UNIVERSITI POLY-TECH MALAYSIA

SHAMELIN STAR E-REPORTING SYSTEM

MUHAMMAD AFIF NAQIUDDIN BIN OTHMAN

BACHELOR OF INFORMATION TECHNOLOGY (HONS) IN BUSINESS COMPUTING

UNIVERSITI POLY-TECH MALAYSIA

Faculty of Computing & Multimedia

SHAMELIN STAR E-REPORTING SYSTEM

MUHAMMAD AFIF NAQIUDDIN BIN OTHMAN

AM2307013973

FYP4105

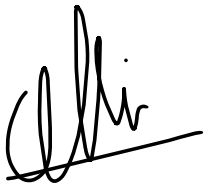
DECEMBER 2025

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: 17/11/2025

Office Stamp

Abstract

An e-Reporting System: Shamelin Star system is an application that developed by using web based to replace the current manual report method in Shamelin Star Condo. The general purpose of the project is to create a more effective form of communication, transparency and efficiency among residents and management by having one simple mobile platform that catalogues complaints from an entire community. Using this system, residents are able to quickly communicate maintenance or facility needs, add pictures for explanation and track condition of requests in real time.

The app includes multiple functionality such as Live Complaint Tracking System, and a AI based FAQ chatbot to give automated responses and structured reporting module to help management compile, filter and analyze complaint data efficiently. These facilities are also designed to reduce manual effort; delay in service delivery, and quality of services based on data-driven decision making.

Created with Android Studio, Java, and Firebase, the reserve sharing system is designed to focus on its UI, as well as reliability and security for user-friendly experience. The Shamelin Star e-Reporting System greatly improves operation efficiency, transparency level and finally boosts satisfaction rate of the residents in the condominium community.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, all praise to Allah SWT for giving me the strength, perseverance and patience to carry on this Final Year Project course. This would not have been possible without His blessings.

My special thank goes to my supervisor, Farah Farzana Binti Abdul Aziz who has provided med with supervising and guidance through the completion of this project. It is to them I owe for making me perceive the light at the end of this tunnel and being able to finish an almost perfect work.

I would also like to express my gratitude to Puan Raznida, the course coordinator for their guidance, prompt support and being able to handle the project processing carefully.

Special thanks to clients from Shamelin Star Condominium management for contributing valuable input to the research and feedback in action when performing requirement collection. Their collaboration and willingness were crucial to the achievement of this work.

Last but not least, thanks to my family who has unconditionally loved, prayed for and supported me since day 1 up until now. Thank you to my friends and classmates for your moral support, cooperation, motivation during the development phase.

Finally, I would also like to express my special gratitude and thanks to Majlis Amanah Rakyat (MARA) and University Poly-Tech of Malaysia (UPTM) for their support to do this task that worthwhile satisfying.

1 INTRODUCTION

1.1 Introduction

This chapter gives brief explanation of the Shamelin Star e-Reporting System. This chapter contains five sections. The first section presents the background of the project. The problem statements are defined in the second section. The project's goals are described in the third section. The fourth section describes the project on which we have been working. Finally, we explain the structure of this report in section five.

1.2 Project Background

The practical side of good facility management in housing estates is gaining even more importance in the world today as residents clamour for instant replies and better communication from the offices where management policies are being formulated. Conventional methods for processing complaints in condominiums, in example, paper-based reporting can result in procrastination of time, loss of reports and poor relationship between the residents and employees (Hassan et al., 2021). These complications result in a lower overall satisfaction of the residents and having to work harder to deliver good management services.

At present, occupants of Shamelin Star Condominium have to submit a complaint personally through the filling of written forms for their facility or maintenance requests. The process is inefficient and non-transparent, because once the complaints are submitted on behalf of residents, they have no way to monitor where their complaints go or what happens with them. Therefore there is frustration if problems don't get fixed in a timely manner. Meanwhile, the management cannot manage issues efficiently as it is difficult to get organized with aggregated complaints in a manual way which does not automatically create structured data or report. This condition causes difficulty to diagnose of a reoccurring anomaly, allocate resources appropriately and scheduling preventive maintenance (Ismail & Ahmad, 2022).

The Shamelin Star e-Reporting System is state of the art system that aims to solve the abovementioned problems by offering a webbased platform which facilitates recording and managing of complaints. The residents can lodge complaints using their smartphones instantly by attaching photos and descriptions for better understanding. Most important, however, Complaints Registry users will be able to take advantage of a Complaint Tracking tool which lets them have live views on Complaint Progress. For

example, it goes first submitted, then in-progress change and finally merged. This characteristic increases transparency and builds trust between tenants and the landlord (Nguyen & Lee, 2020).

Management office has a more structural and implementable reporting system in PIO. Staff is able to filter, download and analyze complaint data to see trends and patterns that often present recurring issues, by doing so, they are able to make more informed decisions on long-term plant maintenance planning. This evidence-based approach not only leads to efficient day-to-day functioning but also informs the adoption of preventive strategies that prevent repeated problems (Rahman et al., 2021).

Another component of the system is an automated AI-enabled FAQ chatbot. Residents can find answers to the frequently asked questions about complaints and condominium facilities received through the chatbot, which enables management staff to reduce answering repetitive questions and residents are able to get immediate answers for their doubts (Zhang & Chen, 2023).

