databases, authentication, cloud functions, and hosting (Google Developers, 2023). In the context of this project, Firebase Firestore is used to store and sync inventory data instantly across devices. Firebase Authentication secures access for different user roles (admin or field staff), while Firebase Hosting and Cloud Functions can automate background tasks. Firebase's real-time capabilities and ease of integration with mobile and web platforms make it ideal for scalable and responsive inventory systems for SMEs.

2.2.3 Role-Based Access Control (RBAC) for Security

Role-Based Access Control (RBAC) is a method of regulating user access based on their role within an organization. It ensures that users can only access information and perform actions that are permitted for their role (Ferraiolo et al., 2001). In the Smart Inventory Tracking System, RBAC is implemented to separate access between admin users (who manage the inventory and generate reports) and team members (who only record stock-outs). This approach minimizes data tampering, prevents unauthorized changes, and supports accountability in inventory management.

2.3 Related Works

For this project, I have been doing my research on three web applications that are related to my project. Those three applications are BoxHero, ManageEngine and ZohoInventory that offer a range of inventory tracking, stock control, and reporting tools. Understanding their features, strengths, and limitations provides a valuable benchmark for developing an improved, tailored solution that meets the specific needs of SMEs in the contracting industry.

2.3.1 BoxHero

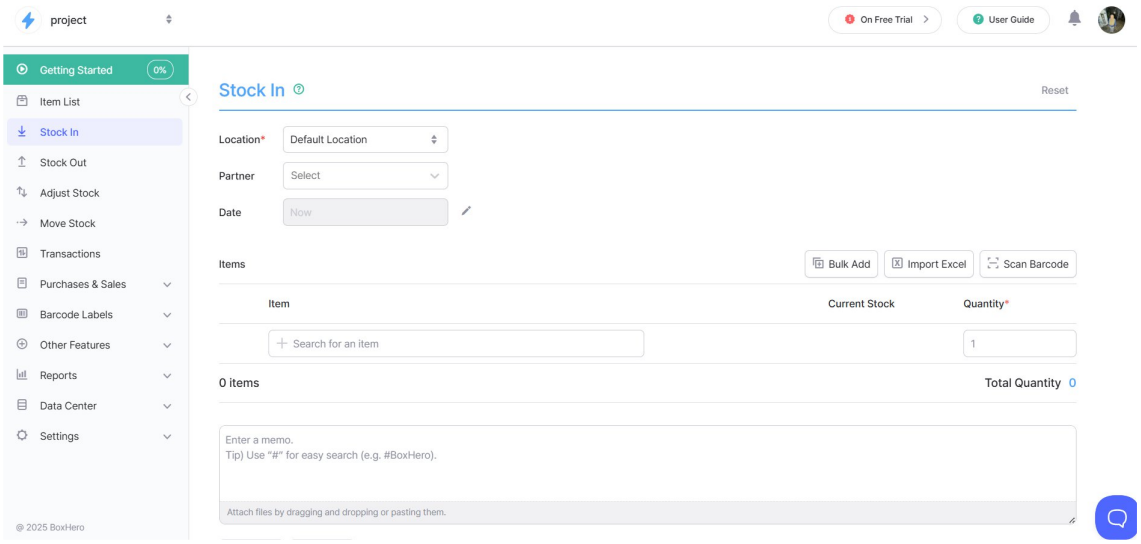


Figure 2.1: BoxHero

Based on Figure 2.1, BoxHero is a cloud-based inventory management software that designed for small to mid-sized businesses. It focuses on simplicity, real -time inventory management tracking and mobile usability. Key features is a real-time inventory sync which is tracks item quantities in real-time across users and devices. As we can see, it divides the categories such as item list, stock in, stock out, adjust stock, transaction and others.

BoxHero also the barcode scanning that supports scanning via smartphone camera or dedicated scanner. The stock in or out logs are automatically records inventory movement for transparency. It also mobile app support which is Android and iOS apps make it suitable for field or warehouse use. The team collaboration allows multiple users with permission controls. BoxHero is ideal for SMEs needing a lightweight and mobile-friendly inventory system and companies with basic stock control needs. Each page has their own functionality and it's free for small teams with limited features. Basically, we need to paid plans with enhanced reporting and user permissions.

2.3.2 ManageEngine

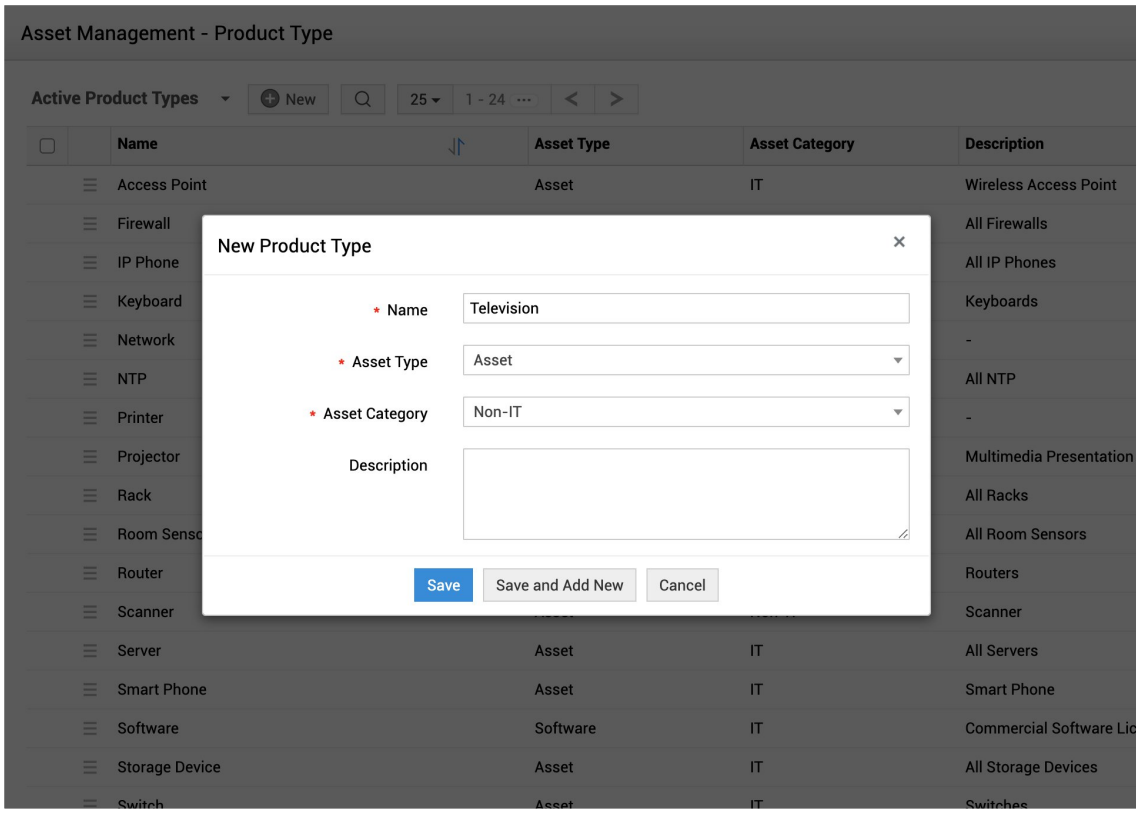


Figure 2.2: ManageEngine

Based on Figure 2.2, ManageEngine is a web-based IT Asset Management (ITAM) tool designed for enterprises. It helps in tracking, managing and auditing IT hardware and software assets throughout their lifecycle. Key features in ManageEngine are asset lifecycle management, CMDB integration, software license compliance, barcode and QR support, custom reports and dashboard.

Asset lifecycle management covers procurement to disposal of IT assets and CMDB Integration includes a configuration management database for relationship mapping. Software licence compliance only for asset tagging and tracking. Custom reports and dashboards are advanced data visualization and export options. The security is role-based access control, integration with active directory, audit and compliance tracking. ManageEngine ideal for enterprise or IT departments that needing comprehensive asset and lifecycle management. The organization requiring software compliance and audits. It's a paid tool, only 14 days free trial available and the pricing depends on number of assets or users.

2.3.3 ZohoInventory

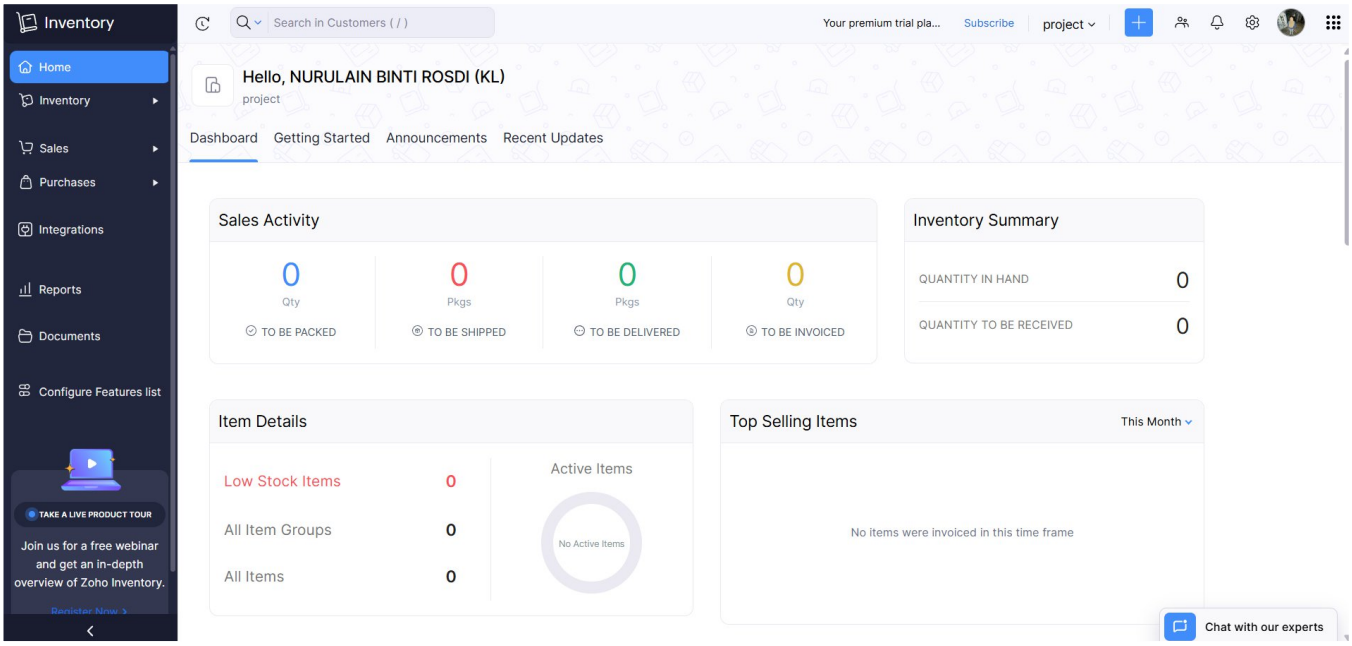


Figure 2.3: ZohoInventory

Based on Figure 2.3, Zoho Inventory is a cloud-based inventory management platform designed for small and medium businesses. It provides multi-channel selling, warehouse tracking, shipment integration, and professional reporting tools. Zoho Inventory also integrates seamlessly with Zoho CRM, Zoho Books, and third-party platforms like Shopify, Amazon, and eBay. Its dashboard is intuitive, offering real-time insights and automation tools for reorder alerts, batch tracking, and sales order management. It supports both web and mobile access, making it suitable for businesses needing flexibility, visibility, and professional documentation capabilities (Zoho, 2024).

2.4 Comparison

Here's a full overview of BoxHero, ManageEngine AssetExplorer, and Zoho Inventory, highlighting their core features, strengths, and ideal use cases.

Table 2.1: Comparison of Existing Project

Criteria	BoxHero	ManageEngine	ZohoInventory
Features	Basic inventory tracking, stock in/out logs, team collaboration	Full IT asset lifecycle, license tracking, audit, CMDB	Sales orders, inventory, shipping, multi-warehouse, integration with e-commerce
Design	Simple, minimal UI; optimized for mobile	Enterprise-style interface with technical terminology	Clean, business-oriented UI with customizable dashboard
Navigation	Very user-friendly; mobile app-focused	More complex; better suited for trained IT staff	Intuitive with step-by-step flows for orders and stock
Usability	Best for small teams with limited tech experience	Requires some IT knowledge; powerful but not beginner-friendly	Easy to use; especially for businesses used to Zoho ecosystem
Availability	Web & Mobile apps (iOS/Android); cloud-based	Web-based only; no mobile app	Web, iOS, Android apps; cloud-based and high availability

2.5 Discussion

Upon reviewing existing inventory management solutions of BoxHero, ManageEngine AssetExplorer, and Zoho Inventory where it is evident that while each platform offers strong features, none are fully optimized for the specific workflows of SME contractors. These applications provided inspiration for the development of the Smart Inventory Tracking System, but the proposed system is carefully tailored to meet the simplicity, mobility, and alert notification required by small and medium-sized enterprises in the construction sector.

BoxHero excels in its minimalist user interface and cloud-based inventory syncing, which supports small teams in maintaining real-time stock records (BoxHero, 2023). This functionality aligns with the needs of field staff who often work off-site and require fast, intuitive ways to log stock-out activities. However, BoxHero lacks deeper customization and administrative controls required by company supervisors. Hence, the proposed system integrates a mobile-first stock-out form for team members and a web-based dashboard for admins to manage inventory records and generate exportable reports.

ManageEngine AssetExplorer, developed by Zoho Corporation, is primarily targeted at IT asset management and excels in audit tracking, license management, and procurement modules (Zoho Corporation, 2022). While feature-rich, its complexity and learning curve make it less ideal for small

contractor companies. Instead of replicating these advanced modules, the proposed system focuses on core inventory functions, such as in/out tracking, low stock alerting, and basic reporting are ensuring accessibility and usability for non-technical users in the construction field.

Zoho Inventory offers an integrated approach to inventory with billing, multi-channel sales, and warehouse management (Zoho Inventory, 2023). However, these features are more suitable for retail or e-commerce businesses rather than SMEs working in decentralized construction environments. Drawing inspiration from Zoho's automated report generation, the proposed system includes Excel and PDF export options filtered by date, providing SMEs with the professional documentation needed for internal reviews and external reporting, without the overhead of CRM or invoicing modules.

In summary, each reviewed system offered valuable insights, but none fully addressed the pain points of small contracting companies are particularly the need for field-friendly data entry, low-complexity admin tools, and real-time cloud synchronization. Therefore, the Smart Inventory Tracking System for SME Contractors is designed as a lightweight, purpose-built solution that combines mobile convenience with professional reporting capabilities, filling a gap in the current market landscape and admin can get alert notifications if low stocks.

2.6 Conclusion

This chapter has explored various literature and related works concerning inventory management systems and how they relate to the development of a Smart Inventory Tracking System for SME Contractors. The investigation highlighted the importance of real-time tracking, automation, and accessibility in modern inventory solutions, particularly for small and medium-sized enterprises operating in fast-paced construction environments.

Through the comparative analysis of BoxHero, ManageEngine AssetExplorer, and Zoho Inventory, it is evident that while these platforms offer comprehensive features, they are often either too generalized or too complex for SMEs in the contractor field. This reinforces the need for a custom-built system tailored to the specific needs of SMEs—focusing on simplicity, mobility, professional reporting, and centralized control.

The insights gained from related works have directly influenced the feature design and implementation strategies for the proposed system. The Smart Inventory Tracking System is thus positioned to bridge the gap between lightweight usability and functional depth—offering real-time inventory visibility, secure login for field staff and admins, and efficient stock in/out tracking.

3 METHODOLOGY

3.1 Introduction

A research methodology describes the techniques and procedures used to identify and analyse information regarding a specific research topic. It is a process by which researchers design their study so that they can achieve their objectives using the selected research instruments. It includes all the important aspects of research, including research design, data collection methods, data analysis methods, and the overall framework within which the research is conducted. While these points can help you understand what research methodology is, you also need to know why it is important to pick the right methodology (Sreekumar, 2023).

3.2 Agile Methodology

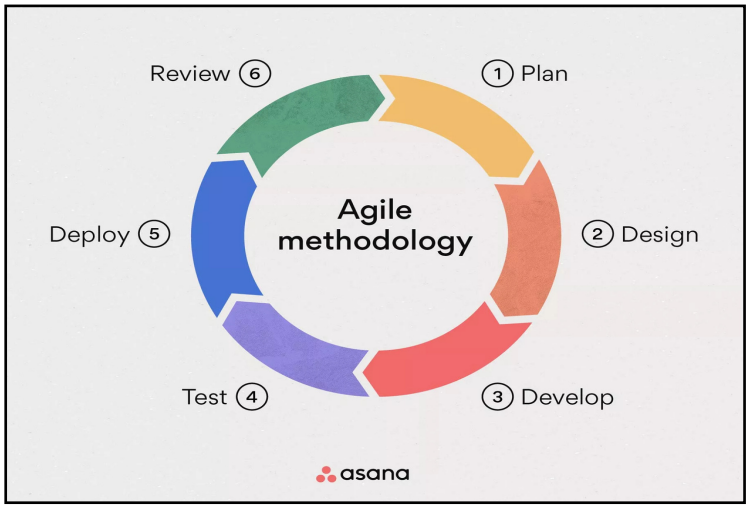


Figure 3.4: Agile Methodology Diagram (Laoyan, 2025)

Agile methodology is a modern approach to software development and project management that emphasizes flexibility, collaboration, and customer-centricity. Originating as a response to the limitations of traditional project management frameworks like the Waterfall model, Agile promotes iterative progress, regular feedback, and continuous improvement. Agile methodology is structured around incremental and iterative development, where projects are broken into small units called sprints or iterations, typically lasting two to four weeks. During each sprint, a cross-functional team

collaborates to design, develop, test, and deliver a working feature or component. This approach enables teams to quickly adapt to changes in requirements or market conditions while ensuring continuous delivery of value to the customer (Highsmith, 2009). Agile has proven to be especially effective in dynamic and fast-paced environments, such as software development, due to its adaptability and focus on user feedback. By prioritizing collaboration and continuous delivery, Agile reduces project risks and helps teams meet customer expectations more efficiently.

3.3 Phases in Agile Methodology

This section outlines each phase of the Agile methodology used in the development of the Smart Inventory Tracking System for SME Contractors. The Agile process is divided into several critical phases which is planning, design, development, testing, deployment, and review—each iteratively contributing toward the final product. These stages are essential in achieving the system's goal of enhancing inventory visibility, accuracy, and reporting for SME contractor operations (Beck et al., 2001).

3.3.1 Plan

The planning phase marks the beginning of the Agile process, where project goals, user requirements, and system functionalities are defined. For this project, planning involved conducting informal interviews with the admin of Avant Garde Solution (M) Sdn. Bhd. to understand challenges in existing inventory processes. Key findings revealed that manual updates using Excel and WhatsApp messages led to inefficiencies, stock errors, and lack of visibility across team members.

Based on this, user feedback survey was distributed to user so that we can get much information from that. The project backlog was structured using Google Form, prioritizing core features including login authentication, stock-in/out tracking, and report generation for admin only. This phase ensured all project goals aligned with user expectations and real operational needs (Highsmith, 2002).

3.3.2 Design

In the design phase, system workflows and UI/UX interfaces were conceptualized. Diagrams such as use case diagrams and flowcharts were developed to visualize the roles of admin and field staff, database interactions, and system flows.

Wireframes for the web platform (admin side) and mobile interface (team side) were created. The admin panel was designed using React for responsiveness and modularity, while the mobile app was planned using Flutter to ensure compatibility across Android and iOS devices. Firebase Firestore was chosen for its real-time capabilities and ease of integration with both platforms (Mourtzis et al., 2016).

3.3.3 Development

The development phase of the Smart Inventory Tracking System marks the transition from design blueprints to a functioning application. In alignment with Agile principles, development was carried out in iterative sprints, with each sprint focusing on implementing core features followed by refinement based on stakeholder feedback (Highsmith, 2002). The system was developed as a two-part solution which is a web-based platform for administrators and a mobile application for team members.

For the web-based admin panel, React.js was chosen due to its component-based architecture, reusability, and efficiency in rendering dynamic interfaces. Firebase Firestore was integrated as the backend cloud database to enable real-time updates, while Firebase Authentication was used to secure access to the admin panel. Admin functionalities include logging in, viewing inventory movements, adding new stock entries, editing stock details, deleting outdated items, and generating inventory reports in PDF and Excel formats.

On the other hand, the mobile application was developed using Flutter, which enables cross-platform deployment on Android and iOS from a single codebase. The app is designed for field staff, allowing them to log in, view their own stock-out history, and submit new stock-out entries with form validation and success/failure feedback. Firebase services were also used on the mobile side to ensure real-time data synchronization and seamless user experience.

During this phase, special attention was given to modular programming and code reusability. Each feature such as login, stock submission, and report generation was developed as an independent module to ensure that future modifications could be implemented without disrupting other parts of the system. The close collaboration between stakeholders and developers throughout each sprint helped ensure that the evolving product remained aligned with user needs (Beck et al., 2001).

3.3.4 Testing

Testing is a critical component of the Agile methodology and ensures that the developed system meets functional, security, and performance standards before deployment. The Smart Inventory Tracking System underwent comprehensive testing across several levels which is unit testing, integration testing, system testing, and user acceptance testing (UAT), in accordance with software engineering best practices (Cohn, 2010).

Unit testing was conducted on individual functions such as login input validation, form submission, and report generation logic. These tests ensured that isolated parts of the system behaved correctly, particularly under edge cases like empty fields or invalid data types. This level of testing was especially important for ensuring input accuracy and avoiding data corruption in Firebase.

Integration testing followed, focusing on the interaction between front-end components and Firebase services. For instance, it was verified that a stock-out submission from the mobile app appeared in the web-based admin dashboard in real-time. Similarly, integration between Firebase Authentication and the login module was tested to confirm that only authenticated users could access respective dashboards.

System testing was then carried out to evaluate the performance of the full system. Functional testing validated that features such as login, stock submission, report generation, and feedback messages worked end-to-end. Non-functional testing evaluated responsiveness across different devices, performance under different network conditions, and security of user credentials during authentication.

Finally, User Acceptance Testing (UAT) was conducted with actual users from Avant Garde Solution (M) Sdn. Bhd., including the company's admin and selected field staff. The users were asked to perform common tasks such as submitting stock entries, reviewing inventory data, and generating reports. Their feedback was collected through observation and informal interviews and was used to make final adjustments before full deployment. UAT helped ensure that the final product not only functioned as intended but also met user expectations in terms of usability and reliability (Schwaber & Sutherland, 2020).

3.3.5 Deployment

The deploy phase involved publishing the working system for real usage. The web-based platform was deployed via Firebase Hosting, and the Flutter mobile application was packaged into an APK file for Android testing. Test installations were conducted on both emulator and physical Android devices. This deployment phase aligned with Agile's principle of delivering working software frequently. Continuous integration ensured that even small changes could be pushed live without disrupting other modules. Early deployment allowed real users to interact with the system and provide insights for the next sprint cycle (Dwyer, 2023).

3.3.6 Review

Review sessions were held at the end of each sprint to evaluate completed tasks, discuss blockers, and refine upcoming features. Sprint retrospectives allowed the developer and stakeholders to assess what worked well and identify areas for improvement. These reviews ensured alignment with the project scope and allowed real-time adjustments based on client feedback. Notable improvements based on review sessions included simplifying form inputs in the mobile app, adding a toast message upon successful submission, and enabling export filters in report generation. These continuous evaluations kept the development agile, user-focused, and adaptable (Schwaber & Sutherland, 2020).

3.4 Conclusion

To summarize, the implementations of Agile concepts for the development of Smart Inventory and Tracking System has proven its capabilities to effectively manage work process involved. The nature of its work iteration fosters continuous improvements to be made, adjusting to the specifications of the system, its technical intricacies and the changing demands of users. Moreover, Agile Methodology promotes self-organization and teamwork – boosting productivity, creative problem-solving approach and critical thinking which leads to a more efficient and effective project outcomes. The specific phases of planning, design, development, and testing, as outlined in this report, align well with the Agile principles of flexibility, collaboration, and customer focus. By breaking down the project into smaller and manageable iterations, risks can be assessed effectively, quality can be enhanced and finally ensure timely delivery throughout the whole development process of Smart Inventory and Tracking System. The successful implementation of Agile will result in a robust and user-friendly Smart Inventory and Tracking System that meets the needs of its users and contributes to the overall efficiency of the organization.

4 REQUIREMENTS

4.1 Introduction

Understanding user needs is essential in building an effective and reliable inventory tracking system. Requirements gathering forms the foundation of system design and development, helping ensure that the final product aligns with business goals and user expectations. For the Smart Inventory Tracking System designed for SME contractors like Avant Garde Solution (M) Sdn. Bhd., a structured approach to requirements elicitation was conducted. This involved direct communication with the intended users—both the admin and field staff—who would interact with the web and mobile applications. The main purpose of this phase was to document the functional and non-functional requirements, as well as to identify the system environment needed to support these features effectively. According to GeeksforGeeks (2023), the process of collecting requirements defines the key features, constraints, and goals of the software, enabling successful project planning and execution. For this project, requirements were identified to support inventory management in small and medium-sized contracting companies, particularly focused on a system that enables centralized stock monitoring and real-time field updates via mobile app. The requirements were collected through interviews and questionnaires with both the admin and team members of Avant Garde Solution (M) Sdn. Bhd., a local contractor company in Cheras, Kuala Lumpur.

4.2 Data Gathering Techniques

To gather accurate and user-centred requirements, a mixed approach was used involving direct interviews and digital questionnaires.

4.2.1 Interview

An in-depth interview was conducted with the admin of Avant Garde Solution (M) to better understand current inventory tracking practices and challenges. Open-ended questions allowed for a deeper exploration of pain points such as data inconsistency, report generation difficulties, and the lack of real-time tracking. The admin highlighted the need for an easy-to-use system capable of generating professional reports and synchronizing stock data across devices, forming the core requirements for the system.

4.2.2 Questionnaire

In addition to interviews, a questionnaire was distributed to field staff through Google Forms. The survey focused on the usability of current tracking methods, preferences for mobile application features, and their technology familiarity. Questions were grouped into demographics, current practices, and feature preferences. Their responses revealed a strong demand for a mobile app that supports quick data entry, editable stock records, and synchronization with the web-based system.

4.3 Functional Requirement

Functional requirements describe the specific behaviours and functions the system must support to fulfil user needs (Altexsoft, 2023). For this project, the requirements are divided into two user roles: Admin (web interface) and Team Members (mobile app).

Table 4.1: Functional requirements table for Admin

Function	Description
Login	Admin must be able to securely log in.
Add/Edit/Delete Inventory	Admin can create, update, or remove inventory items.
View Inventory Data	Admin can see total stock in/out summaries.
Generate Reports	Admin can export inventory data as Excel or PDF.
Filter by Date/Month	Admin can view data by specific timeframes.
Manage Feedback	Admin can review messages submitted by field staff.

Table 4.2: Functional requirements table for Field Staff

Function	Description
Login	Field staff log in using Firebase Authentication.
Submit Stock-Out	Staff can submit a record when materials are used on-site.
Edit Submission	Staff can edit their recent stock-out entries.
View History	Staff can see their recent submissions.
Feedback	Staff can send feedback if data needs admin revision.

4.4 Non-Function Requirement

Non-functional requirements define how the system operates rather than what it does. These include performance, usability, reliability, and security standards (Rome, 2020).

Table 4.3: Non-functional Requirements Table

Requirement	Description
Performance	The system must process stock transactions within 1 second.
Usability	The interface should be intuitive, requiring minimal training.
Reliability	The system must maintain 99% uptime with real-time synchronization.
Security	User authentication and role-based access control must be implemented.
Portability	The mobile app should run on Android and iOS platforms.
Privacy	Sensitive data must be encrypted and inaccessible without admin permission.

4.5 System Requirement

To successfully develop and run the Smart Inventory Tracking System, a set of essential software tools, frameworks, and libraries must be identified and used appropriately. These tools are chosen based on their compatibility with modern web and mobile application development, as well as their support for cloud integration, scalability, and user authentication. The system comprises two main components a web-based admin dashboard and a mobile app for field staff by each requiring specific technologies to function efficiently.

4.5.1 Software Requirements

4.5.1.1 Visual Studio Code (VS Code)

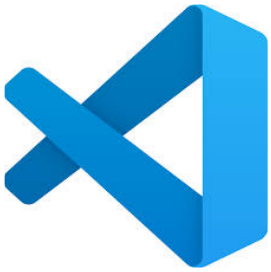


Figure 4.5: Visual Studio Code (Google.com, 2025)

Visual Studio Code is a widely used source-code editor developed by Microsoft. It is a lightweight yet powerful IDE that supports syntax highlighting, code debugging, Git integration, and extensions for multiple programming languages and frameworks. It is ideal for building the React-based web dashboard because of its seamless integration with JavaScript/JSX, Node.js, and Firebase plugins. It also supports live previewing, making the development process more efficient.

Figure 4.6: Android Studio (Wikimedia.org, 2023)

Android Studio is the official integrated development environment for Android app development. It is required to test and build the Flutter mobile app. Even though Flutter supports iOS, Android Studio is essential for Android emulator support, running tests, managing SDKs, and handling Gradle-based builds. It also integrates with Firebase, allowing real-time testing of features like authentication and database reads/writes.

4.5.1.2 React.js

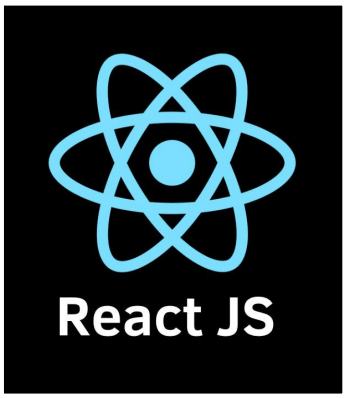


Figure 4.7: React.js (Kasuni Madhushika, 2021)

React.js is a JavaScript library used for building interactive user interfaces, particularly for single-page applications. It was selected for the web admin interface because of its efficiency in rendering dynamic data like inventory records, totals, and charts. React's component-based structure allows the system to be modular, easy to maintain, and responsive.

4.5.1.3 Flutter SDK

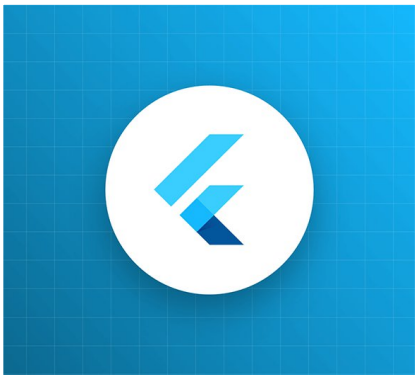


Figure 4.8: Flutter SDK (Auth0, 2022)

Flutter is Google's open-source UI toolkit that allows developers to build natively compiled applications for mobile, web, and desktop from a single codebase. For this project, Flutter is used to build the field staff mobile app that enables users to log stock-out transactions on-site. Flutter's prebuilt widgets and cross-platform capabilities make it ideal for SMEs needing affordable, scalable solutions.

4.5.1.4 Firebase (Firestore & Auth)

Figure 4.9: Firebase (Firebase, 2025)

Firebase is a backend-as-a-service (BaaS) platform provided by Google. It plays a critical role in this project by offering two core services through Firestore that a NoSQL cloud database used to store inventory records with real-time sync between web and mobile apps. Next, Firebase Authentication that manages user logins and ensures role-based access control, allowing only authorized admin and staff to access their respective features. This is important for security and data privacy.

4.5.2 Hardware Requirements

Device	HP Laptop 15
Windows Edition	Windows 11 Home Single Language
Processor	AMD Ryzen 5 5500U with Radeon Graphics 2.1 GHz
Memory (RAM)	8 GB
System Type	64-bit operating system, x64-based processor

Table 4.4: Hardware Specifications Table

4.6 Conclusion

This chapter has outlined the process of gathering requirements for the Smart Inventory Tracking System tailored for SME contractors. Through a combination of interviews and questionnaires, both functional and non-functional requirements were identified to ensure the system meets the operational needs of both admin and field staff. These findings help define the scope and capabilities of the proposed system, providing a solid foundation for its successful development. The gathered requirements ensure that the system aligns with user expectations and supports key inventory tracking functionalities through both web and mobile platforms.

5 ANALYSIS

5.1 Introduction

Analysis plays a vital role in the early stages of software development, where raw data and user feedback are interpreted to derive meaningful requirements and system models. According to Jain (2024), analysis refers to the systematic process of decomposing complex data into manageable components to understand patterns, trends, and needs. In software development, this analysis ensures that the end-product is tailored to the user's requirements and business goals. This chapter elaborates on the findings gathered from stakeholders, including the results of interviews and questionnaires conducted with field staff and the administrator of Avant Garde Solution (M) Sdn. Bhd. Additionally, this chapter includes system models such as the use case diagram, flowchart, and business process model to visually represent user interaction and system workflows for the Smart Inventory Tracking System.

5.2 Data Gathering Analysis

To gather the functional and non-functional requirements of the system, two data gathering techniques were employed by an interview with the client (admin) and a questionnaire distributed among the field staff. These techniques provided insights into how inventory is currently managed and what features are needed for improvement.

5.2.1 Questionnaire Analysis

In the process of gathering requirements for the Smart Inventory Tracking System, a questionnaire was distributed to 3 field staff using Google Forms. The survey was structured into three main sections by demographics, inventory management preferences, and system feature feedback. The demographics section aimed to assess the respondents' familiarity with digital tools, revealing that most staff were moderately comfortable using mobile applications. The second section focused on their inventory management habits, where many staff reported updating stock records inconsistently due to lack of proper tools and time constraints. The final section collected feedback on desired system features, with most respondents emphasizing the importance of mobile access, real-time updates, and notification alerts to streamline their workflow and ensure accurate, timely inventory data synchronization.

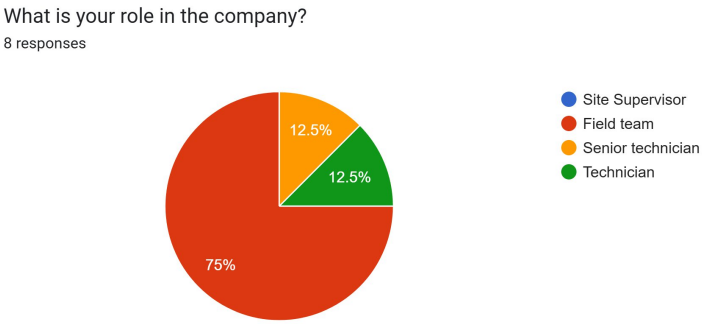


Figure 5.10 : Questionnaire - Question 1

Most respondents identified themselves as field staff, with some also performing site supervisory roles. This indicates that the system should prioritize usability for field-based users with limited time for data entry.

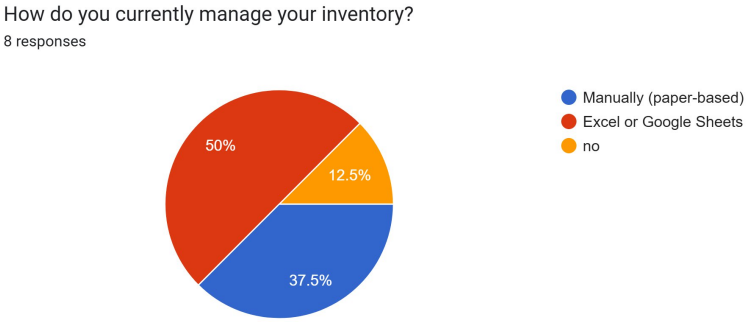


Figure 5.11: Questionnaire – Question 2

Most respondents currently manage inventory manually or using Excel/Google Sheets. This highlights a clear gap for digitization and automation to reduce human error and streamline updates.

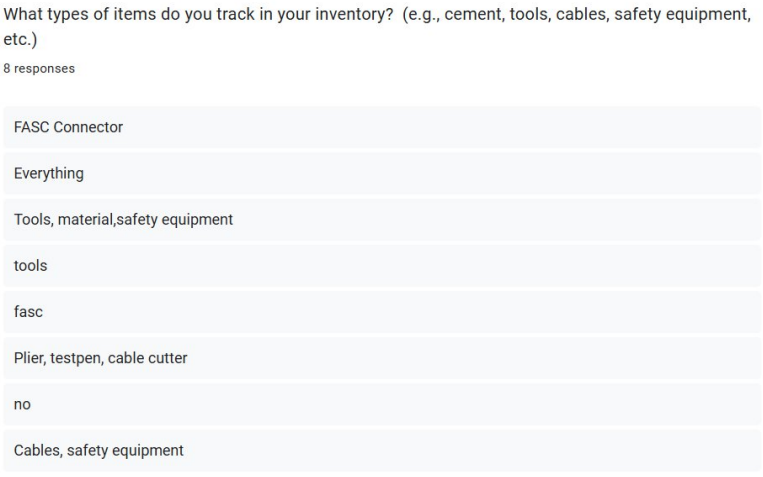


Figure 5.12: Questionnaire – Question 3

Items tracked include tools, cables, safety equipment, and construction materials. This broad scope of inventory emphasizes the need for categorization and search/filter functionality within the system.

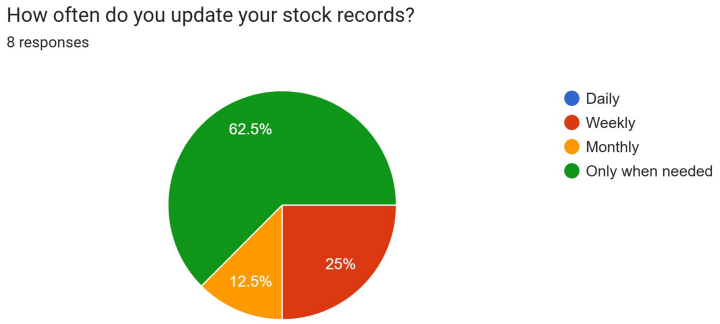


Figure 5.13: Questionnaire – Question 4

Respondents showed varied stock update habits where some do it daily, others only when necessary. This inconsistency underlines the need for reminders and real-time syncing features.

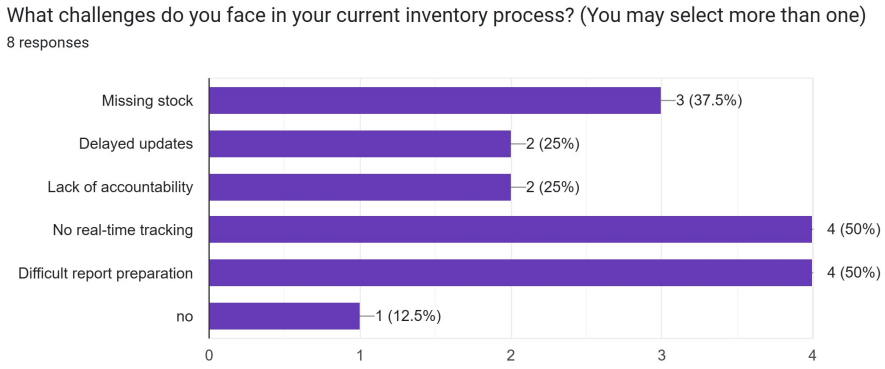


Figure 5.14: Questionnaire – Question 5

Key challenges include missing stock, lack of real-time tracking, and difficulty preparing reports. These responses validate the necessity for features like real-time monitoring, accountability logs, and auto-generated reports.

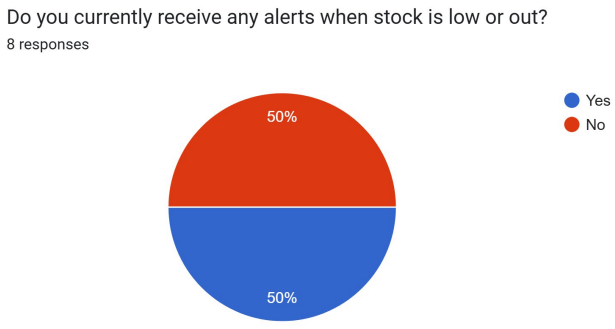


Figure 5.15: Questionnaire – Question 6

Most users reported not receiving any alerts when stock is low or out. Implementing automated alert notifications would greatly enhance stock visibility and planning.

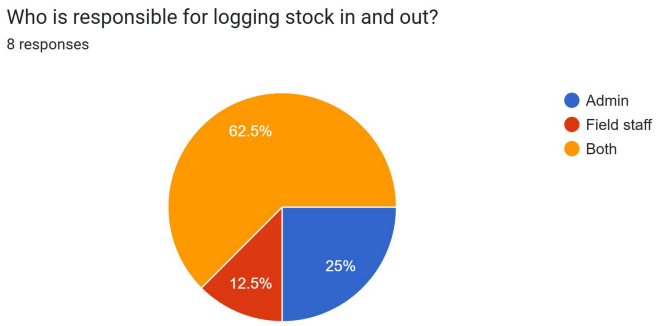


Figure 5.16: Questionnaire – Question 7

Logging responsibility is shared between admin and field staff. The system must allow for dual-role access control and audit trails to ensure accountability.