In conclusion, the Shamelin Star e-Reporting System has tackled three main problems encountered in the existing system which includes inefficient manual reporting, problems to trace complaint and no structured report for management. The system is designed to increase communication, resident satisfaction and management workflow by digitizing the reporting process while adding real time tracking and supporting structured reporting.

1.3 Problem Statement

The section of this chapter describes the problem that Shamelin Star e-Reporting System attempts to solve. The problem statement is a crucial part because it clarifies what problems the project aims to solve. By recognizing these challenges the project can aim to develop a solution that accommodates both residents and management staff in efficient, transparent and data supported facility management. The problem statements detetected are:

1.3.1 Inefficient Manual Reporting

Currently, feedback to deal with the facility and maintenance for Shamelin Star Condominium is done using paper-based form. This manual activity is time consuming and further causes delays in which complaints can get lost in the system and without acknowledgement from the Management Office immediately. The same problems also have been identified in traditional property management systems where manual complaint processing slows down service response and results in inefficiency of administration (Hassan et al., 2021). The frustration of residents arising from snail pace treatment of complaints is a direct reflection of the problem management staff have in generating intercom reply. As

no digital system is implemented, communication between residents and staff remains inefficient and obsolete (Ismail & Ahmad, 2022).

1.3.2 Difficult to Track the Status of Complaints

Once a complaint is made, residents have no way to track that complaint's progress or resolution. Residents feel deceived and disappointed, not knowing when they will be able to take things in their own hands again. The lack of feedback loops and tracking mechanisms in facility reporting systems sometimes results in decreased trust of management, as well as resident satisfaction local perceptions (Nguyen & Lee, 2020). Management workers are also spending longer answering the same questions about complaint status. In the absence of a good tracking system and mechanism, both sides of service providers will eventually have communication gap which affects the overall experience on benefiting from the services (Rahman et al., 2021).

1.3.3 No Structured Reports for Management

There is no structured recording about the complaints and it is only maintained manually that doesn't allow management to do analysis on what are the frequent problems, if there are any high priority issues needs to be addressed or preventive maintenance steps need to be taken. This is a problem typically occurring in asset management facilities that are dependent on paper documentations, since such systems do not possess digital records for us to retrieve information and carry out analysis (Lim & Tan, 2020). Without this type of structured, filterable and downloadable reporting we are not able to look at trends within resident complaints and make trend-driven decisions. This limitation results in operational inefficiency as well as hinders the facility management improvement over time (Zhang & Chen, 2023).

1.4 Project Objectives

Project objectives are important in setting out the goals for the Shamelin Star e-Reporting System. All the objectives are aim to solve known problem in the problem statement.

1.4.1 To Digitalize Complaint Reporting

The primary purpose of the Project is to automate and bring about efficiency in registering complaints, which are presently registered through manual paper based process. Complaints can easily be made by residents from their smart phones and photos/descriptions can uploaded for further clarification. It also means that issue reports are communicated accurately, immediately recorded and available to management where they can't disappear or get lost. According to (Hassan et al., 2021), digitalization in property management increases efficiency by lessening the reliance on physical documents and reducing human errors. Likewise, (Ismail and Ahmad, 2022) reiterated that usage of mobile-based reporting systems will improve communication and create systematic trace accordingly on maintenance problems.

1.4.2 To Provide a Tracking Feature for Residents

Project is to process citizen complaints in an expeditious and organized manner, the application will feature a tracking functionality that displays how is each complaint progressing in real time to/with management. it is first submitted then, in progress, then completed. Through regular updates, it adds a level of transparency that minimizes followup calls and increases customer satisfaction relating to how the issue is being managed. Transparent systems for reporting have been shown to establish trust between residents and administration by giving users a sense of attention and response times (Nguyen & Lee, 2020). Furthermore, (Rahman et al., 2021) found that mobile applications with status tracking can also facilitate service accountability and resident participation.

1.4.3 To Generate Structured Reports for Management

Another goal of this project is to offer the management office reports that are well-organized, filterable and downloadable. These reports will enable the staff to look at patterns of complaints, trends in complaint types and patterns that might suggest where we need to put staff in preventive maintenance mode. The system reinforces long-term efficiency and enhances general maintenance life cycle, leveraging strategy based on data. (Lim and Tan, 2020) emphasized that structured reporting on maintenance data allows management teams to identify trends in maintenance, as well as allocate resources prudently. Similarly, (Zhang and Chen, 2023) submitted that ‘intelligent systems with analytics and reporting capabilities enable management to think in advance and preventive plan for bringing up a change and could enhance the quality of service delivery outcome'.

1.5 Scope and Target User

For this section is the extend for the Shamelin Star e-Reporting System, includes scope, product and the users.

1.5.1 Project Scope

In this project, Shamelin Star e-Reporting System, it is to improve the communication and management processes by moving from a manual paper base complaint report to electronic-based complaint reporting. Following the Waterfall methodology, this project adopts a well-organized development process; it consists of requirement analysis, system design, developing, testing and deploying. The system is built in Android Studio Java and Firebase used for data storage and synchronization where the web technologies such as HTML5, CSS3, and JavaScript are used to manage dashboard. This project specifically aims to enhance the transparency of work flow, ease of access to data and response

times for service trades through rigorous processes that are well-documented with reliability and usability that benefit both residents as well as management.