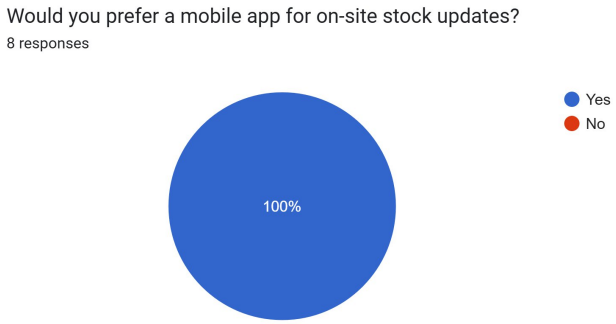


Figure 5.17: Questionnaire – Question 8

All respondents showed interest in a mobile app for on-site updates. This supports the inclusion of a mobile version to complement the web system for ease of access in the field.

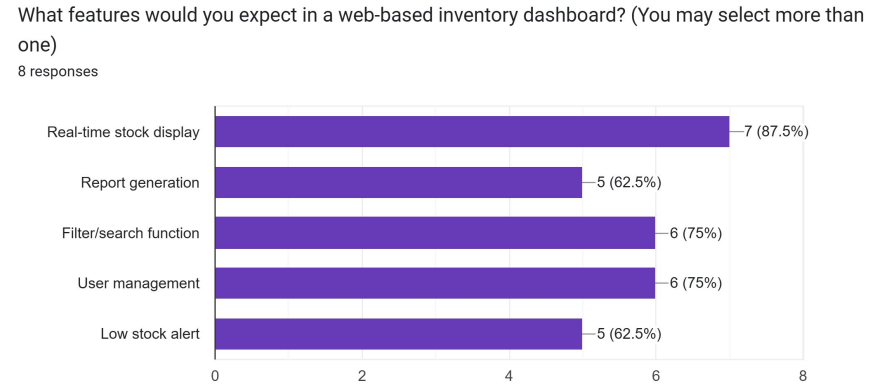


Figure 5.18: Questionnaire – Question 9

Respondents ranked real-time stock display, filter/search function, and user management as top priorities. These should be central components of the dashboard.

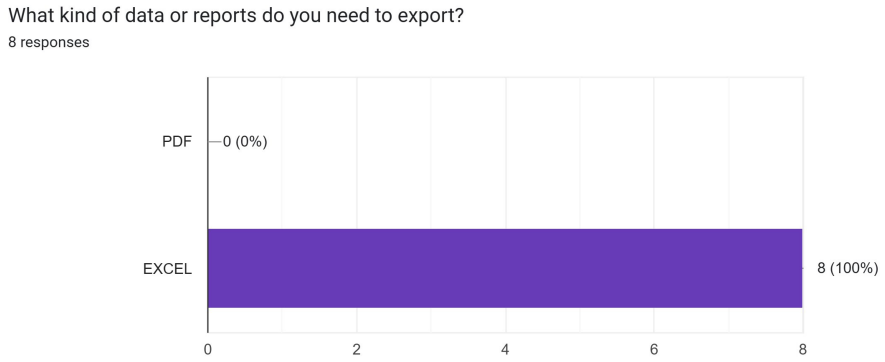


Figure 5.19: Questionnaire – Question 10

Users want to export reports in Excel formats, indicating the system should support multiple file formats for reporting flexibility.

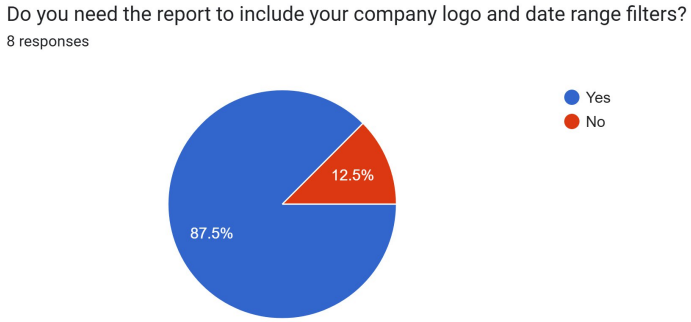


Figure 5.20: Questionnaire – Question 11

87.5% of users want branding and date filtering in their reports. This confirms the need to integrate custom header elements and advanced filtering tools in the report generation module.

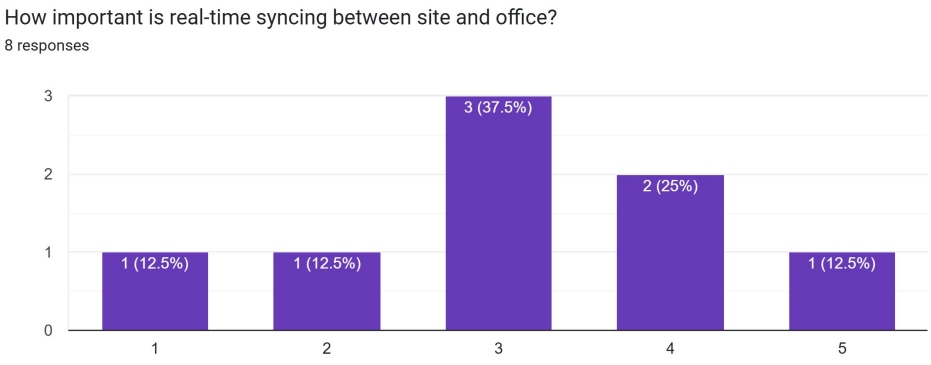


Figure 5.21: Questionnaire – Question 12

6 respondents rated real-time syncing between field and office as very important, reinforcing the need for robust Firebase integration for live data synchronization.

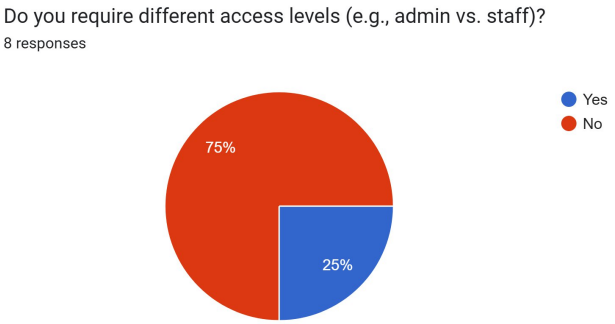


Figure 5.22: Questionnaire – Question 13

Users not agreed on the need for different access levels (admin vs. field staff), because they are not understanding the importance of implementing secure role-based access control (RBAC).

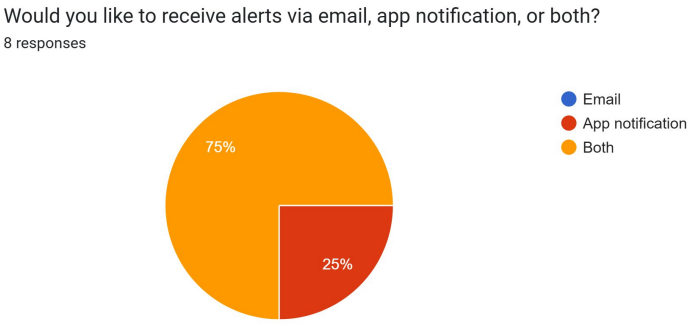


Figure 5.23: Questionnaire – Question 14

Most participants preferred receiving alerts via both email and app notifications. The system should integrate push notification APIs and email services like Firebase Cloud Messaging and SendGrid.

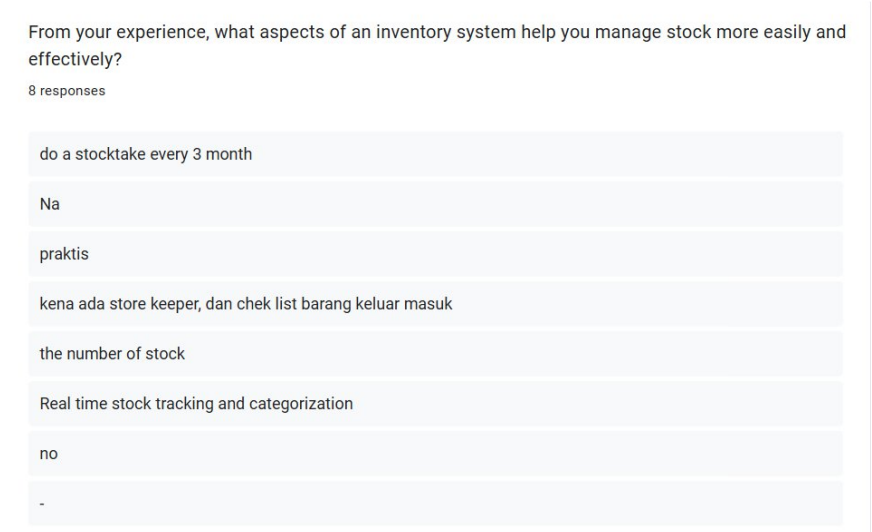


Figure 5.24: Questionnaire – Question 15

Responses included suggestions like real-time tracking, easier stock categorization, and having a storekeeper for validation. These insights offer valuable recommendations for improving inventory visibility and control.

5.2.2 Interview Analysis

Figure 5.25: Interviewee Picture

Name: Puan Nuraza Binti Jamaluddin

Position: HR Administrator

Location: Avant Grade Solution (M) Sdn. Bhd, No 1 Jalan Budiman 2 Bandar Tun Razak 56000 Cheras Kuala Lumpur

Date: 24 June 2025

Purpose: The purpose of the interview was to identify pain points in the current manual inventory tracking process and understand the specific needs of the admin. Key takeaways included the need for real-time stock tracking, an easier way to update stock out records from the field, and the ability to generate professional reports for monthly audits and stock reviews.

Table 5.1: Interview Question – 1

Question 1	How long has the business been?
Answer	The business has started at 2021, so it's been 4 years.
Analysis	According to the business been 4 years, indicates it has already undergone significant development and possibly faces scalability issues in stock management. As the company continues to grow, the need for a structured and digital inventory tracking system becomes essential to handle more complex operations and expanding inventory records effectively.

Table 5.2: Interview Question – 2

Question 2	How many users will use this system (e.g., admin, staff, field workers)?
Answer	The users will be 1 admin and 8 field staff.
Analysis	This highlights the system's role in supporting multiple user types with different responsibilities.

Table 5.3: Interview Question – 3

Question 3	Can you describe your current method of managing inventory movement?
Answer	Our current method just for Control Items only.
Analysis	The current inventory system is limited in scope, indicating a need for expansion to include full tracking of in/out movements and digital documentation.

Table 5.4: Interview Question – 4

Question 4	Do you face any difficulties, during the process of handling your stocks?
Answer	Yes, we cannot get the accurate total stocks.
Analysis	This highlights the inefficiency and lack of real-time data in the current system. Accuracy is critical for operations, and the new system must solve this challenge.

Table 5.5: Interview Question – 5

Question 5	Do you produce any report in on your products stocks?
Answer	Yes, we only using Microsoft Excel.
Analysis	Manual report generation is inefficient and error prone. An automated report generation feature will significantly improve workflow and accuracy.

Table 5.6: Interview Question – 6

Question 6	Do you need to track inventory in and out, or only stock out?
Answer	Yes, we need to track the inventory.
Analysis	This confirms the system must support full stock lifecycle tracking like both inbound and outbound inventory events.

Table 5.7: Interview Question – 7

Question 7	Do you have intention to use web and mobile apps for the business?
Answer	Yes, we had the intention but still in hold.
Analysis	The business is ready to transition to digital tools, indicating the proposed system aligns with their planned technology adoption strategy.

Table 5.8: Interview Question – 8

Question 8	Do you agree with the new proposed system which is an inventory system and mobile apps?
Answer	Of course, we agree the proposed system.
Analysis	Strong endorsement from client validates the project's relevance and suggests good adoption potential.

Table 5.9: Interview Question – 9

Question 9	Do you aware or understand about inventory system and mobile apps?

Answer	Yes, we understand about inventory system and mobile apps.
Analysis	Admin's familiarity with such systems implies reduced training time and better user adoption.

Table 5.10: Interview Question – 10

Question 10	What key features should the admin dashboard include? (e.g., total stock report, user management)
Answer	It should include total stock in or out, can edit and delete items.
Analysis	The admin requires full control over stock management with the ability to modify and audit records, highlighting the need for an intuitive and powerful dashboard.

Table 5.11: Interview Question – 11

Question 11	Do you want the ability to filter inventory by date, month, or project?
Answer	Yes, we want the filter.
Analysis	Filtering is essential for performance audits and report customization, indicating that advanced filtering options must be included in the report generation module.

Table 5.12: Interview Question – 12

Question 12	What features do your team members in the field need? (e.g., stock-out request)
Answer	They can add stock in or out through the system.
Analysis	Field staff need basic functionality like stock in/out recording, reinforcing the need for a simple and responsive mobile UI.

Table 5.13: Interview Question – 13

Question 13	Should team members be able to edit or delete stock entries?
Answer	They shouldn’t to avoid misunderstanding.
Analysis	Access control is a key security requirement. Only admins should have edit/delete permissions to maintain accountability and prevent errors.

Table 5.14: Interview Question – 14

Question 14	Should the admin be able to generate and download PDF reports?
Answer	Yes, admin should be generating report.
Analysis	Reporting is a core requirement. The system must support exporting inventory data in PDF and Excel formats for transparency and documentation.

Table 5.15: Interview Question – 15

Question 15	Would you be open to feedback sessions and testing during development?
Answer	Yes, I would like to.

Analysis	Active collaboration with the client throughout the project will help refine features and ensure the system aligns closely with real-world needs.

5.3 Use Case Diagram

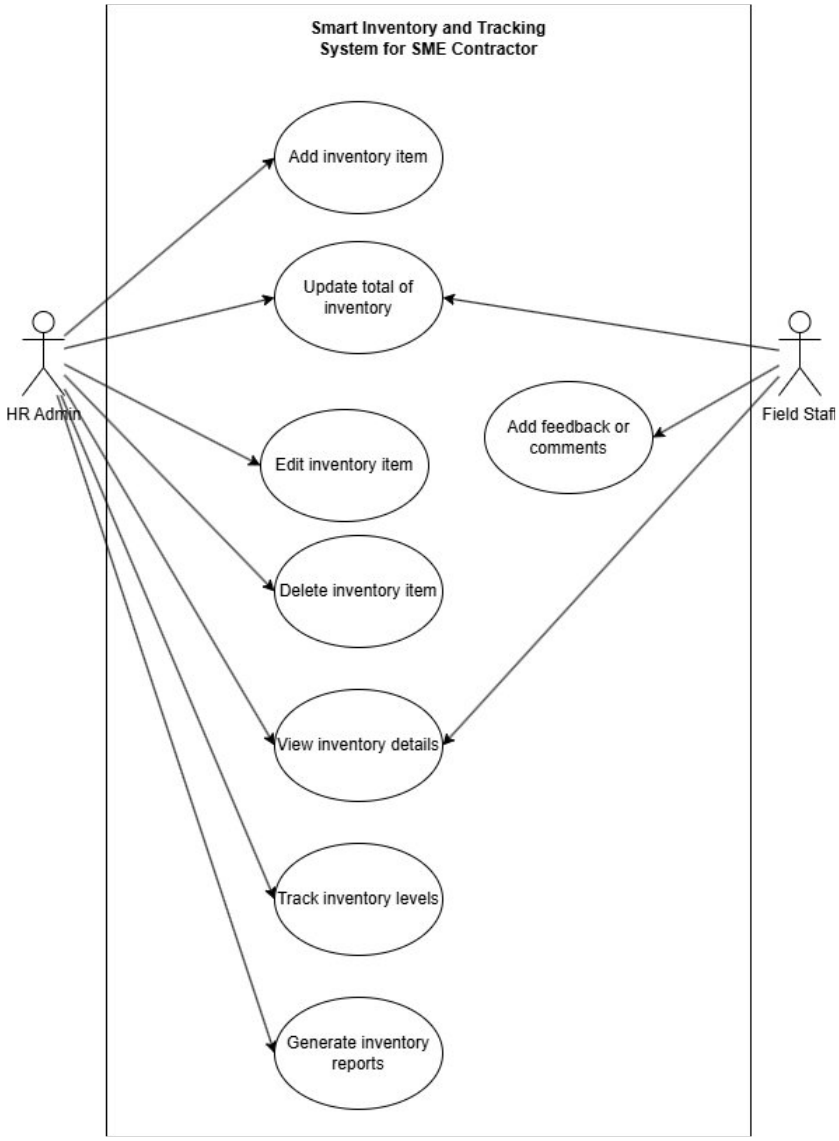


Figure 5.26: Use Case Diagram

The use case diagram for the Smart Inventory and Tracking System for SME Contractor provides an overview of how two primary user roles HR Admin and Field Staff will be interact with the system to manage inventory operations. The HR Admin has full access to core functions such as adding, editing, deleting, and tracking inventory items, as well as generating reports, reflecting their role in overseeing the system. Field Staff can update inventory totals and provide feedback or comments, supporting real-time field-level updates. Both users can view inventory details, ensuring transparency and collaboration. This diagram helps define system requirements and role-based access to maintain efficiency and data accuracy in inventory management.

5.4 Flowchart

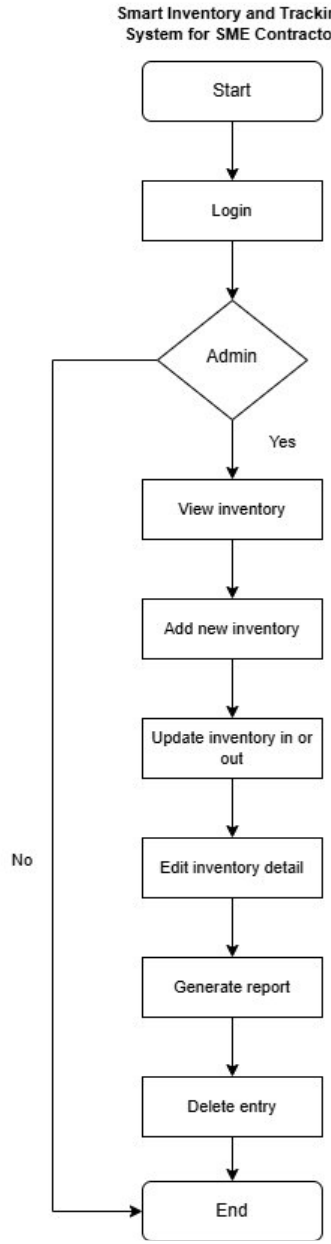


Figure 5.27: Flowchart

The flowchart for the Smart Inventory and Tracking System for SME Contractor outlines the step-by-step process for admin follows to manage inventory activities. It begins with a login process, followed by a decision point that checks if the user is an admin. If verified, the admin can proceed to view the inventory, add new items, update inventory records (in or out), edit inventory details, generate reports, and delete entries as needed. If the user is not an admin, the process loops back without access to these features. This structured flow ensures only authorized personnel can perform critical inventory tasks, promoting secure and efficient inventory management.

5.5 BPMN (Business Process Modelling Notation)

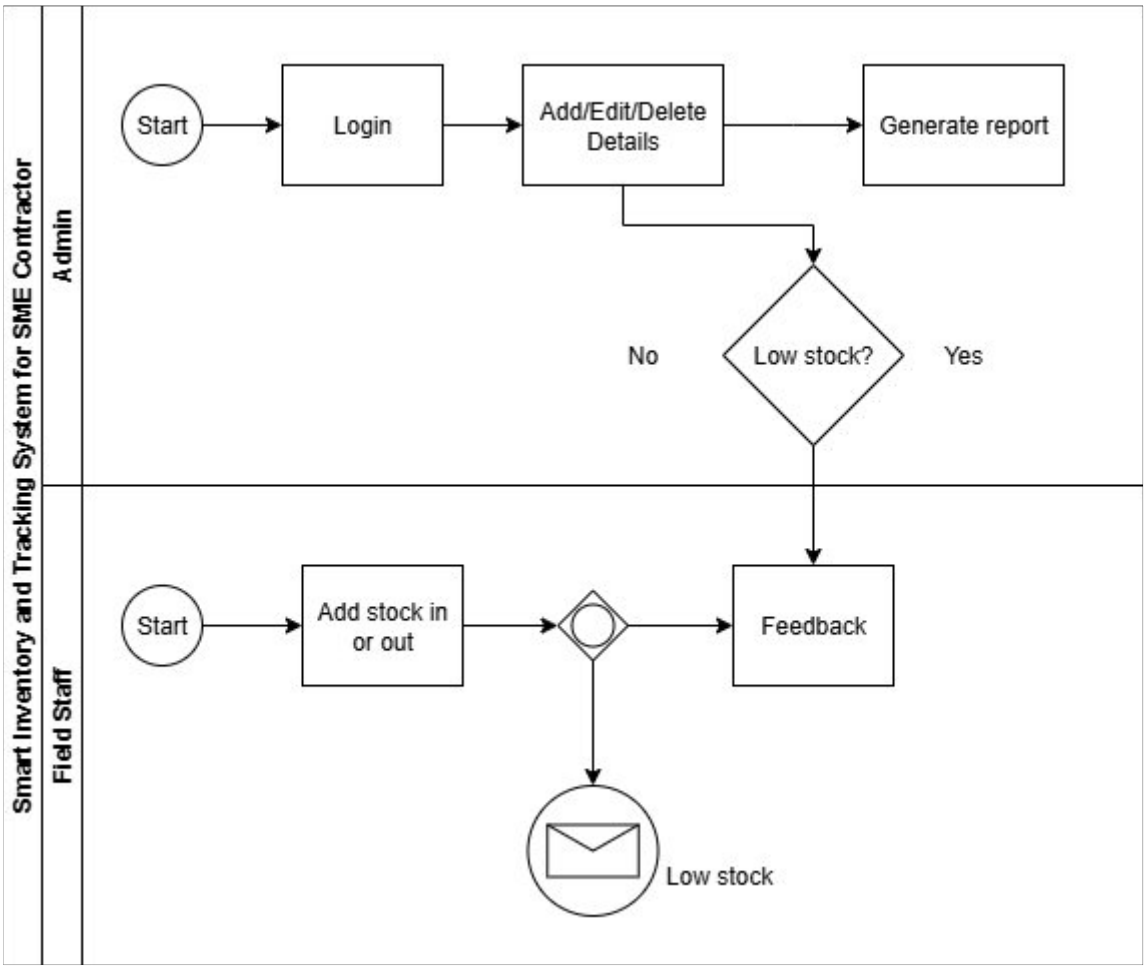


Figure 5.28: Business Process Modelling

The Business Process Modeling Notation (BPMN) diagram illustrates the workflow between two main user roles in the Smart Inventory Tracking System which is the Admin and the Field Staff. The admin accesses the system through a web platform, where they begin by logging in securely. Once authenticated, the admin can manage inventory data by adding new stock entries, editing existing details, or deleting outdated records. They are also notified automatically when stock levels fall below a predefined threshold and can generate professional reports in PDF or Excel formats filtered by date or project. Additionally, the Admin can view and respond to feedback submitted by Field Staff. On the other hand, Field Staff log in through a mobile application designed for real-time inventory input. Their main responsibilities include adding stock in or out records while on-site, receiving alerts for low stock levels, and submitting feedback or comments regarding inventory discrepancies or issues. Both user types interact with a shared Firebase database to ensure data synchronization and real-time communication, allowing the system to maintain transparency, improve inventory accuracy, and streamline collaboration between field operations and administrative control.

5.6 Conclusion

This chapter has provided a detailed analysis of the system's user requirements and workflow using both qualitative and visual tools. The interview and questionnaire helped validate real-world challenges faced by SME contractors in managing inventory. Use case diagrams, flowcharts, and BPMN further clarified the interaction between users and the system. These analytical tools and user feedback will inform the design and development phases, ensuring the solution aligns with actual operational needs.

6 DESIGN

6.1 Introduction

The previous stages' requirements translate into implementation work breakdown for the system design phase. per Sommerville (2016), system design functions as the architects' essential guide to configuring the systems' architecture, interfaces, databases, and processes to meet the system's functional and non-functional requirements. This project focuses on design and construction of a web and mobile based Smart Inventory Tracking System for SME contractors. The system is built to be comprehensive, user-friendly, and scalable, backed interfaces and databases, security frameworks, and system flow designs elaborated as the chapter leads.

6.2 Interface design - wireframe

6.2.1 Admin - Login Page

Figure 6.1: Admin Login

Figure 6.1 display the rough sketch of Smart Inventory and Tracking System's login page. The details needed are the email and password for admin.

6.2.2 Admin - Dashboard Page

Figure 6.2: Admin - Dashboard Page

Figure 6.2 shows that after admin login, it will goes to the dashboard page and admin can see the quantity of item and the list of inventory. Through here, admin can update the quantity of inventory.

6.2.3 Inventory Form Page

Figure 6.3: Inventory Form

Figure 6.3 illustrate the inventory form page which admin can add new inventory here such as Item Name, Quantity and Date. Admin need to click the submit, then the database will captured and show in dashboard page.

6.2.4 Report Page

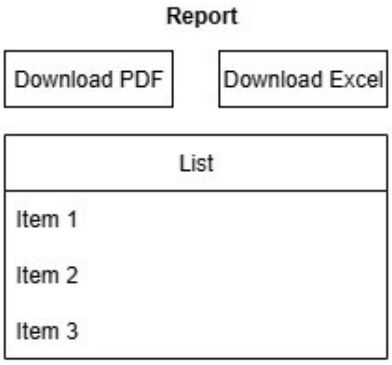


Figure 6.4: Report Page

Figure 6.4 shows the report page that admin can generate professional report from this page. Admin can download the document in PDF or Excel and admin doesn't need to manually put the quantity in the report. It will reduce the workload as admin and he/she doesn't take a long time to prepare the report.

6.2.5 Staff Login

Figure 6.5: Staff Login

Figure 6.5 highlight the login page for Staff only. The details needed are the email and password from admin before they can proceed to login and gain access to the system if the details provided are valid.

6.2.6 Staff Dashboard

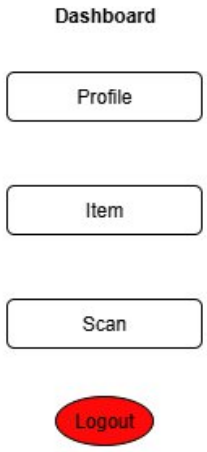


Figure 6.6: Staff Dashboard

Figure 6.6 display the staff dashboard and they can navigate page by the button such as Profile, Item and Scan. The dashboard page for staff more simple than admin because the features is not complicated as an admin. After they finished, they can logout the mobile apps.

6.2.7 Profile Page

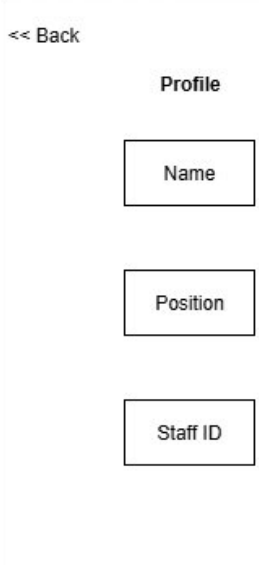


Figure 6.7: Profile page

Figure 6.7 highlight the profile page for Staff only. The details from this page gained from their sign up so they only can view the profile. If they would like to update, the admin will change based on their request.

6.2.8 Scan Page

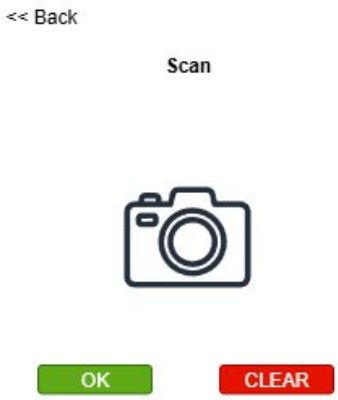


Figure 6.8: Scan Page

Figure 6.8 display the scan page for staff, they need to scan barcode from the item then they can reduce the stock. The barcode is unique and the admin will produce from the admin dashboard. Staff can only reduce the stock, they cannot add new inventory or add stock.

6.2.9 Item Details Page

Figure 6.9: Item Details Page

Figure 6.9 illustrate the item details page that staff can only view the list of inventory. They cannot update the item name or add new inventory because the edit for admin only.

6.3 Database design

6.3.1 Data Dictionary

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.

6.3.1.1.1 Inventory Table

Table 6.1: Data Dictionary – Inventory Table

Data Item	Data Type	Data Format	Description	Example
barcodeUrl	String	NN	Unique identifier for each item	data:image/png;base64
date	Timestamp	Date	Date entry of item	September 23, 2024 at 7:50:44 PM UTC+8
item	String	Text	Name of item	cable
lowStock	Boolean	NN	Indicates whether an item has reached its minimum threshold level.	false
qty	Number	Number	Total quantity of item	4
threshold	Number	Number	Total stock out	0
type	String	Text	Stock In or Stock Out	IN

The system from Table 6.1 uses a structured system to capture all relevant stock data for recording, updating, and monitoring of stock. The system supports functionality for tracking stock and sending alerts automatically and for reporting. The barcodeUrl attribute contains the barcode images for each item in the system. These images are generated and stored in Base64, serving as a unique identifier and facilitating the scan and retrieval of the item record. The date attribute captures and stores the timestamp for the creation or last update of the item record. This captures a useful data point for maintaining accurate stock/timelines used for traceability purposes. The item attribute stores the inventory item name as a text value and aids the users in product identification for the database.

The lowStock field is a binary flag that indicates whether the item is below its minimum stock level. Setting this value to true activates the system's alert feature for low stock notifications to the administrators. The qty field indicates the total number of that item in inventory. This value is always changing due to stock-in and stock-out transactions. The threshold field indicates the minimum number of that item that is allowed to exist in inventory. When inventory remaining amount is below this figure, that item is marked as low stock. Lastly, the type field indicates the action of the record. It is used to bring clarity between the Stock In action of adding new items, and the Stock Out action of removing items, allowing the system to preserve an accurate picture of stock movements.

6.3.1.1.2 Staff Table

Table 6.2: Data Dictionary – Staff Table

Data Item	Data Type	Data Format	Description	Example
active	Boolean	Text	Approve or Reject access	false
createdAt	Timestamp	Date	Date create the account	October 7, 2025 at 12:50:28 AM UTC+8
email	String	Text	Email of the staff	faris@gmail.com
name	String	Text	Name of the staff	faris
role	String	Text	Role of the staff	Staff
uid	String	NN	Unique identifier for Staff	8ms5VHrhrzYSEvuVEZZwb4aZO4q1

The data stored in table 6.2 describes the structure of data employed in the tracking of the staff accounts in the Smart Inventory System. Each account record contains a number of attributes that enable us to properly distinguish, authenticate, and control access to users. The active field indicates whether staff members account has been approved access to the system and, as a boolean, can be set to true which effects the ability of the user to log in and do system operations, while, false means that user access is blocked or rejected. The attribute createdAt contains the date and time account creation of the staff accounts was done, which is useful for tracking user registration and for administratively maintaining this record.

The email attribute contains the staff email which serves as the primary credential for logging into the system and for all other communications. The name attribute keeps the staff full name for system identification. The role attribute describes the user position or the level of access, such as Admin and Staff, which dictates what functions users can do in the system, enabling role-based access control. Finally, the uid attribute contains the system-

generated identifier for every staff member which is unique. This value is automatically generated by Firebase Authentication and ensures that every staff profile is uniquely linked to login credentials and system activities.

6.3.2 Entity Relational Diagram

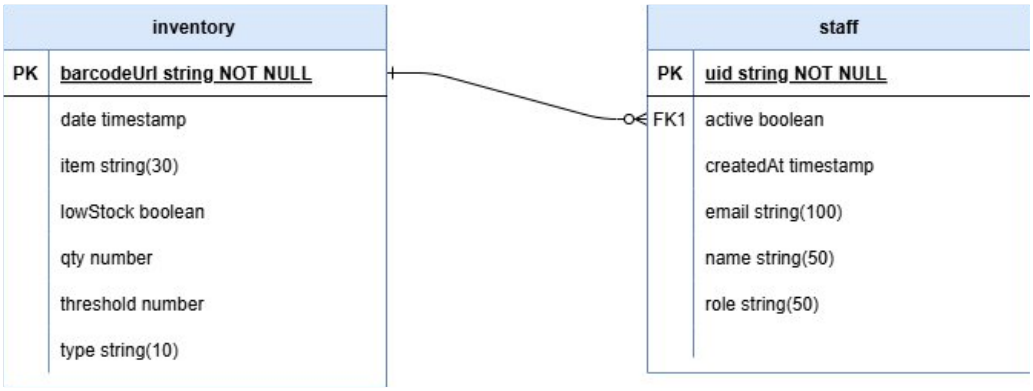


Figure 6.10: Smart Inventory and Tracking System Entity Relationship Diagram (ERD)

The Entity Relationship Diagram (ERD) provides a structural overview of how data is organized within the Smart Inventory and Tracking System. It outlines the relationship between two key entities: Inventory and Staff, both of which are essential for ensuring accurate stock management, system accountability, and secure access control.

The Inventory entity is responsible for storing all information related to stock items. Its primary key, barcodeUrl, uniquely identifies each item through a generated barcode image, ensuring that stock entries are distinct and easily retrievable. Additional attributes such as date, qty, item, lowStock, threshold, and type support core inventory functions including tracking item additions, determining stock availability, and triggering low-stock alerts. These attributes work together to enable automated stock monitoring and efficient recording of stock-in and stock-out activities.

The Staff entity manages all user accounts within the system. Its primary key, uid, ensures that each user has a unique and traceable identifier. Other attributes such as email, name, createdAt, active, and role support user authentication, access management, and activity tracking. Roles such as Admin and Staff determine the level of access and actions each user can perform.

The ERD also includes a foreign key relationship connecting the Inventory and Staff entities. This relationship allows inventory operations, such as updating stock levels, to be linked directly to the staff member who performed them. Such relationships strengthen accountability and ensure that all inventory transactions are securely recorded.

Overall, the ERD offers a clear visual framework of how inventory data and staff information are logically structured. As noted by Boardmix (2024), diagrams like these help represent system components and illustrate how they interact within a wider process, enabling viewers to understand how the system operates as a whole.

6.3.3 Data Flow Diagram

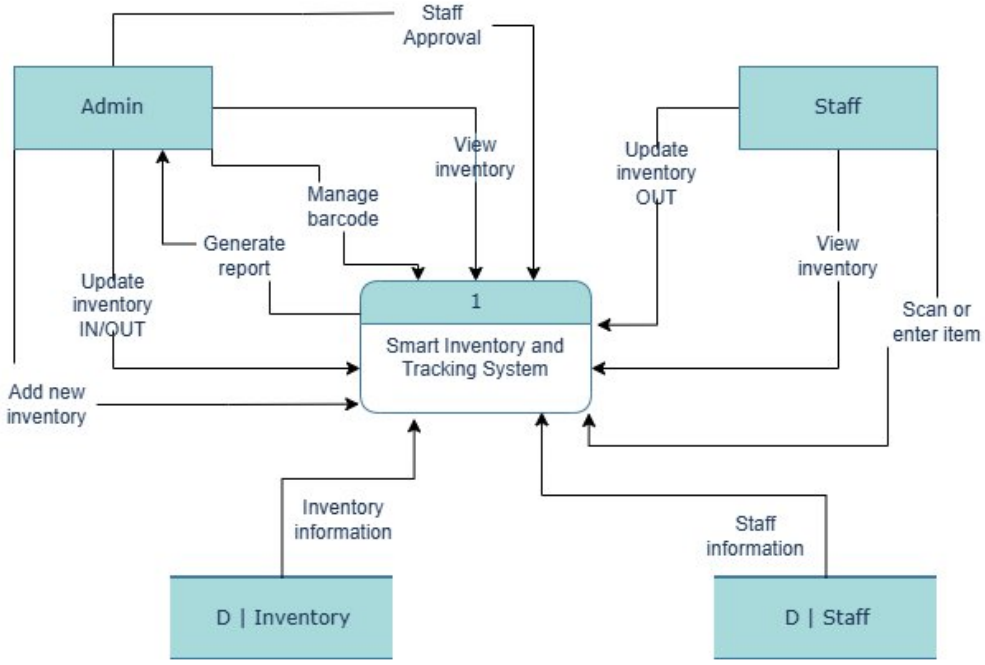


Figure 6.11: Smart Inventory and Tracking System Data Flow Diagram

The Data Flow Diagram (DFD) illustrates how data moves throughout the Smart Inventory and Tracking System. It highlights users, system processes, and data repositories, offering a visual representation of how the system functions. Such diagrams, according to Boardmix (2024), help provide a clear depiction of workflow operations, representing “what happens when” within a system and showing how different system elements interact.

The Admin user plays a major supervisory role within the system. Admins perform tasks such as adding new inventory items, updating stock levels, approving staff accounts, generating reports, and managing barcodes. They also have the ability to view real-time inventory information, which supports operational decision-making and stock monitoring. Their broad set of permissions reflects their responsibility in overseeing the system's integrity and functionality.

The Staff user performs operational tasks such as viewing inventory, scanning barcodes, and updating stock-out records. Staff mainly handle day-to-day stock movement and rely on the system to ensure accurate updates. Their interactions with the system allow for real-time synchronization of inventory data.

At the center of the DFD is the Smart Inventory and Tracking System, which manages all data inputs and outputs. It processes stock movements, generates barcode information, manages staff approvals, and retrieves inventory or staff information when requested. It acts as the core processor connecting all user interactions to the appropriate data stores.

Two data stores support these operations: Inventory and Staff. The Inventory data store contains all item-related details, while the Staff data store houses user credentials and access status. These repositories ensure that information is stored in a structured manner and can be retrieved or updated based on user actions.

In line with the definition by Boardmix (2024), the DFD visually demonstrates the system's workflow by showing tasks, data movements, and processes through symbols and arrows. This allows viewers to understand how users, processes, and data elements interact to produce the desired outputs within the Smart Inventory and Tracking System.

6.4 Flow of the system

A system flowchart is a diagrammatic representation that depicts the workflow or processes involved in a system. It uses a variety of symbols, shapes, and arrows, each representing different types of tasks or steps in a process. The sequence of operations is illustrated by arrows connecting these symbols. System flowcharts can cover a range of scenarios – from manufacturing processes in industries to software operations in information technology, business procedures in organizations, and even biological or chemical processes in science. These charts provide a visual layout of 'what happens when' in a system, giving viewers a comprehensive understanding of how different elements within the system interact with each other to achieve the final output.

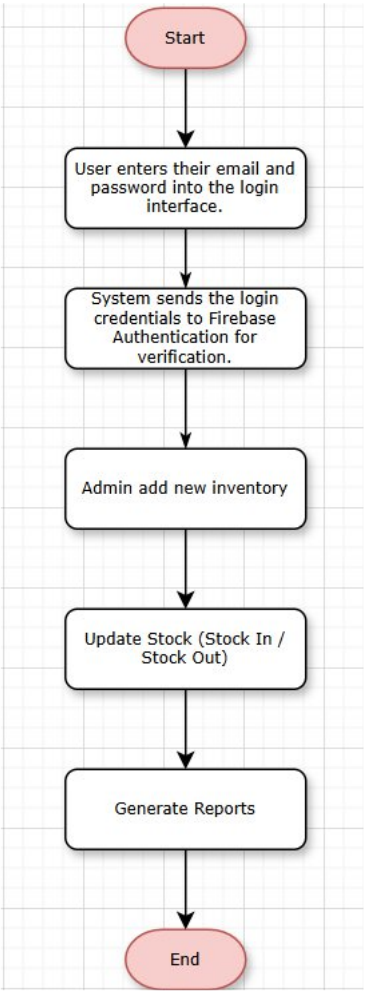


Figure 6.12: Flow of the System Diagram

The above figure showcases the overall flow of Smart Inventory and Tracking System as a web-based and mobile apps tracking and management inventory. Its users must be registered first before being able to use the features as shown in Figure 6.12. The system will determine the user role by its work position, leading them to specific homepages for admin and staff. Each user side will have specific pages cater to their needs when utilizing the system.

6.5 Conclusion

In conclusion, the design of the Smart Inventory Tracking System ensures that user needs, identified in the requirements and analysis stages, are systematically transformed into practical system models. By developing intuitive interfaces, a scalable database design, robust security frameworks, and clear system flows, this project establishes a solid foundation for implementation. The system design ultimately aims to provide SME contractors with a reliable and efficient inventory management tool that enhances productivity and decision-making.

7 IMPLEMENTATION

7.1 Introduction

This chapter emphasizes the implementation phase of the Smart Inventory and Tracking System, highlighting the details of how the project was executed. The implementation stage involves the transition from theoretical design and planning to a fully functional software system, where all modules are developed, integrated, and tested to ensure they meet the needs of the end-users. This section will discuss the tools and technologies used in the development of the system, the actual system interface based on the prepared wireframes, and the significant features implemented to enhance usability, performance, and security. Additionally, it will highlight key functionalities, such as real-time inventory tracking, professional report generation, and data visibility, which are central to achieving the objectives of the project.

7.2 Execution Platform

7.2.1 Windows 11

Figure 7.1: Windows 11

For the development of the Smart Inventory and Tracking System, Windows 11 served as the primary operating system supporting the hardware environment used to develop and execute the web-based application. With its enhanced processing performance, Windows 11 provides a stable and efficient platform, ensuring that the system operates smoothly. The improved startup speed, seamless application transitions, and responsive performance allow developers to run multiple tools and applications simultaneously without lag, which greatly facilitates the execution of project tasks and testing processes during development.