1.5.2 Product Scope

The mobile application shall have the following main modules: Complaint registration along with photo, real-time status of the complaint submitted, dashboard for back-office staff to assign and monitor the task assigned to Field Staff, Reporting and whose results can be filtered/downloaded. Comparable functionalities have proven to be efficient for responsiveness and accountability in smart home systems (Nguyen & Lee, 2020). What is more, another AI chatbot will be built to enable residents with quick answer via the frequently used questions on the reporting process and condominium facilities. Integrating AI assistants into digital systems increases user satisfaction by providing in-time guidance and minimizing the administrative load on users (Rahman et al., 2021). Ease of use, data security, and dependable performance will be key factors for the product so residents can quickly report concerns, and management can address issues with more efficiency.

1.5.3 Target User

The main user of this system is Shamelin Star Condominium Management Office. The end users are classified into two groups:

1. Residents will make complaints: via the mobile app upload evidence of complaint [photos] monitor status of complaint FAQ chatbot with which they can interact.
2. Managers, who will work with the system to accept, log and action complaints whilst creating a structured set of reports for strategic analysis and planning.

It is designed for use in residential complexes where communication between residents and management is essential. It is particularly suited for people who admire ease, transparency and to-the-minute problem resolution in communal amenities.

1.6 Overview of This Report

Chapter 1: Introduction. This chapter presents the summary of Shamelin Star e-Reporting System which consists of introduction, problem statement, objective, scope as well as the target user. It describes the rationale for the system, and what it is trying to achieve.

Chapter 2: Literature Review. In this chapter, current mobile CMSs (complaint management systems) and related technologies are reviewed. It enumerates the pros and cons of existing methods and discusses the application of digitalization and AI chatbots to facilitate effective communication, complaints record keeping, as well increased efficiency in management.

Chapter 3: Methodology. The methodology followed in e-Reporting System development is described in this chapter. It explains how the project was conducted, its steps and the technologies/environments used. It also details, how requirements are collected, implementing and integrating modules to meet the chatbot needs.

Chapter 4: Requirement. This chapter provides the system design, including architecture, UI mockups and database schema. Furthermore, it describes how the complaint submission, tracking, report generation and chatbot modules collaborate to satisfy the aim of the project.

Chapter 5: Analysis. The results of testing and evaluation are presented in this chapter. It examines the performance of the mobile app, the correctness and efficacy of chatbot implementation, and overall success in solving identified issues with this ambition. The results are examined, and potential future progress is discussed.

2 LITERATURE REVIEW

2.1 Introduction

This chapter conducts a systematic analysis and critique of existing studies on digital complaints management and facility reporting. A literature survey helps to combine the existing knowledge and identify a lack in service provision and technology implementation (Maulana & Legowo, 2024). This post will look at mobile apps and websites that can be leveraged to manage resident complaints, increase openness and improve the way that property managers communicate with residents.

2.2 Investigation

In the following we present the digital complaint reporting concept and show how mobile technology can support both communication and service quality in residential property management. This investigation determines how digitalization, real-time tracking and chatbot integration can address the problems arising in condominium management system in example, delayed response of helpdesk complaint, lack of transparency and manual process inefficiency.

2.2.1 Real-Time Tracking

Realtime tracking even enables citizens to track status of the complaint by seeing movement in the statistical graph. In regular methods of reporting the residents may find themselves stuck once complaints are submitted as it does not provide any visibility in regards to the resolution progress of an issue (Ismail & Ahmad, 2022). Real-time tracking resolves this by indicating complaint statuses like Submitted, In Progress, and Completed so occupants know their complaints are handled in a timely manner.

This feature is technically facilitated by database synchronization and push notification mechanisms. The system supports that, when a staff worker updates the status of the complaint, an automatic update occurs in the user's mobile with reference to backend server which is enabled by cloud-based synchronization (Nguyen & Lee, 2020). It's to ensure every user sees only the freshest data without needing a manual refresh.

Having this feature adds more transparency and builds trust between tenants and management. It also lessens the required amount of the follow-up phone calls, or office visits with a building management office, thereby increasing efficiency in complaints handling process (Rahman et al., 2021).

2.2.2 AI Chatbot FAQ

The AI chatbot works as an intelligent virtual assistant which allows residents to receive immediate answers to frequently asked questions related to both facilities and the complaint process. Chatbots are also being widely adopted in property management, because they offload the workload from administrative service personnel and can enhance responsiveness of service systems (Zhang & Chen, 2023).

The chatbot leverages the Natural Language Processing (NLP) capabilities to understand user questions and give accurate answers. It can help to answer simple questions like “How do I file a complaint?”,”How is my problem doing?” or, “Who could I call for a repair?”. Residents have peace of mind 24/7 using the app and AI, without waiting for staff replies.

From an implementation standpoint, the chatbot can be developed using Dialogflow, IBM Watson Assistant or Microsoft Bot Framework and integrated with the app through an API. This not only enables seamless communication between the chatbot and database, but also it aids in fetching complainant data without time lapse. AI chatbot offers not only better access but is the right step in to promote modern, tech savvy condominium management as well.

2.2.3 Structured Report

Well organized reporting is a vital characteristic which allows management personnel the ability to see, filter and download well-organized complaint data for decision authentication and long record carrying by maintenance. There is very little standardization to traditional manual reports that leaves a lot of room for human errors and it is difficult to spot common occurrences or track employees’ performance (Hassan et al., 2021).