7.3 Implementation Tools

7.3.1 Visual Studio Code

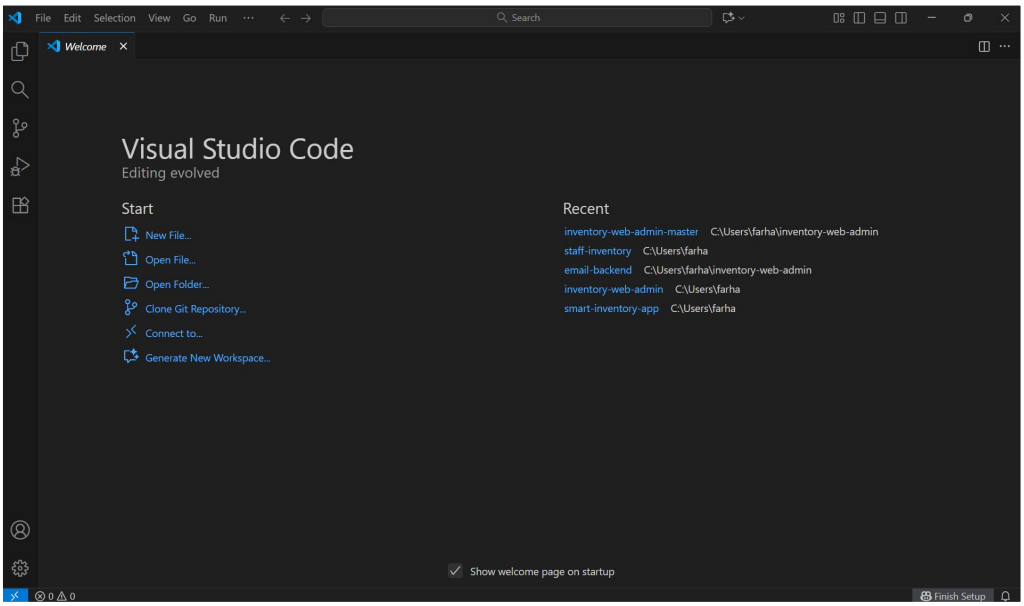


Figure 7.2: Visual Studio Code

Visual Studio (VS) Code is a versatile programming software that supports developers throughout the development process. Its lightweight, simple, and powerful interface makes it an ideal choice for daily programming tasks. Supporting multiple programming languages, VS Code enhances productivity during the implementation of the Smart Inventory and Tracking System by providing features such as syntax highlighting, bracket matching, auto-indentation, box selection, and more, which help streamline the coding process and reduce errors.

According to Microsoft (2016), VS Code offers capabilities such as code auto-completion (IntelliSense), debugging, and an integrated terminal, all of which contribute to improved developer efficiency and code quality. In addition, the extensive extension marketplace allows developers to incorporate tools for real-time previewing, code formatting, and backend programming efficiency. These features make VS Code an attractive choice for developing and maintaining the full application stack, including backend scripts, database integration, and frontend interfaces, ensuring a smooth and efficient implementation process for the system.

7.3.2 Google Firebase

Figure 7.3: Google Firebase (Firebase,2025)

Google Firebase is a comprehensive cloud-based platform that provides backend services to support web and mobile applications. For the Smart Inventory and Tracking System, Firebase serves as the primary backend infrastructure, offering real-time database management, authentication services, cloud storage, and hosting capabilities. Its real-time database allows inventory data to be updated and reflected instantly across all connected devices, ensuring accurate and up-to-date information. Furthermore, Firebase simplifies the integration of user authentication and secure data storage, which enhances both the functionality and security of the system. The scalability and reliability of Firebase make it a suitable choice for managing dynamic inventory data efficiently.

7.3.3 React.Js

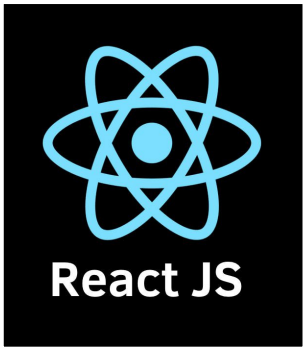


Figure 7.4: React.js (Kasuni Madhushika, 2021)

React.js is a popular JavaScript library used for building dynamic and responsive user interfaces. In this project, React.js is utilized to develop the frontend of the Smart Inventory and Tracking System, providing a seamless user experience. Its component-based architecture allows developers to build reusable and modular components, which simplifies maintenance and enhances the scalability of the application. Features such as virtual DOM rendering and efficient state management contribute to fast and smooth performance, allowing users to interact with the system without noticeable delays. React.js is therefore instrumental in creating an intuitive and responsive interface for both administrators and users.

7.3.4 CSS

Figure 7.5: CSS (OWS, 2025a)

Cascading Style Sheets (CSS) is employed to style and enhance the visual appearance of the system's user interface. CSS enables the creation of responsive layouts, ensuring that the application adapts seamlessly to different screen sizes and devices. By using modern CSS techniques, the system achieves a consistent and professional look, improving usability and user engagement. Animations, color schemes, and typography are also controlled through CSS, which contributes to a polished and user-friendly interface. Overall, CSS plays a critical role in making the system visually appealing and accessible.

7.3.5 Ionic

Figure 7.6: Ionic (Outsystems, 2022)

Ionic is an open-source framework used for building cross-platform mobile applications using web technologies. For the Smart Inventory and Tracking System, Ionic facilitates the development of a mobile application that is compatible with both Android and iOS devices. Its integration with Angular and Capacitor allows for seamless access to native device features, such as the camera and push notifications, which are essential for inventory tracking and user alerts. By leveraging Ionic, the project can provide a consistent and responsive mobile experience without the need to develop separate applications for different platforms, reducing development time and effort.

7.3.6 Hardware

Table 7.1: Hardware Specifications Table

Device	HP Laptop 15
Windows Edition	Windows 11 Home Single Language
Processor	AMD Ryzen 5 5500U with Radeon Graphics 2.1 GHz
Memory (RAM)	8 GB
System Type	64-bit operating system, x64-based processor

7.4 System Interface

7.4.1 Admin Interface

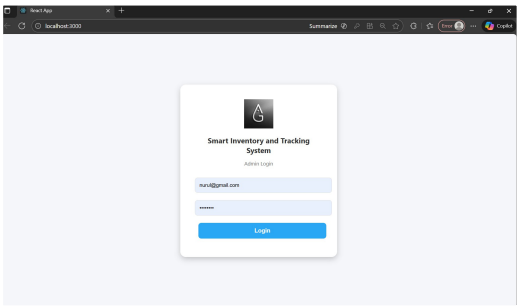


Figure 7.7: Login page

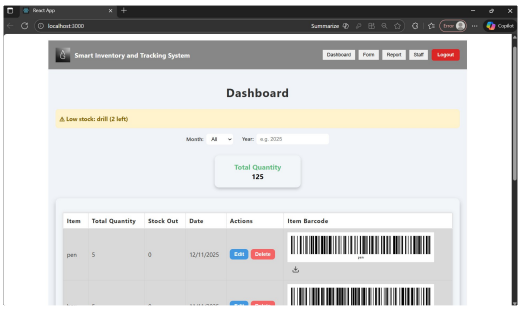


Figure 7.8: Dashboard page - Edit, Delete Item

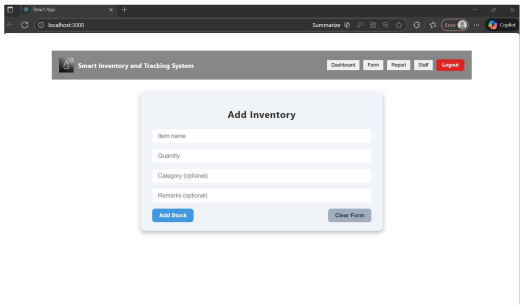


Figure 7.9: Add new inventory page

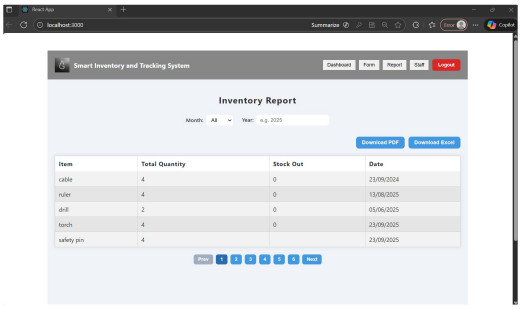


Figure 7.10: Admin - Generate inventory report

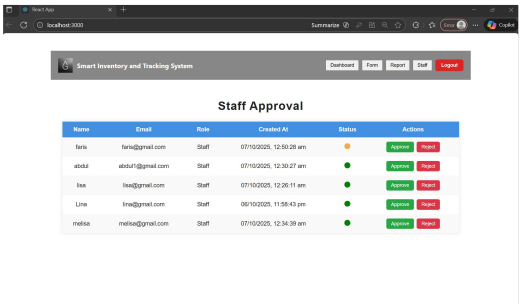


Figure 7.11: Admin - Staff aproval for Mobile Apps

The figures above illustrate the various interfaces and pages of the Smart Inventory and Tracking System for the Admin panel. While admins can access many of the core functionalities available to regular users, such as managing inventory and viewing stock movement, the admin panel also includes additional features to enhance control and oversight. Admins can generate professional inventory reports, view and manage all inventory items, and monitor low-stock alerts in real time. Furthermore, admins are able to oversee user accounts, track system activity, and approve requests for inventory adjustments. These enhanced capabilities provide administrators with comprehensive control over the system, ensuring accurate inventory management and efficient operational oversight.

7.4.2 Staff Interface

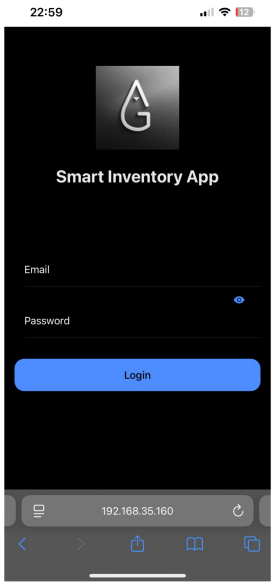


Figure 7.12: Staff login page

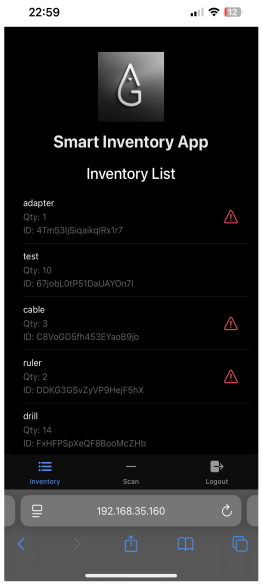


Figure 7.13: Staff - Inventory List

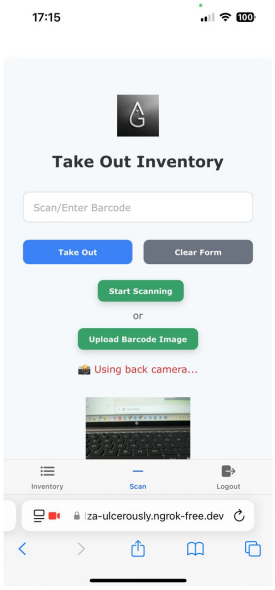


Figure 7.14: Take Out Inventory - Enter, Scan or Upload

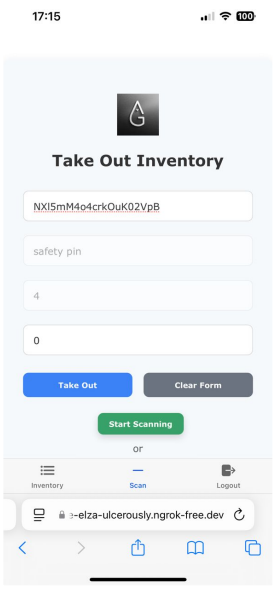


Figure 7.15: Take Out Stock Form

The figures above illustrate the various interfaces and pages of the Smart Inventory and Tracking System for the Staff panel. Staff can perform essential inventory-related tasks such as reduce stock, viewing inventory list, and update stock levels in real time by enter ID, scan or upload image. These features provide staff with the necessary tools to perform daily inventory management tasks efficiently while ensuring that all actions are accurately recorded in the system.

7.5 Significant function

7.5.1 Database

```
// Fetch inventory data
useEffect(() => {
  const q = query(collection(db, "inventory"), orderBy("date", "desc"));
  const unsub = onSnapshot(q, async (snapshot) => {
    const items = snapshot.docs.map(docSnap => ({
      id: docSnap.id,
      ...docSnap.data()
    }));
```

Figure 7.16: Code segment for database

The database serves as the core foundation of the Smart Inventory and Tracking System, storing all essential information such as inventory items, user accounts, stock movement history, and system activity logs. Firebase Firestore is used as the main database due to its scalability, fast performance, and real-time data synchronization capabilities. Every update made to the inventory—whether stock-in, stock-out, or item modification—is stored securely and instantly reflected across all devices. This ensures data consistency, reliability, and accuracy, which are crucial for effective inventory management. The database also supports secure data access through Firebase Authentication and role-based permissions.

7.5.2 Login

```
src > components > JS Login.js > Login > handleSubmit
import React, { useState } from "react";
import { auth } from "../firebase/config";
import { signInWithEmailAndPassword } from "firebase/auth";
import "../login.css"; // custom CSS

const Login = ({ onLogin }) => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");

const handleSubmit = async (e) => {
  e.preventDefault();
  setError("");
  try {
    // Sign in with Firebase
    const userCredential = await signInWithEmailAndPassword(auth, email, password);
    const user = userCredential.user;

    // ✅ Allow only this specific admin email
    const adminEmail = "nurul@gmail.com"; // <-- replace with your real admin email
```

Figure 7.17: Code segment for login

The login function ensures that only authorized users can access the system. Using Firebase Authentication, the system provides secure login through email and password verification. This feature also includes error messages for invalid credentials and session management to maintain user identity throughout the system. Role-based access control is implemented to differentiate between Admin and Staff users, ensuring that each role only accesses features relevant to their responsibilities. This enhances system security and prevents unauthorized data manipulation.

7.5.3 Real-Time Inventory Monitoring

```
// Auto-update lowStock status based on live data
for (const item of items) {
  const currentQty = Number(item.qty || 0);
  const threshold = Number(item.threshold || 0);
  const shouldBeLow = currentQty <= threshold;

  // only update if Firestore value mismatches current condition
  if (item.lowStock !== shouldBeLow) {
    await updateDoc(doc(db, "inventory", item.id), {
      lowStock: shouldBeLow
    });
  }
}
```

Figure 7.18: Code segment for real-time inventory

Real-time inventory monitoring is one of the system's most important features, allowing users and administrators to view up-to-date stock levels instantly. Any changes made to the database—such as adding new items, updating quantities, or removing stock—are immediately displayed in the interface without requiring page refresh. This function helps prevent delays, reduces errors, and provides an accurate overview of the organization's inventory status at all times. Real-time visibility also supports timely decision-making, especially when addressing sudden stock shortages or adjusting stock allocation.

7.5.4 Inventory Management

```
const handleDelete = async (id) => {
  await deleteDoc(doc(db, "inventory", id));
};

```

Figure 7.19: Code segment for delete inventory

The Delete function allows administrators to remove outdated, damaged, or unnecessary stock records from the system. This is crucial for maintaining data accuracy and preventing clutter within the inventory database. When an admin selects an item for deletion, the system retrieves the corresponding document from the Firestore collection and permanently removes it. A confirmation prompt is commonly implemented to prevent accidental deletion, ensuring data integrity is preserved. This function helps the system remain organised and ensures that reports and stock summaries reflect only active, relevant items.

```
160  const handleEditClick = (entry) => {
161    setEditId(entry.id);
162    setEditData({
163      item: entry.item,
164      qty: entry.qty,
165      threshold: 0
166    });
167  };
```

Figure 7.20: Code segment for edit inventory

The Edit function enables administrators to update existing inventory information such as item name, quantity, category, supplier details, or stock condition. This feature is essential as inventory values often change due to restocking, item usage, or corrections in previous entries. Upon editing, the updated data is immediately written back to the Firestore database, ensuring real-time accuracy across both the web-based admin panel and mobile application. This capability supports dynamic inventory management, reduces manual errors, and ensures that all team members—especially field staff—have access to the most current and accurate stock information.

7.5.5 Low-Stock Alerts and Notifications

```
216    <div className="dashboard-container">
217      <h1>Dashboard</h1>
218
219      {/* Low Stock Alerts Panel */}
220      {alerts.length > 0 && (
221        <div className="alert-panel">
222          {alerts.map(alert => (
223            <div key={alert.id} className="alert-item">
224              ⚠ Low stock: {alert.item} ({alert.qty} left)
225            </div>
226          ))}
227        </div>
228      )}
```

Figure 7.21: Code segment for low stock alert below 2

This feature automatically notifies users when inventory levels fall below a predefined threshold. The system checks stock quantities in real time and triggers alerts when items require replenishment. These notifications enable proactive planning by helping administrators identify which products need to be restocked before they run out. This reduces the risk of stockouts, enhances operational efficiency, and supports smoother workflow management. The low-stock alert function also contributes to maintaining consistent inventory levels, which is essential for uninterrupted business operations.

7.5.6 Search and Filter Functionality

```
const handleFilterChange = (month, year) => {
  setFilterMonth(month);
  setFilterYear(year);
  applyFilter(data, month, year);
};
```

Figure 7.22: Code segment for filter month or year

This feature automatically notifies users when inventory levels fall below a predefined threshold. The system checks stock quantities in real time and triggers alerts when items require replenishment. These notifications enable proactive planning by helping administrators identify which products need to be restocked before they run out. This reduces the risk of stockouts, enhances operational efficiency, and supports smoother workflow management. The low-stock alert function also contributes to maintaining consistent inventory levels, which is essential for uninterrupted business operations.

7.5.7 Mobile Compatibility

```
14  import '@ionic/react/css/core.css';
15  import '@ionic/react/css/normalize.css';
16  import '@ionic/react/css/structure.css';
17  import '@ionic/react/css/typography.css';
18  import '@ionic/react/css/padding.css';
19  import '@ionic/react/css/float-elements.css';
20  import '@ionic/react/css/text-alignment.css';
21  import '@ionic/react/css/text-transformation.css';
22  import '@ionic/react/css/flex-utils.css';
23  import '@ionic/react/css/display.css';
24
25  import '@ionic/react/css/palettes/dark.system.css';
26  import './theme/variables.css';
27
28  setupIonicReact();
29
```

Figure 7.23: Code segment for ionic

Mobile compatibility ensures that the system can be accessed through smartphones and tablets, allowing staff to manage inventory on the go. Using Ionic and responsive React components, the system provides a smooth experience across different screen sizes and devices. This is particularly important for field staff or teams working outside the office, as they can update stock levels, scan items, or review inventory information from their mobile devices. Mobile compatibility increases flexibility, improves workflow efficiency, and supports real-time inventory updates anywhere and anytime.

7.6 Conclusion

This chapter provides a comprehensive explanation of the tools and technologies used to develop the Smart Inventory and Tracking System, covering software, programming languages, and hardware specifications. The frontend of the system was built using React.js and CSS, ensuring a responsive, structured, and user-friendly interface. This approach guarantees that the overall look and feel of the application, including layout, colors, fonts, and interactive elements, remain consistent and visually appealing across different devices.

The system interfaces presented in this chapter align closely with the requirements established during the design and planning stages, demonstrating that the implementation effectively translates theoretical designs into functional components. Essential functionalities, such as real-time inventory tracking, professional report generation, user authentication, and role-based access control have been successfully implemented. These features ensure that both administrators and staff can perform their tasks efficiently, while the system maintains accurate and secure inventory data. Overall, the implementation phase confirms that the Smart Inventory and Tracking System meets its intended objectives and provides a reliable platform for effective inventory management.

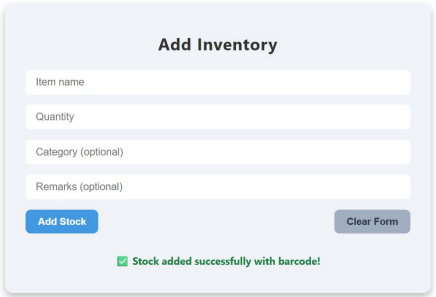
8 TESTING

8.1 Introduction

This chapter presents the testing approaches used to evaluate the Smart Inventory and Tracking System, ensuring that every component functions as intended. The testing process is essential in determining the system's functionality, usability, and overall performance. Several testing methods were applied, including Unit Testing, Integration Testing, System Testing, and Acceptance Testing, to verify that the system meets the specified requirements and performs reliably for both administrators and field users.

8.2 Unit Testing

Table 8.1: Unit TestingTable

Unit/Function	Description	Input	Output	Expected Error	Status
Login	Verifies user authentication using email and password	Valid email + valid password	User successfully logs into the system	Incorrect email/password error if invalid	Pass
Barcode scanner	Tests camera ability to detect and decode barcode	Valid item barcode	Item details displayed instantly	"Barcode not detected" if unclear	Pass
Add inventory	Adds new inventory item into database	Item name, category, quantity	Item saved in Firebase and appears in list 	Missing fields or invalid quantity	Pass
Stock Out update	Updates stock quantity after removal	Item name + amount to reduce	Stock value decreases in real-time	Negative stock or invalid quantity	Pass
Edit inventory	Edits existing item details	Updated quantity	Updated item saved and displayed	Invalid data format	Pass
Delete	Removes	Item name	Item deleted and removed from	Non-	Pass

inventory	selected item from dashboard		list	existent item error	
Real-time inventory monitoring	Checks live synchronization of stock changes	Item edited or stock-out triggered	Dashboard updates instantly without refresh	Delay in real-time update	Pass
Low stock alert	Shows warning when stock falls below threshold	Quantity below set limit < 2	Alert message appears 	Alert not triggered	Pass
Filter function	Filters items based on month or year	Month or Year selection	List displays month and year 	No items in month or year	Pass
Report generation (PDF)	Generates inventory report in PDF format	Click “Download PDF”	PDF downloaded successfully 	PDF fails to generate	Pass
Report generation (Excel)	Generates inventory report in Excel format	Click “Download Excel”	Excel file downloaded successfully 	File export failure	Pass
Mobile Compatibility	Ensures UI works on mobile screens	Open system on mobile device	Responsive layout appears	Layout not responsive	Pass
Logout	Logs user out securely	Click “Logout”	Redirect to login page	Session not cleared	Pass

8.3 Integration Testing

Table 8.2: Integration Testing Table

Test Case	Description	Input	Output	Expected Error	Status
Login to Dashboard	Test integration between login module and dashboard display	Valid login credentials	Redirects to dashboard with real-time inventory data	Incorrect credentials	Pass
Barcode Scan to Item Display	Test integration between camera scanner and item database	Valid barcode	Item details retrieved and displayed	Item not found	Pass
Stock-Out Update to Real-Time Database	Check stock-out form integration with Firebase real-time database	Item ID + quantity	Quantity updates across all connected pages	Update delay	Pass
Edit Item to Report Generator	Ensure edited item data updates in report	Updated item details	Updated values appear in PDF & Excel	Missing updates	Pass
Low-Stock Alert to Dashboard Notification	Test threshold check and alert system	Item below threshold	Alert appears on dashboard	Alert not triggered	Pass
Filter to Inventory Table	Check integration between filter inputs and displayed data	Month or Year	Filtered list appears	No matching results	Pass
Mobile UI to Functional Modules	Ensure mobile UI triggers correct backend functions	App opened on mobile	All features work responsively	Layout errors	Pass

8.4 System Testing

8.4.1 Functional Testing

Table 8.3: Functional Testing Table

#	Test Function	Expected Result	Actual Result	Status
1	Login	User logs in successfully	Works as expected	Pass
2	Barcode scanning	Item detected and shown	Works smoothly	Pass
3	Add inventory	New item stored in database	Item added correctly	Pass
4	Edit inventory	Item updates saved	Updated successfully	Pass
5	Delete inventory	Item removed from list	Deleted correctly	Pass
6	Stock-out update	Stock quantity decreases in real-time	Displays immediately	Pass
7	Filter	Month and Year filtered accurately	Works smoothly	Pass
8	Low-stock alerts	Alerts triggered when below threshold	Alerts appear correctly	Pass
9	PDF & Excel report generation	Files downloaded successfully	Files generated correctly	Pass
10	Logout	User session cleared and redirected	Works as expected	Pass

8.4.2 Non-functional testing

Table 8.4: Non-functional Testing Table

#	Test Function	Expected Result	Actual Result	Status
1	Performance	Stock updates instantly	Updates immediately	Pass
2	Usability	Pages transition smoothly	Navigation smooth	Pass
3	Reliability	Stable connection	No errors detected	Pass
4	Security	Login authentication	Only valid users can access	Pass

8.5 User Acceptance Testing

Due to time constraints, User Acceptance Testing (UAT) for the Smart Inventory and Tracking System was conducted on a limited scale. A demonstration session was carried out with the project supervisor and the intended system administrator. During this session, key system functionalities such as adding and updating inventory records, real-time stock monitoring, low-stock alerts, search and filter operations, and mobile accessibility were presented. Feedback regarding the system's usability, accuracy of inventory updates, interface design, and overall user experience was gathered.

Although a full-scale UAT involving multiple end users (such as admin or field staff) could not be conducted, the feedback received from the administrator was essential in assessing the system's readiness for practical use. It helped identify minor refinements and confirmed that the core features function as expected. For future development, a more extensive UAT involving a wider group of users is recommended to obtain broader insights and ensure the system meets the operational needs of real-world SME environments.

8.5.1 Client testing and result

Figure 8.1: Interviewee Picture

- **Interviewee Name:** Puan Nuraza Binti Jamaluddin
- **Position:** HR Administrator
- **Location:** Avant Grade Solution (M) Sdn. Bhd, No 1 Jalan Budiman 2 Bandar Tun Razak 56000 Cheras Kuala Lumpur
- **Date:** 16 October 2025
- **Purpose:** her interview aims to gather necessary system feedback for the User Acceptance Testing Process.

8.5.2 Testing and result – Functionality Feedback (Admin Side)

Table 8.5: Interview Question - 1

Question 1	How do you feel about the system's overall performance?
Answer	The system works efficiently and performs well even when multiple users access it simultaneously.
Analysis	The admin's response indicates a positive performance evaluation. The system demonstrates stability and reliability under concurrent use, which suggests that the backend (Firebase) handles real-time data synchronization effectively.

Table 8.6: Interview Question - 2

Question 2	How user-friendly is the admin interface for managing inventory?
Answer	The interface is intuitive and easy to navigate, especially the dashboard and report features.
Analysis	The admin particularly values the dashboard layout and reporting tools, highlighting that the design meets usability principles like clarity and efficiency

Table 8.7: Interview Question - 3

Question 3	Does the system accurately track stock in and stock out?
Answer	Yes, it provides accurate real-time updates and helps reduce manual tracking errors.
Analysis	The admin's remarks demonstrate that the system successfully fulfills one of its core functional requirements are accurate and real-time inventory tracking.

Table 8.8: Interview Question - 4

Question 4	How helpful do you find the low-stock notification feature?
Answer	It's very useful. I can restock items immediately without checking inventory manually.
Analysis	The low-stock notification feature enhances proactive stock control, enabling the admin to avoid shortages.

Table 8.9: Interview Question - 5

Question 5	Do you find the generated reports (PDF/Excel) clear and professional?
Answer	Yes, the reports are neat and easy to share during monthly meetings.
Analysis	The admin appreciates the clarity and presentation quality of the generated reports, confirming that the data visualization and export features meet business reporting standards.

Table 8.10: Interview Question - 6

Question 6	What improvements would you suggest for the admin system?
Answer	Maybe add a chart to visualize monthly stock movement trends more clearly.
Analysis	This indicates potential for future UI improvement through graphical dashboards

Table 8.11: Interview Question - 7

Question 7	How do you feel about the integration between the mobile app and the admin system?
Answer	It's seamless because data from staff updates appear instantly on the admin dashboard.
Analysis	The admin's positive view confirms that the Firebase real-time synchronization between mobile and web systems operates effectively.

Table 8.12: Interview Question - 8

Question 8	Does the system help you save time in managing inventory records?
Answer	Definitely. It saves hours of manual work and minimizes errors.
Analysis	The admin highlights significant time savings and efficiency gains, showing the system's impact on workflow optimization.

Table 8.13: Interview Question - 9

Question 9	Are you satisfied with the system's security and access control?
Answer	Yes, only authorized users can make edits, which ensures data integrity.
Analysis	This response validates the system's role-based authentication and authorization design. It ensures data protection and prevents unauthorized changes.

Table 8.14: Interview Question - 10

Question 10	Would you recommend this system for future company use?
Answer	Absolutely. It's efficient, reliable, and suitable for SME operations.
Analysis	The recommendation suggests that the Smart Inventory Tracking System successfully meets organizational needs, making it a scalable solution for other SMEs

8.5.3 Testing and result – Functionality Feedback (Staff Side)

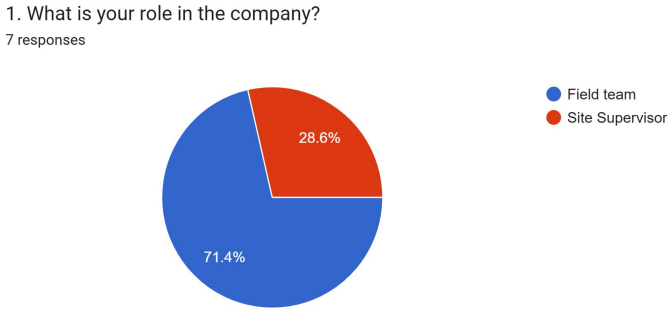


Figure 8.2: Questionnaire - 1

Most respondents identified themselves as part of the field team or site supervisors, which aligns well with the target users of the system. This ensures that the feedback is relevant and accurately reflects real operational needs.

Figure 8.3: Questionnaire - 2

All respondents indicated that the login process was easy and worked without issues. This shows that the authentication flow is user-friendly, stable, and well-integrated with Firebase.

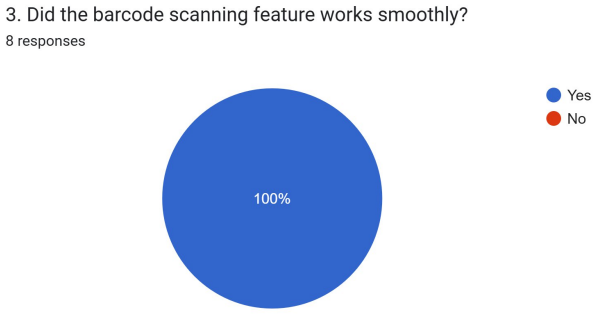


Figure 8.4: Questionnaire - 3

Feedback confirmed that the barcode scanning function worked smoothly, without lag or scanning errors. This indicates that the camera integration and ZXing library are operating efficiently for real-world usage.

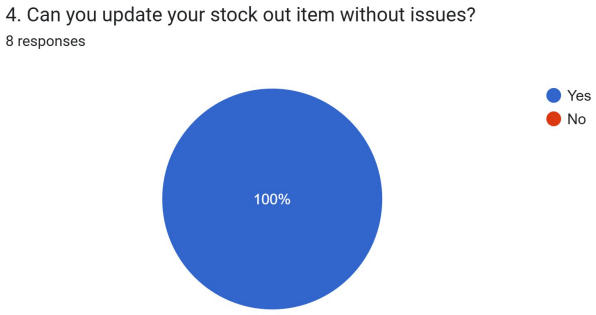


Figure 8.5: Questionnaire - 4

Staff reported that they could update stock-out items without any problems, demonstrating that the inventory update logic, database write functions, and form validation are functioning correctly.

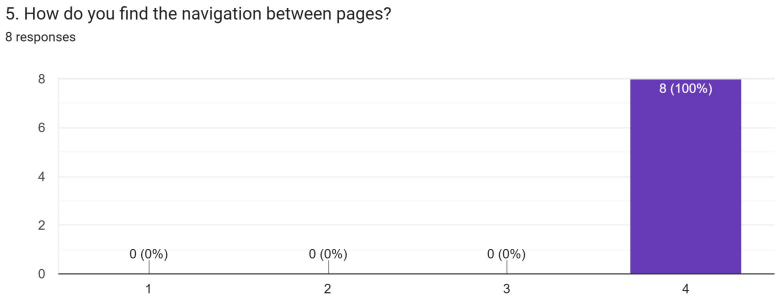


Figure 8.6: Questionnaire - 5

All users rated the navigation experience between pages as very smooth, indicating that the app layout is intuitive, transitions are quick, and pages are well-organized for field operations.

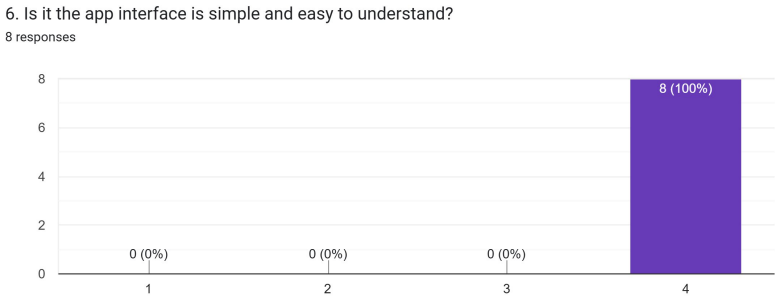


Figure 8.7: Questionnaire - 6

Respondents rated the interface as simple and easy to understand. This shows that the UI design meets usability expectations and supports fast task completion, even for non-technical users.

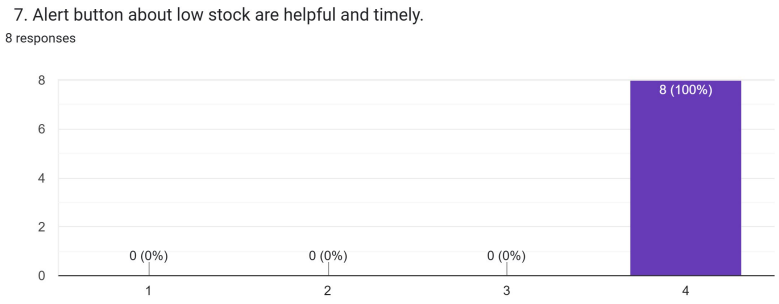


Figure 8.8: Questionnaire - 7

Staff strongly agreed that the low-stock alert button was helpful and timely. This confirms that the threshold-based alert system works reliably and provides valuable notifications for restocking decisions.

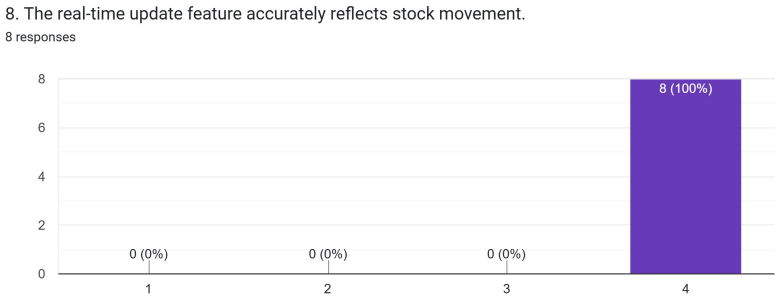


Figure 8.9: Questionnaire - 8

Respondents strongly agreed that the system accurately displays stock movement in real time. This demonstrates that Firebase's real-time synchronization is working properly and reflects live inventory changes.

Figure 8.10: Questionnaire - 9

All respondents agreed that the app helps them complete stock-related tasks more efficiently. This suggests that the system reduces manual effort, speeds up workflows, and supports on-site operations effectively.

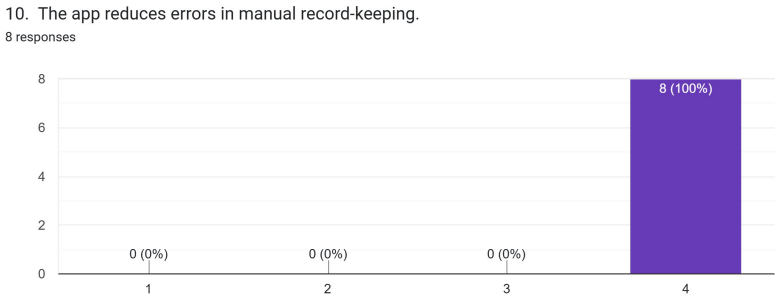


Figure 8.11: Questionnaire - 10

Staff strongly agreed that the system reduces errors in manual record-keeping, showing that automation, barcode scanning, and real-time updates successfully eliminate common human mistakes.

8.6 Conclusion

This chapter focuses on several categories of software testing, including unit testing, integration testing, system testing, and user acceptance testing. These testing approaches are essential in evaluating the system's readiness throughout each phase of development. System testing plays a crucial role in verifying that all components of the Smart Inventory and Tracking System function correctly when integrated as a complete solution. Meanwhile, user acceptance testing (UAT) serves as a critical assessment to determine whether the system meets the requirements and expectations of the intended users, as defined during the earlier stages of planning and analysis.

9 Project Management

9.1 Introduction

Project management is a systematic approach to planning, executing, monitoring, and completing project activities to achieve specific objectives within defined constraints such as time, budget, and available resources. It incorporates established principles, skills, methodologies, and management tools to ensure that project requirements are met and successful outcomes are delivered by the end of the project cycle (Project Manager, 2023). Effective project management enables tasks to be carried out efficiently, promotes coordination among stakeholders, and ensures that potential risks are identified and addressed throughout the project duration. This chapter highlights the significance of time management in guiding the development of the Smart Inventory and Tracking System, ensuring that each project component is completed within schedule and contributes to the overall system's success.

9.2 Project Scheduling

A project schedule is a critical document that outlines the timeline of a project, including the tasks involved, their duration, dependencies, milestones, and allocated resources such as tools and equipment. It serves as a blueprint or road map that guides the project execution, allowing progress to be tracked, deadlines to be managed, and potential delays to be identified. Preparing a detailed schedule to manage and monitor tasks such as designing the system interface, performing quality testing, and gathering user feedback is essential due to the complexity of the Smart Inventory and Tracking System. Given the constraints of time, each task must be executed efficiently to ensure high-quality outcomes. A carefully planned schedule also helps mitigate possible disruptions or delays, ensuring that the system is developed effectively and according to the defined plan.

9.2.1 Work Breakdown Structure

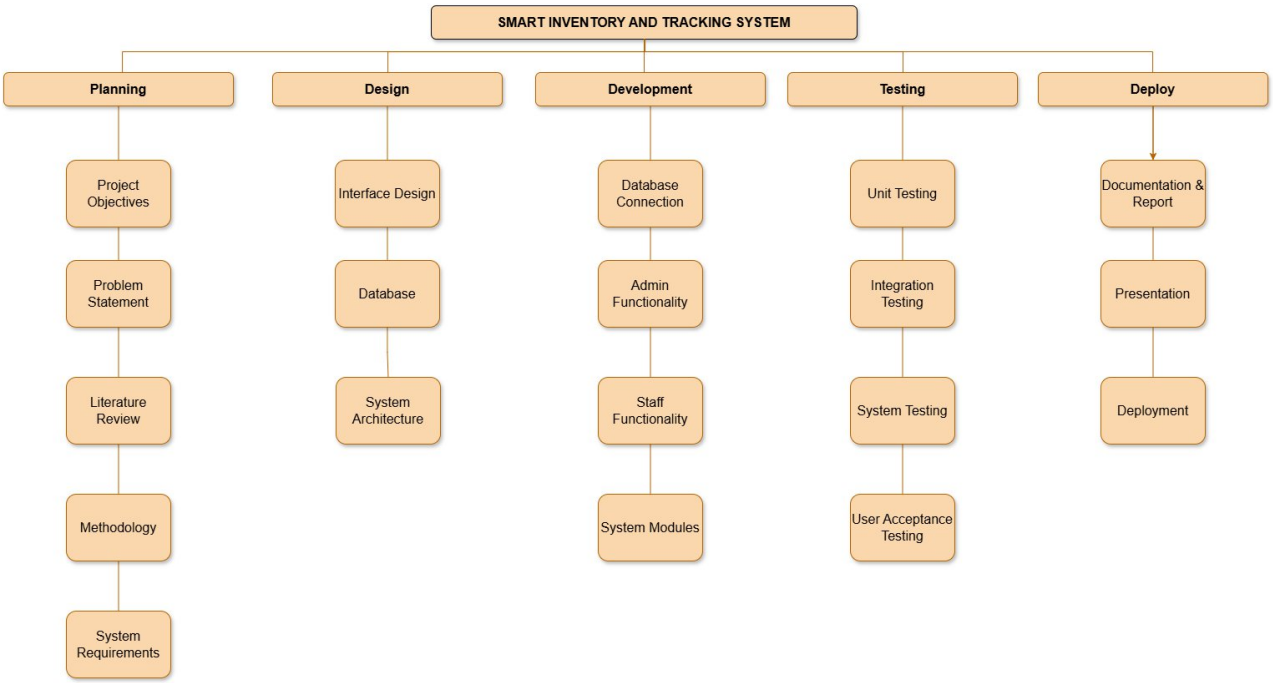


Figure 9.1: Work Breakdown Structure

The development of the Smart Inventory and Tracking System follows a structured and systematic approach to ensure the creation of a reliable, functional, and user-friendly system. The entire process is divided into five major phases such as Planning, Design, Development, Testing, and Deployment each containing a series of tasks that guide the project from its early conceptualization to its final implementation. This essay explains each phase in detail based on the workflow illustrated in the diagram.

The first phase, Planning, serves as the foundation of the project. This phase begins with identifying the project objectives, which outline what the Smart Inventory and Tracking System aims to achieve. These objectives typically include improving accuracy in inventory tracking, reducing human errors, and enabling real-time stock management. Following the objectives, a problem statement is created to highlight the issues in existing inventory practices, such as manual errors, delays in stock updates, and inefficient tracking processes. To support the development direction, a literature review is carried out to study previous research, existing technologies, and similar systems. This review helps determine suitable features and tools that should be incorporated. The methodology is then selected, indicating whether the project adopts an agile, waterfall, or hybrid approach. Finally, system requirements are documented, covering both functional and non-functional needs such as user roles, security, performance, and usability.