The system also has a structured reporting module that will create reports for you by types of complaints, when they were submitted, time when completed, and staff responsible. The data are kept in a relational database that staff can query based on different criteria, for example issue type or response time (Nguyen & Lee, 2020).

Structured reporting facilitates data-based management which enables the decision makers to identify patterns and hot spots for preventive maintenance (Rahman et al., 2021). It also makes it easier to hold people accountable because leaders can easily watch how their offices, factories and call centers are responding to grievances. This will in turn make operations more proactive, strategic and measurable.

2.3 Related Works

For this section, related systems are reviewed to better help understand how it function, technologies it uses and all challenges for them. From this it helps to guide the final product for Shamelin Star e-Reporting System.

2.3.1 iNeighbour Community App

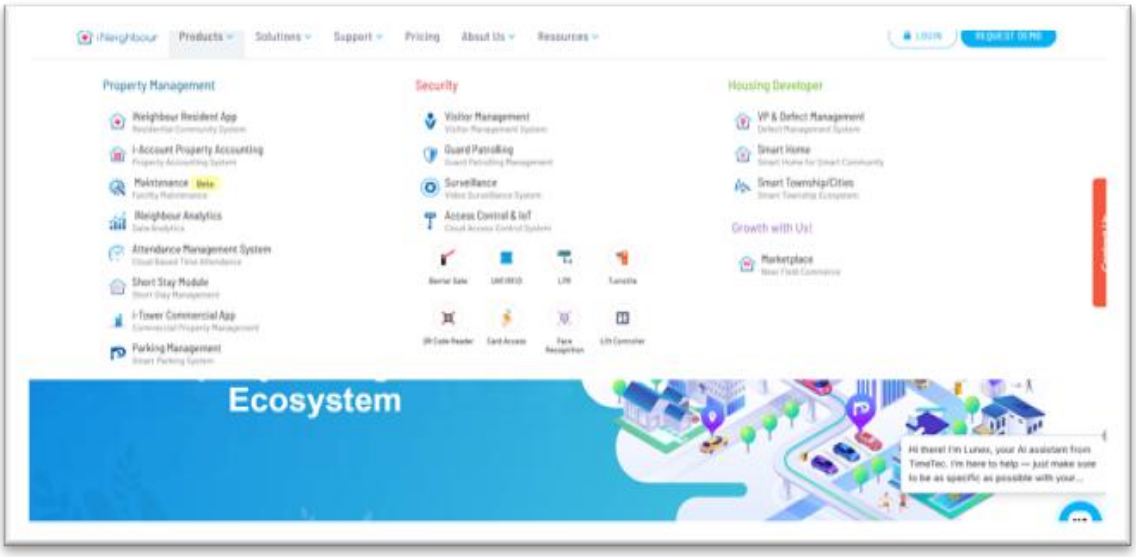


Figure 2.3.1: iNeighbour Community

iNeighbour is the most widely deployed residential community management system in Malaysia, currently used by condos, apartments and communities across the country. The software then offers individual modules ranging from visitor registration to facility booking all the way through announcements, billing and a complains module. It comes with a complaint form which the residents can use to file complaints with photos and descriptions, and management personnel can check case status within an admin dashboard panel. iNeighbour protects data by user authentication, access control and encrypted communication process, restricts unauthorized person from viewing or managing the complaint information.

One of the other advantages of iNeighbour is its very polished Mobile Apps to ensure both residents and management experience are sleek enough. It also gets connected to Smart community devices like QR access, smart intercoms, connecting an array of futuristic appliances for a futuristic living in apartment. But for those who just require a simple complaint-reporting system, iNeighbour's wide array of features can be quite daunting. Platform is also subscription-based which could be expensive for smaller Joint Management Bodies. Moreover, several users complain that response times are slow at peak hours leading to ineffective complaint follow-up.

2.3.2 Report & Run

Figure 2.3.2: Report and Run

Report & Run is a crowd sourced reporting app for Social ERP that allows the public to issue complaints on municipal issues; damaged buildings, garbage removal and safety risks. Residents can file reports by uploading photos, tagging locations and mentioning what the issue is. The system directly forwards the complaint to responsible agency, resulting in efficient disposal of community issues. Security features such as user registration, location validation and complaint verification for processing legitimate reports.

Straight & Simple is the greatest asset of Report & Run. The app is dedicated entirely to reporting, so unlike many others, it's easy and quick for residents looking for a fast way to file a complaint. It also facilitates geo-tagging that aids in pinpointing issues, enabling response teams to be more efficient. However, Report & Run does not offer extensive complaint tracking beyond a basic status update possibly detracting from transparency to users who yearn for real-time progress. Its also an app for public use, not dedicated to condominiums. So there's no management dashboard, statutory book or internal building communication tools essential for residential communities.