The Design phase translates the planning outcomes into structured system blueprints. Interface design is developed to outline how users will interact with the system, including layout, navigation, and overall aesthetics. This is followed by database design, which identifies all necessary data tables, relationships, and storage structures required for effective inventory tracking. System architecture is also created to define how different components and modules of the system will interact with one another. This includes diagrams showing data flow, system workflow, and technical structure. Through careful planning and visualization, the design phase ensures that developers have a clear roadmap before coding begins.

Once the system design is finalized, the project proceeds to the Development phase. This is where the actual system is built. The first step involves setting up the database connection, enabling the system to store and retrieve information properly. Admin functionality is then developed, providing administrators with advanced features such as managing stock, generating reports, and overseeing user activities. Staff functionality is also implemented to allow regular users to perform daily inventory operations like updating stock, checking item availability, and recording item movements. Additional system modules—such as login authentication, dashboards, and notification systems—are created and integrated during this phase. By the end of the development stage, the entire system begins to take shape as a working product.

After development, the system undergoes a comprehensive Testing phase to ensure its reliability and performance. Unit testing is conducted first to verify that each individual function works correctly. This is followed by integration testing, which checks whether different modules operate smoothly together. System testing is then carried out to evaluate the performance, security, and overall functionality of the full system. Finally, user acceptance testing (UAT) allows real users to test the system in a practical environment. Their feedback is gathered to identify any issues or improvements needed before final deployment. This phase is crucial as it ensures that the system is stable and user-friendly.

The final phase, Deployment, prepares the system for official use. Before deployment, documentation and reports are completed, providing detailed explanations of the system's design, functionality, and technical aspects. A formal presentation is usually conducted to demonstrate the system's features, workflow, and benefits to stakeholders or evaluators. After approval, the system is deployed into the live environment, making it accessible for actual users. This marks the completion of the development process and the beginning of the system's operational phase.

9.2.2 Gantt Chart

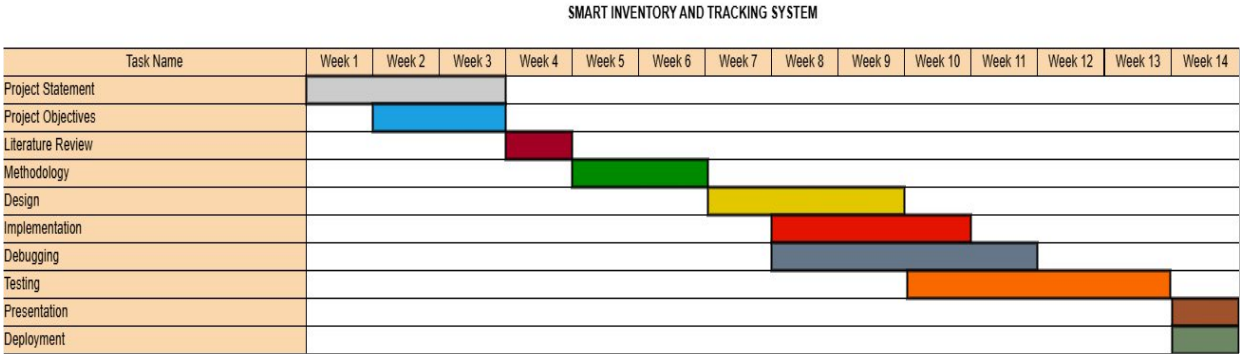


Figure 9.2: Gantt Chart

The Gantt chart for the Smart Inventory and Tracking System outlines the timeline and sequence of activities involved in completing the project over a fourteen-week period. Each task is scheduled in a logical order to ensure smooth development, testing, and final deployment of the system. The chart also illustrates how certain tasks overlap, showing the parallel work and dependency between phases.

The project begins in Week 1 with the Project Statement, where the main problem and purpose of the system are defined. This foundational work is followed immediately by the Project Objectives, which are carried out from Week 2 to Week 3. During this time, the project goals are established to provide clear direction for the entire development process. Once the objectives are set, the Literature Review is conducted in Week 4. This stage involves researching existing systems, technologies, and academic references that support the development of the Smart Inventory and Tracking System.

Moving into Week 5, the project focuses on selecting an appropriate Methodology for the system. This step is crucial because it determines the structure and workflow that will guide the team throughout the development process. After the methodology is finalized, the project transitions into the Design phase from Week 6 to Week 7. This includes creating the system architecture, interface design, and database design, ensuring that the developers have a clear blueprint to follow.

From Week 7 to Week 10, the project enters the Implementation stage, which is the core of the development phase. This task involves writing code, building system modules, connecting the database, and developing functionalities for both admin and staff users. Implementation overlaps with the Debugging stage, which begins in Week 8 and continues to Week 11. Debugging focuses on identifying and fixing errors to ensure that all components of the system run smoothly.

Following debugging, the project moves into Testing, starting from Week 10 until Week 13. During this period, multiple levels of testing such as unit testing, integration testing, system testing, and user acceptance testing are conducted. The aim is to ensure that the system functions correctly, is free from major defects, and meets the requirements set earlier in the project. After testing, the project progresses to the Presentation phase in Week 13, where the system is demonstrated to stakeholders or evaluators. This stage allows the team to showcase the features, workflow, and results of the system.

Finally, the project concludes with the Deployment phase in Week 14. This is where the system is officially launched and made operational for end-users. Deployment marks the completion of the project, as the system is now ready for real-world application.

9.3 Risk Management

According to McGuire (2024), risk management is a critical process that involves identifying, assessing, and mitigating potential threats that could impact the successful outcome of a software project. During the development of the Smart Inventory and Tracking System, several key risks were identified, including technical risks, scheduling risks, and data loss risks, which could potentially occur during development or after deployment. Proper risk management strategies are essential to minimize the impact of these risks and ensure the project is completed successfully and operates reliably in real-world use.

Table 9.1: Risk Management Table

#	Risk Identification	Description	Risk Analysis	Mitigation Plan
1	Technical Risk	Issues with system functionality, bugs, or failure of integration between modules (e.g., barcode scanner, real-time updates).	High – could delay development or affect system performance.	Conduct thorough unit and integration testing, implement code review, and maintain proper documentation for troubleshooting.
2	Scheduling Risk	Delays in completing tasks due to resource unavailability or underestimated task duration.	Medium – may impact overall project timeline.	Use Gantt chart and WBS to plan tasks, set realistic deadlines, and monitor progress weekly to adjust schedules as needed.
3	Data Loss Risk	Loss or corruption of inventory data due to database failure or accidental deletion.	High – could compromise reliability and user trust.	Implement regular backups, Firestore security rules, and version control to restore data when necessary.
4	Performance Risk	System may experience slow response times, especially with large inventory or real-time updates.	Medium – could affect usability and efficiency.	Optimize database queries, implement efficient data handling, and test performance under load conditions.
5	Security Risk	Unauthorized access to the system or sensitive inventory data.	High – could compromise data integrity and confidentiality.	Use role-based authentication, secure login, encrypted connections, and access control rules in Firebase.

The risk management process for the Smart Inventory and Tracking System involves identifying potential risks, assessing their impact, and implementing mitigation strategies to ensure the project is completed successfully and operates reliably. Smart Inventory and Tracking System provide a proactive framework to identify, assess, and mitigate potential challenges. By addressing these risks, the project ensures reliability, security, and timely delivery, ultimately supporting the system's effectiveness for end-users and administrators.

9.4 Conclusion

This chapter emphasizes the importance of time management in ensuring that the development of the Smart Inventory and Tracking System meets its defined objectives. Project management tools, such as the Work Breakdown Structure (WBS) and Gantt Chart, provide a clear overview of the tasks that need to be executed and their respective timelines, helping to ensure that the project progresses according to plan. Additionally, a risk management analysis has been conducted to identify potential issues that could cause delays, allowing proactive measures to be taken to keep the project on schedule and ensure timely completion.

10 Conclusion

10.1 Introduction

This chapter concludes the project by highlighting the key achievements of the Smart Inventory and Tracking System. It reflects on the successful implementation of the system and the objectives it has managed to fulfil in improving inventory monitoring, stock accuracy, and real-time tracking. In addition, this chapter discusses the constraints and limitations encountered during the system's development, offering insight into the technical and operational challenges faced during implementation. Lastly, it outlines several recommendations and future enhancements that could further expand the system's functionality, performance, and overall efficiency in future development work.

10.2 Achievement

The Smart Inventory and Tracking System has successfully fulfilled all the objectives outlined at the beginning of the project. Each objective has been addressed through careful design and implementation, resulting in a system that is functional, efficient, and user-friendly.

10.2.1 To Develop an Application That Can Monitor Inventory Management

The primary objective of developing an application capable of monitoring inventory management has been successfully achieved. The system provides users with an intuitive platform to add, update, and track inventory items efficiently. By automating inventory processes and providing structured management tools, the system significantly reduces errors commonly associated with manual tracking. Features such as notifications for low-stock items and automated alerts ensure that inventory levels are maintained effectively, supporting proactive decision-making and enhancing overall operational efficiency.

10.2.2 To Generate Professional Inventory Reports

Another important objective achieved by the system is the generation of professional inventory reports. The system allows users and administrators to create comprehensive reports in both PDF and Excel formats. These reports detail stock-in, stock-out, and low-stock items, providing valuable insights for management. By presenting inventory data in an organized and professional manner, the system supports informed decision-making, simplifies record-keeping, and improves accountability in inventory management processes.

10.2.3 To Provide Real-Time Visibility of Inventory Data to Users and Management

proof??

The third objective, providing real-time visibility of inventory data, has been successfully accomplished. Inventory data is updated instantly across the system, allowing both users and management to access current information at any time. This real-time visibility promotes transparency, enables timely decision-making, and enhances operational control. By offering immediate access to accurate data, the system ensures that all stakeholders are well-informed, facilitating efficient inventory management and effective communication across the organization.

10.3 Constraints and Limitation

Despite the successful implementation of the system, several constraints and limitations were encountered during development. One of the main limitations is the absence of AI-based predictive capabilities, which could enhance stock forecasting and inventory optimization. Additionally, the system relies on a stable internet connection to maintain real-time updates, which can pose challenges in environments with unreliable connectivity.

Another limitation is that certain operations, such as inventory updates, still require manual input, which may introduce human error. Although the system is designed to automate most processes, complete automation has not yet been achieved. Furthermore, hardware integration, such as IoT sensors, is not currently implemented. This restricts the system's ability to track physical inventory automatically, limiting its efficiency in certain scenarios. Finally, while the system is suitable for small and medium enterprises, it may require further optimization to handle very large-scale inventory operations. These constraints highlight areas for improvement while acknowledging the system's current capabilities.

10.4 Future Works and Recommendation

Although the Smart Inventory and Tracking System has successfully achieved its objectives, there are several opportunities to enhance its functionality and efficiency in the future. These improvements focus on integrating advanced technologies to optimize inventory management and provide more insightful analytic for users and administrators.

10.4.1 AI-Based Demand Forecasting

One potential future enhancement is the implementation of AI-based demand forecasting. By utilizing artificial intelligence, the system can analyze historical sales and usage patterns to predict future stock requirements accurately. This predictive capability would allow the system to recommend optimal reorder quantities, helping to prevent both stockouts and overstock situations. Integrating AI into inventory management not only increases operational efficiency but also supports strategic planning, enabling management to make data-driven decisions with greater confidence.

10.4.2 Cloud-Based Analytic Dashboards

Another recommended improvement is the development of cloud-based analytic dashboards. By storing and processing all inventory data in the cloud, the system would enable administrators to access real-time reports, charts, and trends from anywhere at any time. Cloud-based dashboards enhance transparency and accessibility, allowing management to monitor inventory performance continuously and respond swiftly to changes in stock levels. This approach also facilitates scalability, ensuring the system can handle increasing volumes of inventory data as the organization grows.

10.4.3 AI-powered camera integration

A further future development is the integration of AI-powered camera technology to automate inventory tracking. Instead of relying solely on manual input or physical sensors, AI-enabled cameras can detect, recognize, and count inventory items in real time. Through computer vision, the system would be able to identify stock movement, detect missing or misplaced items, and update inventory records automatically. This advancement would significantly reduce human error, enhance data accuracy, and provide a more efficient and seamless method of monitoring stock. In addition, AI-powered visual analytic can generate deeper insights into usage patterns, stock behaviour, and potential irregularities, empowering management with more informed decision-making capabilities.

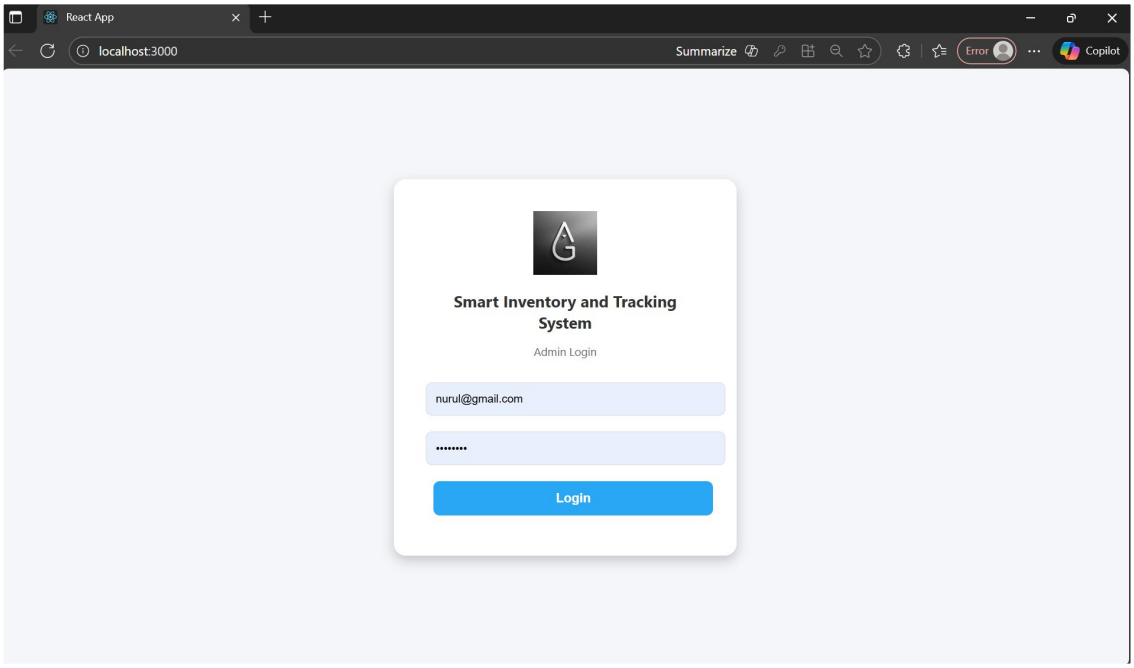
10.5 Conclusion

The Smart Inventory and Tracking System successfully achieves its primary objectives, providing a reliable, real-time platform for inventory monitoring, reporting, and management. While certain limitations exist, the system demonstrates the potential to significantly improve inventory control for SMEs. Future enhancements, including AI-based forecasting, cloud-based dashboards, and IoT integration, will further strengthen its functionality and scalability, positioning the system as a comprehensive solution for modern inventory management.

Appendix A – User Manual

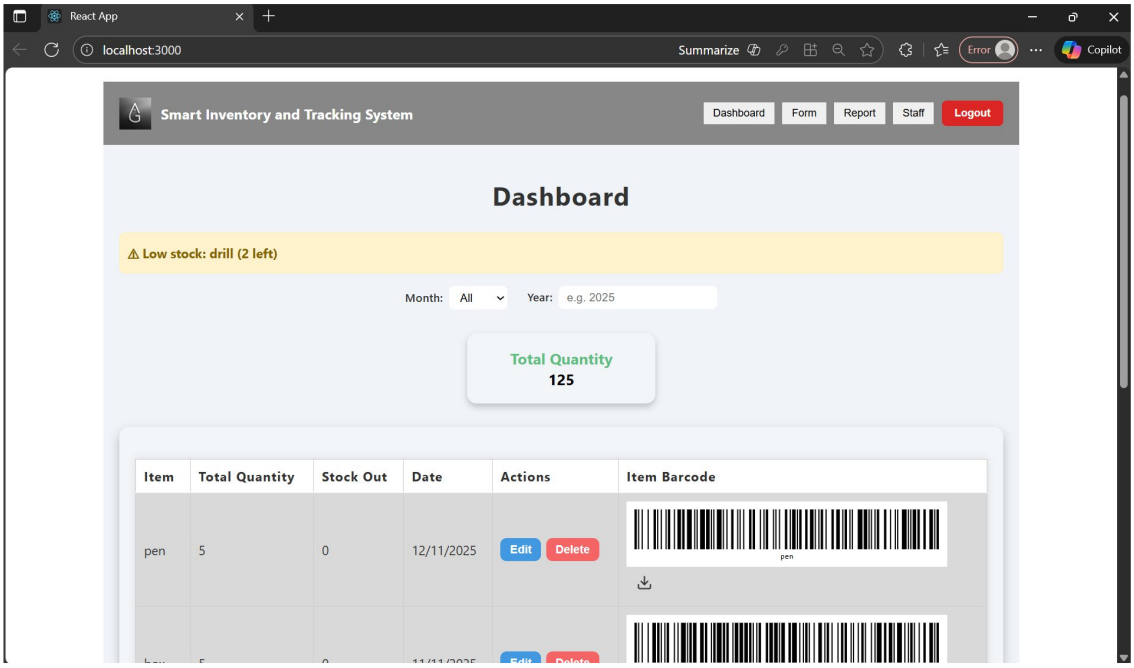
1.0 ADMIN

a. Login



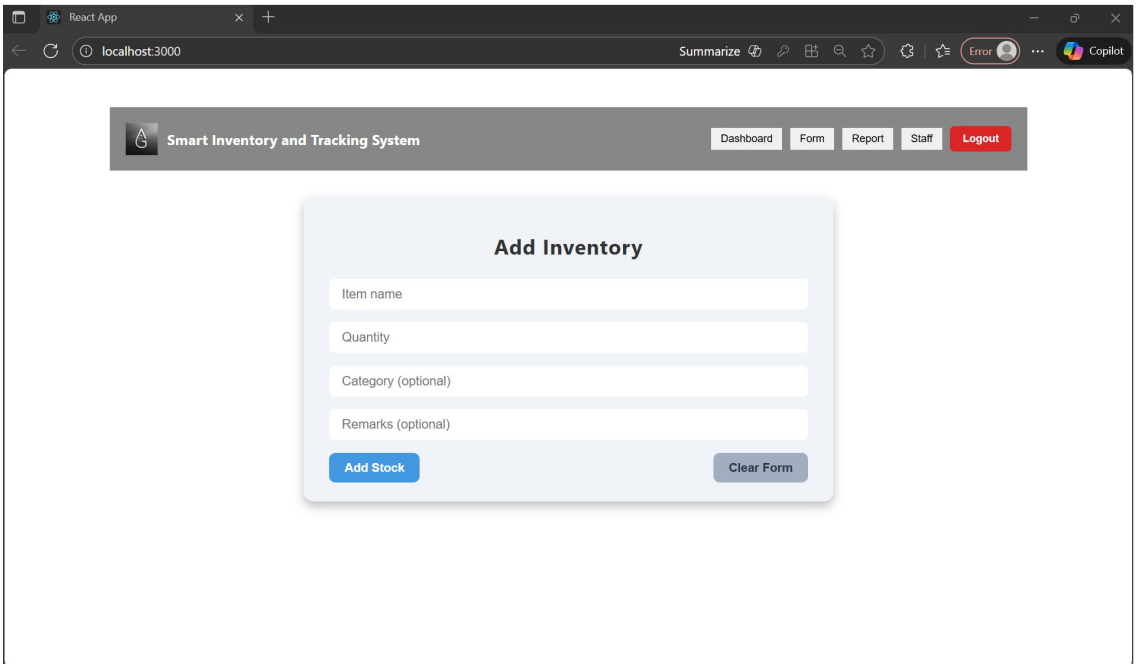
- Admin need to login by registered email and password.
- Click the login button for the login page to start authenticating its user

b. Dashboard



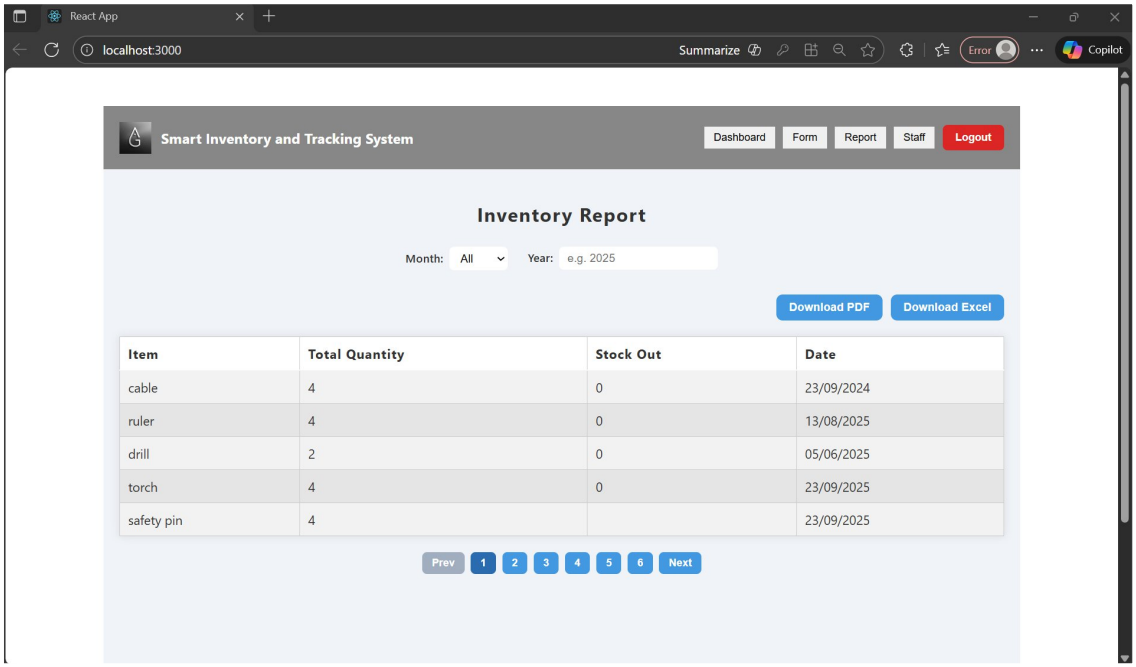
- After login, it will direct to Dashboard page and here admin can view the inventory list.
- Admin can edit the quantity and delete the item. Admin can download and print the barcode for item by clicking the download icon.