2.3.3 MYREDS

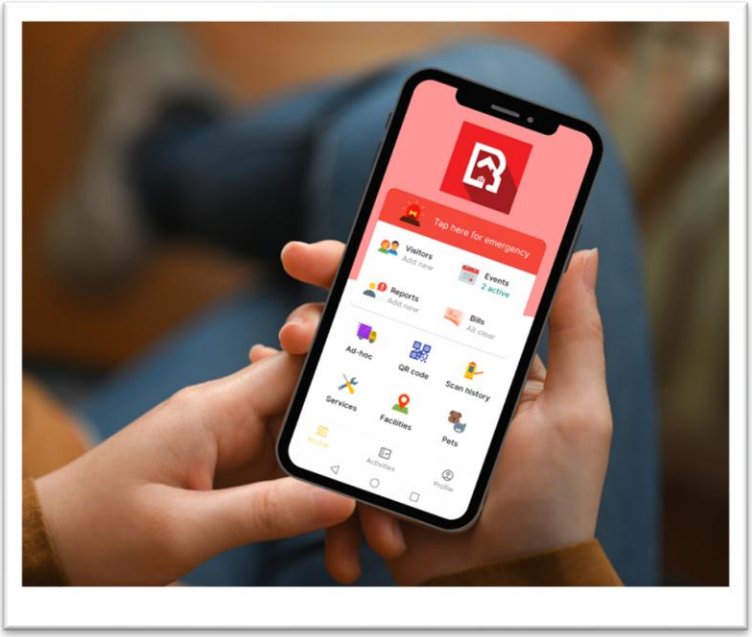


Figure 2.3.3: MYREDS

MYREDS (My Residential Easy Decision System) is a commonly used residential in check system for condo and gated community in Malaysia. It allows people to file grievances, let visitors in, read notices, and pay bills via web and mobile. The report module enables users to submit reports with text and attached files, and management can view, update, and close cases from the admin control panel. MYREDS adopts a account-based login, rolebased access control and communications mechanisms to guarantee that sensitive data are only accessible via authorized roles.

The integrated system MYREDS has built enhancing the residential ecosystem of multiple resident services in one place is a significant benefit to management in optimizing and facilitating their day-to-day business activities. Centralized processes will save time on manual work and increase transparency between tenant and management office. However, MYREDS is not a complaint-based data platform since its reporting capabilities are rather restricted compared to dedicated systems. Some condos also emphasize budget issues, given the subscription amounts may not be appropriate for smaller communities. The system focuses more on overall residential management and is not a comprehensive complaint-tracking solution with real-time reporting or deeper reporting capabilities.

2.4 Comparison

Table 2.4.1: Comparison of Existing Project

Criteria	iNeighbour	Report & Run	MYREDS	Shamelin Star e-Reporting System
Target User Group	Residential communities	General public reporting municipal issues	Residential communities	Residential communities
Complaint Tracking	Provides status tracking with updates in app	Limited tracking with basic status	Provides basic tracking	Provides real-time tracking update
Reporting & Analytics	Supports structured reports and insights	Minimal reporting, focused on issue submission	Limited reporting functions	Support reporting with image
Mobile App Accessibility	Well-developed mobile app	Simple mobile app with limited features	Mobile and web access	Mobile app
Function Complexity	Many modules	Single-purpose complaint reporting	Moderate complexity	Easy to use reporting and AI Chatbot
Suitability for Condominium Use	Highly suitable offers full ecosystem	Not suitable designed for public/government	Suitable provides core residential functions	Suitable for condominium uses for everyday task

2.5 Discussion

Review of iNeighbour, Report &Run and MYREDS Strengths and Weaknesses The findings from the review on existing software CitySense systems inform development of the Shamelin Star E-Reporting System. Both of the systems thus give useful knowledge to us on how digital complaint platforms work, but they are not completely fulfilling the requirement for Shamelin Star Condominium because currently it lacks simplicity, transparency and structure reporting.

iNeighbour showcases the advantages of a complete residential ecosystem such as one with visitor management, announcement broadcasts and electronic payments. Its complaint component is well-structured with status tracking; this characteristic is compatible with the requirement of transparency in the Shamelin Star e-Reporting System. But the platform is complex, and its wide array of features may confuse residents who simply want an easy way to report complaints and make sure they are heard. This is reason

enough to develop the Shamelin Star system as an ease-of-use enriched complaint-based rather a heavy all-in-one management community application.

Report & Run is exactly the opposite - it excels through its simplicity. The system enables users to lodge complaints quickly, often with location tags and photos. Its minimalist workflow makes it suitable for quick reports and it will be a handy guide in building the user flow for Shamelin Star's residents. However, the system does not have specific features that aim at internal condominium management for example, to allocate tasks within a condominium, or middles for structured complaints reports and private communication. This constraint highlights the necessity of dedicated dashboard for Shamelin Star e-Reporting System to facilitate management in categorizing, filtering and comparative analysis of complaints for maintenance planning.

MYREDS includes instruments for households-advice bulletin, billing histories and complaints submission. But its complaints module provides only rudimentary monitoring and little or no analytic reporting. This gap highlights a need to develop a more advanced structured reporting system module in the Shamelin Star system so that management can produce reports, downloads reports and thirdly recognises common problems and thus provide better strategies for long-term maintenance. Although MYREDS is mobile friendly, its interface doesn't feel as up-to-date or optimal for quickly posting complaints as newer platforms that provide even more evidence of the need to develop a more user-friendly and responsive mobile application.

On the whole, this evaluation comparison demonstrates that there are no systems which meet all of Shamelin Star Condominium's specific needs although other systems serve as useful benchmarks. iNeighbour has effective tracking features, Report & Run shows the benefit of simplicity and MYREDS is a case in point for core residential tools. But they don't have structured reporting, get too fancy or just aren't built for dealing with condo-level issue management. These results support the adoption of a specialized e-Reporting system with real-time tracking, web-based complaint submission process, the use of dashboards for management decision making reports and an AI-driven FAQ chatbot to enhance communication and service efficiency between residents and management.

2.6 Conclusion

A comprehensive overview of current digital reporting system and the significant technology elements has been described in this chapter, which is related to Shamelin Star e-Reporting System development. From the discussion on iNeighbour, Report & Run and MYREDS, it is clear that existing platforms are able to provide valuable features like reporting a complaint, very basic tracking function and communication or announcement between members but fails to present an integrated in terms of simplicity, transparency and structured reporting cater for facility management at condominium level.

Research into technology such as live complaint tracking, AI chatbot support, and structured reporting likewise emphasizes the value of increased communication efficiency, improved user experience and data-driven decision-making. This understanding, coupled with the weaknesses and strengths of existing systems answers for a well-developed digital solution that meets the real needs of residents and management.

In total, the results of this chapter deliver a solid basis for the construction and development of method adapted to the new system. The Shamelin Star e-Reporting System aspires to provide a more transparent, responsive and efficient means of condominium facility & maintenance issue reporting by tackling the deficiencies found in current platforms and incorporating critical features enabled by ongoing technological advancement.

3 METHODOLOGY

3.1 Introduction

A methodology for software development constitutes a base on which the design of a system, its realization and its evaluation are directed. Selection of methodology affects quality, schedule and project success. The Waterfall model is a sequential process and is ideal for projects in which the requirements are well understood and remain relatively stable (Pressman & Maxim, 2020). This stands in marked contrast to Agile, which is all about flexibility and repetition of the development process allowing quickly adjustment based on user feedback (Beck et al., 2001). Prototyping model is based on the system by developing early versions in order to improve requirements with demonstration of user needs (Sommerville, 2016). There are unique strengths regarding the IT companies analyzed, with inherent weaknesses in each approach and decisions on them have to be made depending on project (Balaji & Murugaiyan, 2012).

Chapter the following chapter presents the selected methodology for Shamelin Star e-Report System by concentrating on Waterfall methodology. The Waterfall model: This methodology is characterized by a linear sequence, best for projects that have highly set requirements and end goals. The process involves several steps like requirements gathering, design of system, development of system, testing of system and deploying as well maintaining the products. All phases shall be carried out in their entirety before commencement of the following phase, ensuring clear & negligible re-work.

3.2 Waterfall Methodology

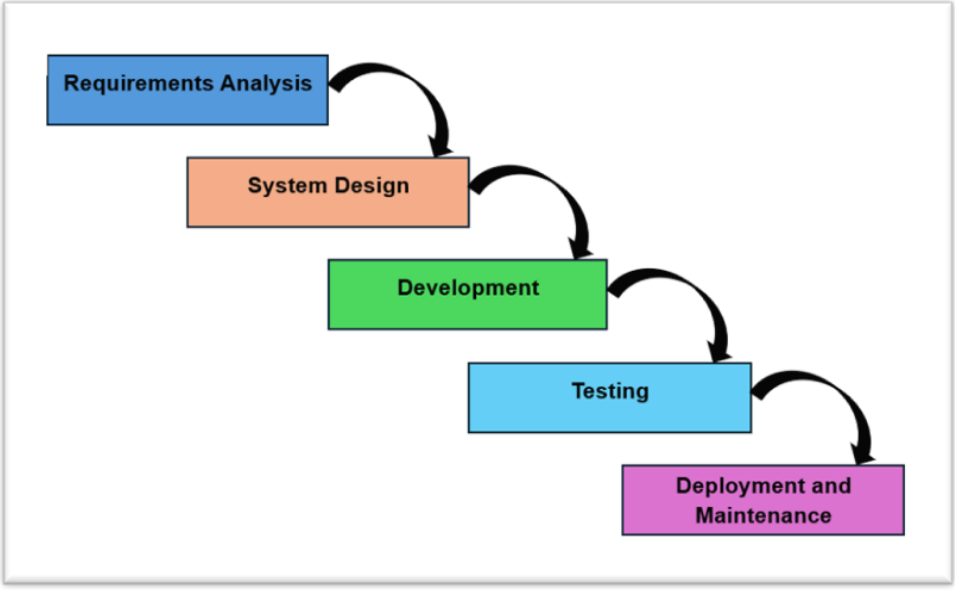


Figure 3.2.1: Waterfall Methodology Diagram (Afif, 2025)

The Waterfall model was chosen for this project as the objectives and system features including complaint registration, tracking status, sending structured reports, and chatbot integration are clear and stable. Using the screwdriver framework, this can be achieved by making sure every (major) step in the development process gets documented and reviewed before going on to the next. It offers an obvious definition of a progressive and quality guarantee, this methodology is appropriate for projects with clear requirements and well-defined objective (Pressman & Maxim, 2020).

This process is straightforward and sequential, meaning that each step must be fully completed before the next step can start. In contrast to iterative methods such as Agile, the Waterfall method focuses on significant document deliverables and well-defined features with an orderly sequence of routine from design until deployment (Sommerville 2016). This strategy is particularly effective when demands are known and do not generally change during project development (Balaji & Murugaiyan, 2012).