c. Form



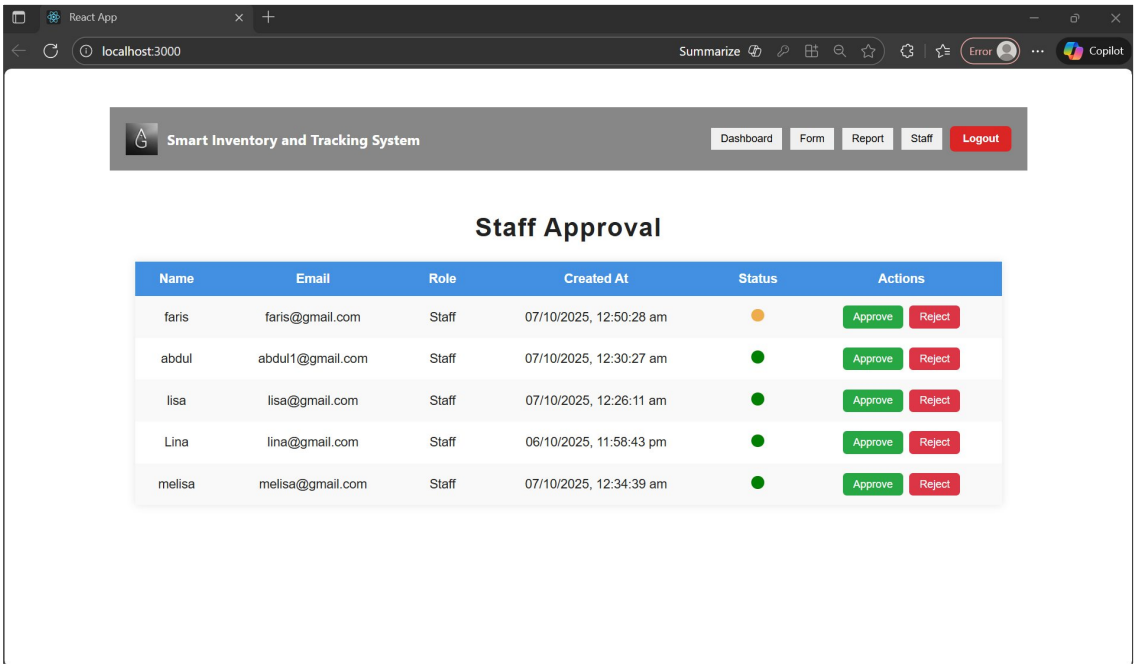
- Admin can add new inventory by fill the name and quantity then just click the Add Stock button.

d. Report



- Admin can generate report in PDF or Excel by click the Download button and admin also can filter by month or year.

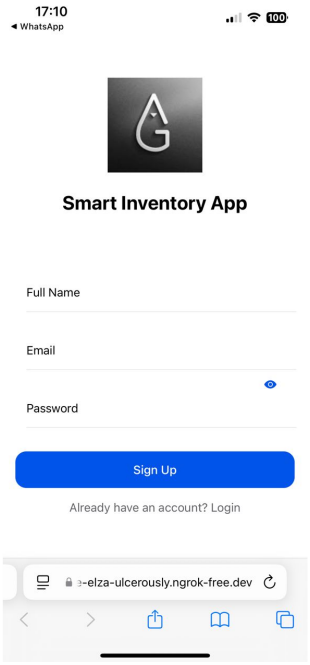
e. Staff Approval



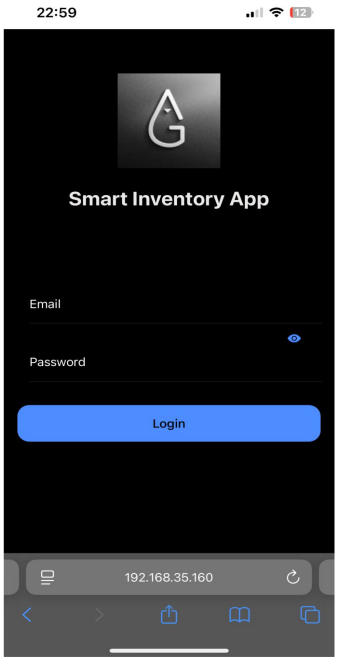
- Admin need to approve new staff for the staff can access to the system. Status will be yellow for pending, after approved it will turned to green.

2.0 Staff

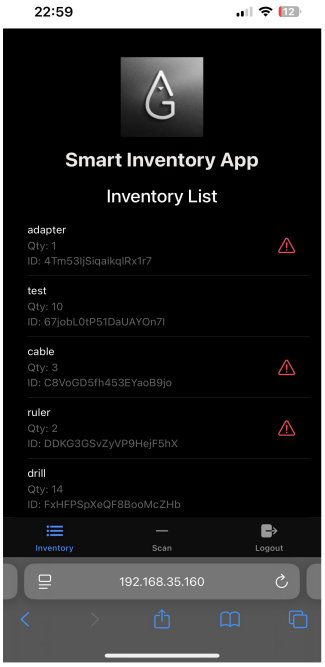
a. Sign Up



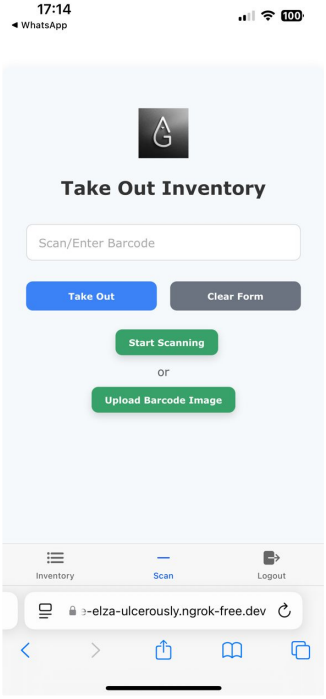
- Staff need to sign up their name, email and password but they can login after admin approved their account.



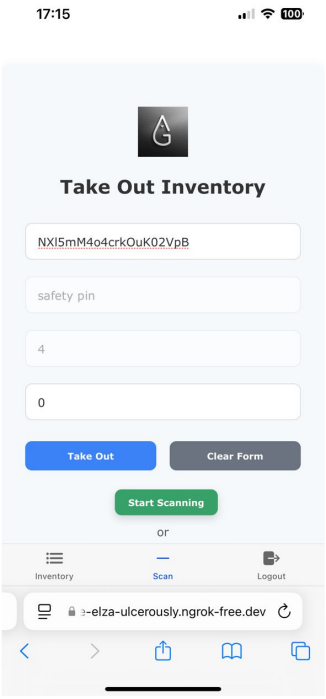
- Staff can proceed to log into the system by inserting registered email and password.



- Staff can view the inventory list and alert icon for stock below than 3.

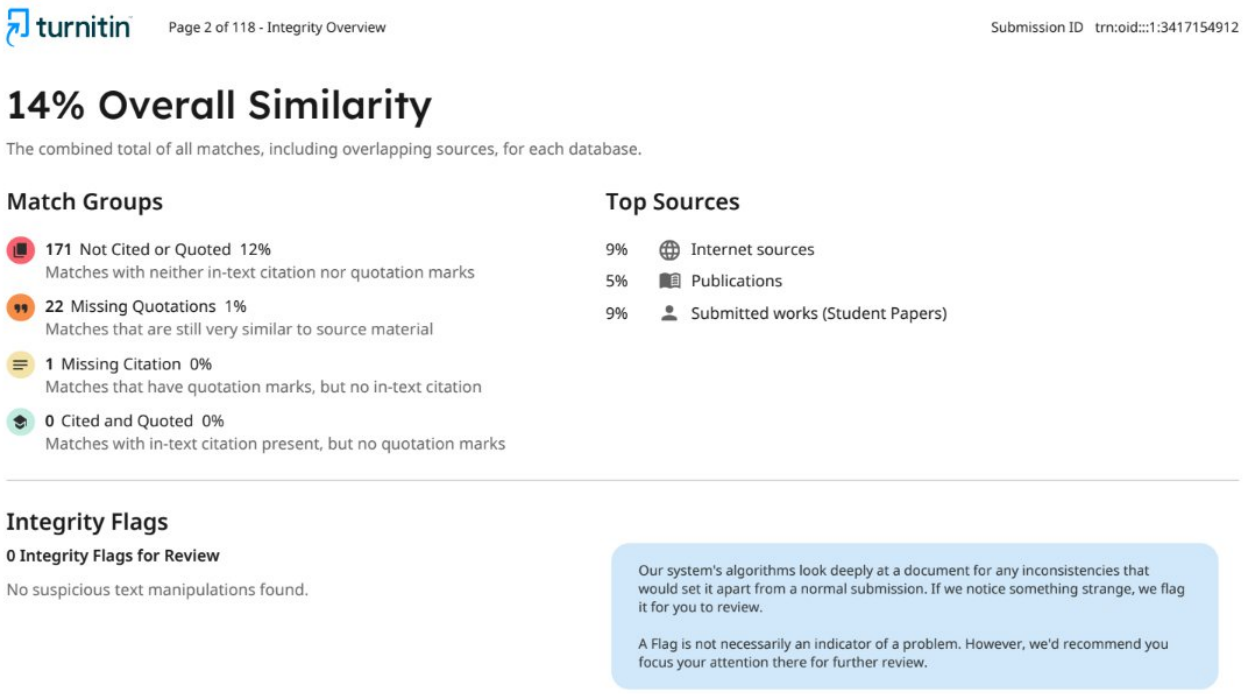


- Staff can reduce the stock throught enter item ID, scan the barcode by clicking Start Scanning or Upload image by clicking button Upload Barcode Image.



- Staff can view name, quantity and they only can update by reducing stock and just click the button Take Out or they can click Clear Form for mistaken item.

Appendix B - Turnitin Result



turnitin Page 2 of 118 - Integrity Overview Submission ID trn:oid:::1:3417154912

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **171** Not Cited or Quoted 12%
  Matches with neither in-text citation nor quotation marks
- **22** Missing Quotations 1%
  Matches that are still very similar to source material
- **1** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
5% Publications
9% Submitted works (Student Papers)

Integrity Flags

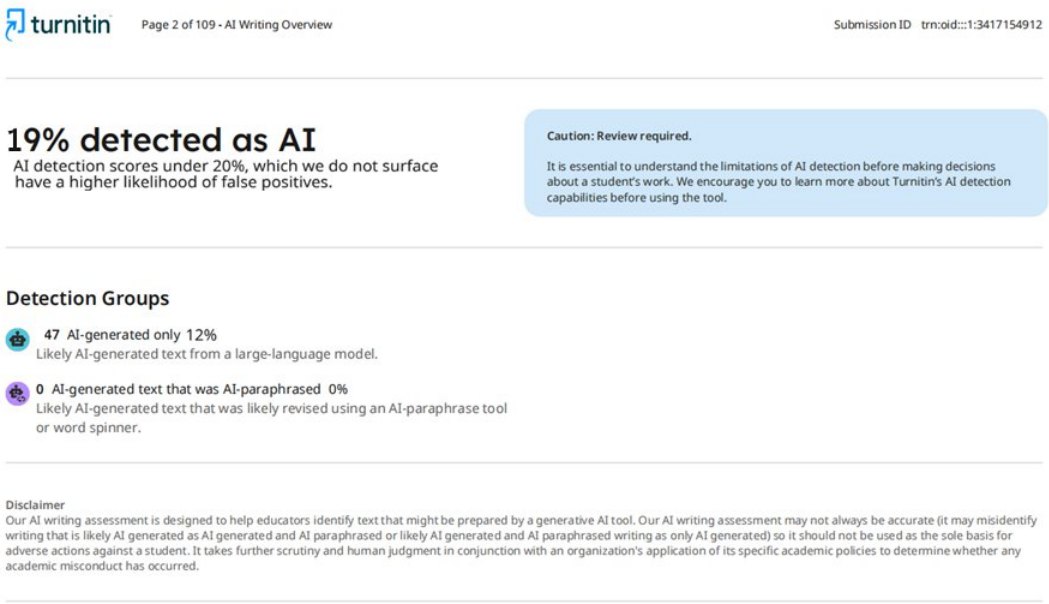
0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Appendix C - AI Result



turnitin Page 2 of 109 - AI Writing Overview Submission ID trn:oid:::1:3417154912

19% detected as AI

AI detection scores under 20%, which we do not surface have a higher likelihood of false positives.

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

47 AI-generated only 12%
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 (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) 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.

Appendix D - Log Book FYP 1

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

FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

BUSINESS COMPUTING PROJECT 1

(FYP4094)

LOG BOOK

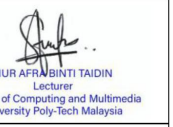
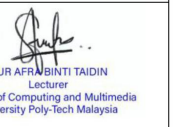
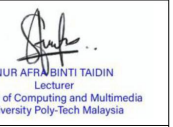
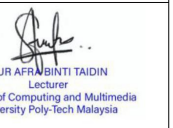
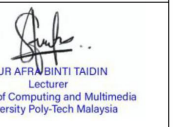
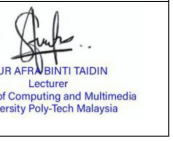
STUDENT'S NAME : NURULAIN BINTI ROSDI

ID NO. : AM2311015218

SUPERVISOR : PUAN AFRA BINTI TAIDIN

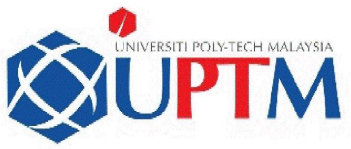
PROJECT TITLE : SMART INVENTORY AND TRACKING SYSTEM FOR SME CONTRACTOR

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

Date/ Week		Agenda	Next Agenda	Signature (Supervisor / Coordinator)
20/5/2025	1	Online class with Madam Nora and introduction about the course	Find potential Supervisors	
27/5/2025	2	Finalize topic and title with SV	Discuss project title with SV	
9/6/2025	3	Preparation project proposal	Submission of project proposal	
16/06/2025	4	Preparation of Chapter 1, 2 and 3	Complete Chapter 1,2,3	
23/06/2025	5	Conduct an interview and questionnaire to gain user requirements	Finish Chapter 4 and 5 for reporting	
30/6/2025	6	Prepare slide presentation FYP 1	FYP 1 Presentation with SV and Examiner	
3/7/2025	7	Presentation	Report submission	

Appendix E - Log Book FYP 2

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



FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

BUSINESS COMPUTING PROJECT 2

(FYP4105)

LOG BOOK

STUDENT'S NAME : NURULAIN BINTI ROSDI

ID NO. : AM2311015218

SUPERVISOR : PUAN AFRA BINTI TAIDIN

PROJECT TITLE : SMART INVENTORY AND TRACKING SYSTEM FOR SME CONTRACTOR

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

Date/ Week		Agenda	Next Agenda	Signature (Supervisor / Coordinator)
4/8/2025	1	Physical class with Madam Nora and lecture over the next chapter of FYP report	Prepare User Acceptance Testing (UAT) Questions for users	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
11/8/2025	2	Consultation with SV about UAT questions	Compile question in Microsoft Words and sahre in Padlet	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
18/8/2025	3	Prepare rough sketch and compile ideas regarding system's User interface	Prepare wireframe of staff and admin.	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
25/8/2025	4	Preparation of wireframe system	Prepare draft Chapter 6	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
1/9/2025	5	Conduct unit testing and integration testing	Prepare draft Chapter 7	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
8/9/2025	6	Prepare draft for Chapter 8	Conduct testing	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
15/9/2025	7	Continue system development	Report submission	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
22/9/2025	8	Continue system development	Continue writing Chapter 7 and 8	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
29/9/2025	9	Continue system development	Continue writing Chapter 7 and 8	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia

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

6/10/2025	10	Continue system development	Continue writing Chapter 7 and 8	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
13/10/2025	11	Continue system development	Continue writing Chapter 9 and 10	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
20/10/2025	12	Complete development to start conducting UAT	Continue writing Chapter 9 and 10	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
27/10/2025	13	Preparation slide for FYP 2 Presentation	FYP 2 presentation with examiner	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia
12/11/2025	14	Presentation FYP 2	Report submission	NUR AFRA BINTI TAIDIN Lecturer Faculty of Computing and Multimedia University Poly-Tech Malaysia

References

1. AltexSoft. (2023) *Functional and non-functional requirements: Specification and types*. Available at: https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements

2. Atlassian. (2024) *Project objectives: How to define and meet your project goals*. Available at: https://www.atlassian.com

3. Breakcold. (2023) *Target audience: What it is & how to find yours (with examples)*. Available at: https://www.breakcold.com/blog/target-audience

4. Christopher, M. (2016) *Logistics and supply chain management*. 5th edn. Harlow: Pearson Education.

5. Dwyer, C. (2023) *What is software deployment?* Available at: https://www.techopedia.com

6. Elsevier. (2022) *What is a problem statement and how do I write one?* Available at: https://www.elsevier.com

7. GeeksforGeeks. (2023) *Requirement gathering techniques*. Available at: https://www.geeksforgeeks.org

8. Gunasekaran, A., Subramanian, N. and Rahman, S. (2017) 'Supply chain resilience: Role of complexities and strategies', *International Journal of Production Research*, 55(22), pp. 6735–6753. doi:10.1080/00207543.2017.1349946.

9. Hayes, A. (2024) *Inventory management: What it is, how it works, and examples*. Available at: https://www.investopedia.com/terms/i/inventory-management.asp

10. Jain, A. (2024) *What is data analysis? Process, types, and methods*. Available at: https://www.simplilearn.com

11. Lutkevich, B. (2021) *Project scope*. TechTarget. Available at: https://www.techtarget.com/searchcio/definition/project-scope

12. Mourtzis, D., Doukas, M. and Psarommatis, F. (2016) 'Design and operation of a cyber-physical system for warehouse management in the context of Industry 4.0', *Procedia CIRP*, 52, pp. 299–304. doi:10.1016/j.procir.2016.07.056.

13. Muller, M. (2011) *Essentials of inventory management*. 2nd edn. New York: AMACOM.

14. Rai, A. and Sahoo, B.K. (2012) 'Performance evaluation of inventory control policies using simulation', *International Journal of Business and Management*, 7(2), pp. 143–156. doi:10.5539/ijbm.v7n2p143.

15. Rome, E. (2020) *Non-functional requirements: What you need to know*. ClearlyAgile. Available at: https://www.clearlyagileinc.com

16. Waller, M.A. and Fawcett, S.E. (2013) 'Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management', *Journal of Business Logistics*, 34(2), pp. 77–84. doi:10.1111/jbl.12010.

17. Boardmix.com. (2024). System Flowchart 101: Definition, Application, Benefits, Symbols and E. [online] Available at: https://boardmix.com/knowledge/system-flowchart/.

GITHUB

Admin: https://github.com/nurulainr00/inventory-web-admin
Staff: https://github.com/nurulainr00/staff-inventory

YOUTUBE

Demo video link: https://youtu.be/1jGoQI_iRlo