Under the Shamelin Star e-Reporting System, Waterfall model is suitable as system requirements to be fulfil are digitalization of complaint submission, also enable real-time monitoring, structured reporting will be generated and AI based chatbot will be embedded. Each step builds on the previous stage, and it is only when requirements are gathered before design can be made, or when designs are done prior to development, or when testing starts after full development (Pressman & Maxim, 2020).

In addition, the model can also reduce project risks by identifying possible mistakes at an early stage and avoiding expensive rework for any rule violations during downstream stages (Sommerville 2016). At the same time, however, its hierarchical documentation and sequential flow can be more easily related to academic research standards that demand verification, traceability and periodicity of at least some accomplishments. In summary, the discipline, structure and predictability of the Waterfall methodology provides an effective framework for delivery of a reliable and user-centric CMS that will benefit both residents and condominium management personnel.

3.3 Phases in Waterfall Methodology

This section will present an overview of the phases of the Waterfall method. Which are requirement analysis, system design, development, testing, and deployment & maintenance. All the phases of the project will be described in details to demonstrate how the system evolves incrementally.

3.3.1 Requirements Analysis Phase

Requirements Analysis Phase of Project Shamelin Star e-Reporting System concentrates on soliciting and documenting user as well as system requirements in details. This includes specifying issues like manual complaint filing that generates a time-consuming process, the opacity of the way complaints are followed up on and lack report structure. Features include digital complaint filing, real-time tracking of complaints, reports in a structured manner and an integrated chatbot. The non-functional requirement included usability, performance and data security.

In these field studies, data will be gathered from user perception and interviews with condominium inhabitants and administration to comply the system with the user profile as well as administrative requirements. Balaji and Murugaiyan (2012) also asserted that having well-stated requirements from the beginning lays a foundation for everything else, where less confusion leads to a reduction of wasted time correcting errors as development continues.

3.3.2 System Design Phase

System Design Phase -The collected requirements are translated into exact specifications. This involves the design of system architecture (user interface, management interface), defining the database model (complaints) and planning for integration of he AI-powered FAQ chatbot.

The user interface (UI) will be built for ease of use and clarity by residents, while the back end system is designed for management to sort, track and report/filter queries with ease. According to Pressman and Maxim (2020), there are clear benefits to a well-organized design phase that offers a strong design pattern for developers, better system maintainability, ease of use by the users. At the end of this stage, full system specifications and design notes would be complete and ready for implementation.

3.3.3 Development Phase

During the Development Phase, Shamelin Star e-Reporting System will be developed according to the approved design specifications. This phase consists of building the mobile app interface, adding complaint submission and tracking module as well as integrating reporting dashboard for management and enable chatbot support for AI based help.

All code will be release quality code modular and maintainable. As Waterfall is a sequential approach development commences only after design phase completion, it prohibits alteration of the design during the development process and maintain consistency throughout the system (Balaji & Murugaiyan, 2012). At this point, developers will also need to ensure that the database and server options work in tandem to facilitate live operations.

3.3.4 Testing Phase

For the Testing Phase, Shamelin Star e-Reporting System will be fully tested based on requirement specification and it is function accurately as design. Various testing techniques will be utilized, including unit testing to test individual functions, integration for module interoperability, and system testing overall functionality.

Real-world tests are planned to be run that test complaint submission, tracking, reporting and chatbot responses. The system will be usability tested with representatives from actual residents and building services staff to ensure ease of use in water detection and system reliability. (Sommerville, 2016) explains that comprehensive testing makes the errors visible early, improves software quality and verifies whether system works perfectly which becomes effective after it goes live.

3.3.5 Deployment and Maintenance Phase

Implementation of Shamelin Star e-Reporting System will take place during Deployment and Maintenance Phase for the usage by residents and management of condominium. The implementation includes downloading the mobile-app, setting up databases and staff training to ensure management can lookiously use this system.

After the app is live, comes in maintenance by updating the system regularly to fix bugs and add features. The AI chatbot will also grow to meet complex user queries, with feedback from residents and management shaping future system upgrades. Also maintenance aims to maintain the system to keep it up and running, secure, functional, and adjusted with users' evolving requirements (Pressman & Maxim 2020).

3.4 Conclusion

Finally, the Waterfall method gives a systematic and structured approach for developing the Shamelin Star e-Reporting System that guarantees each phase is completed thoroughly before moving to next phase. This waterfall tactic mitigates ambiguity by specifying all requirements upfront in the project and then translating that into detailed designs that serve as blueprints for development, testing, and deployment. (Pressman and Maxim, 2020) hold that this architecture supports coherent, consistent and predictable processes of the software development lifecycle.

The Waterfall approach also relies heavily on documentation at each step something that is crucial for projects with a broad number of stakeholders, ranging from residents to management staff. The thorough documentation of the system would aid developers and administrators in knowing what they are working with and help prevent any misunderstandings or additional scope (Sommerville, 2016). In addition, linear processes are used, enabling early detection of issues, thereby reducing risk of expensive rework in the later stages of development.

The Waterfall model is not as flexible as iterative methods, such as Agile, and should be considered for projects like the Shamelin Star e-Reporting System which has clear and specific objectives including digital complaints submission forum, real-time tracking function and structured reporting at the beginning. By taking into consideration proper planning, refinement of the design issues and appropriate testing outcomes with competitive, presentable and error free product that meet the objectives of achieving to enhance communication and operational performance for condominium management (Balaji & Murugaiyan, 2012). Ultimately, the Waterfall model delivers to the project a robust & effective reporting system which adds value for both residents in terms of satisfaction and for administration.

4 REQUIREMENTS

4.1 Introduction

This chapter described the requirement gathering process performed to elicit critical functional and non-functional requirements for the Shamelin Star e-Reporting System. This stage is designed to verify if the system truly meets resident and condominium management requirements. System requirements are specified in order to develop system design of software and hardware (Sommerville, 2016). It was challenging and resourceintensive, with direct stakeholder engagements to include meetings with condominium management team, interviews, and surveys administered to residents in order to obtain feedback on the existing problems of manual complaint reporting system. By doing so the project team can identify ‘what’ are essential features like submitting a complaint, real-time tracking, structured reporting and AI chatbot for FAQ.

4.2 Data Gathering Techniques

Collecting information is an important stage as it ensures development of a system that corresponds to user needs, and corporate objectives. The third stage is about gathering stakeholders' needs and expectations, problems they encounter and preferred functionalities of a system. As (Sommerville, 2016) points out, the success of a software project is largely determined by the manner in which requirements are elicited, analyzed and documented. Depending on the scope and objectives of the project a range of data collection methods can be used, such as interviews, questionnaires, observation and document analysis.

4.2.1 Interview

The interview modality was applied in order to gather in-depth qualitative data directly from relevant system users and controllers. Selected management office, technical support staff, and administrative staff were interviewed. The selected participants were directly involved in managing the facility operations and user feedback, thus representing a source of knowledge about workflow constraints and process improvement requirements.

Both interview sessions were semi-structured, with some scope to deviate from the predetermined list of questions and probe into anything relevant. It has been suggested by (Myers and Newman, 2007) that such a method can uncover deeper issues, expectations and preferences that may not otherwise surface with a structured investigation. Interview questions were used to gain an understanding of the present day processes, issues in dealing with reports or requests and expectations for a more streamlined, open process. The available response data offered new insights leading to the development of both functional and non-functional requirements.

4.2.2 Online Questionaire

A similar questionnaire online was sent to a wider pool of potential system users to collect quantitative evaluations on user expectations in a larger population. The respondents primarily are residents. The resident was selected based on the premise that it is an end-user whose life can be affected by a better digital reporting and communication system.

The survey was made with Google Forms and posted on the management email and social media platforms to be easily reached. This included the opportunity for additional feedback using multiple choice questions. As claimed by (Brace, 2018), online questionnaires are a quick and easy way to capture data from large populations although it protects confidentiality and honesty of responses. The information collected has been useful to identify common problems, measure user satisfaction with existing processes and confirm the benefit of system features such as complaint tracking, automatic updates and easy/accessible mobile options.

4.3 Functional Requirement

Functional requirements are the items that Shamelin Star e-Reporting System must contain or accomplish in order to satisfy user needs (Altexsoft, 2023). These "requirements" define how the system would work in various scenarios of use to enable capabilities like residents complain submission, photo upload, tracking of submitted reports states, communicate with management office. Communicating these needs properly to the development team and stakeholders is paramount for producing a system that serves its purpose well.

Table 4.3.1: Functional requirement resident

Function	Description
User Registration and Authentication	Residents must be able to register, log in, and manage their profiles securely to access the complaint system.
Complaint Submission Module	Users can submit complaints by entering descriptions, selecting categories, and uploading photos for evidence.
Real-Time Complaint Tracking	The system must display complaint statuses, such as "Submitted," "In Progress," and "Completed," enabling residents to monitor progress transparently.

AI Chatbot for FAQs	The chatbot should provide automated responses to common resident questions to reduce staff workload.

Table 4.3.2: Functional requirement administration

Function	Description
Management Dashboard	The management team can view, update, assign, and resolve complaints using a centralized dashboard.
Structured Reporting	The system should generate filterable and downloadable reports summarizing complaint patterns and staff response times.

4.4 Non-Function Requirement

A non functional requirement is a criteria that acts as a guide to the operation of a system rather it specifies what the system will do. These requirements define the levels that are: the performance, usability, reliability, maintainability and security of this system. Non-functional requirements are typically considered to specify the scope and boundaries of the system by software engineers and analysts. They make sure that the solution they deliver actually works in real world circumstances and matches organizational objectives, allowing for technical realities (Beaton, 2024).

Table 4.4.1: Non-Functional requirement

Requirement	Description
Performance	The system should process complaint submissions and updates within seconds to maintain responsiveness.
Usability	The interface should be user-friendly, simple, and accessible for residents of all ages.
Maintainability	Future updates, such as new features or bug fixes, should be easily integrated without disrupting service.
Reliability	The mobile application and server must ensure high availability and consistent operation to avoid downtime.
Security	All complaint data and user credentials must be encrypted to prevent unauthorized access.

4.5 System Requirement

Rouse (2019) notes that system requirements are the hardware and software resources required for an application to run efficiently. This will allow for smooth operation across devices, as well as stability and compatibility. To the Shamelin Star e-Reporting System, clear system requirements are crucial in order to avoid installation problems, performance bottlenecks or compatibility issues. Such requirements are usually divided into functional, data, quality and constraint so that the system is accessible and useful functionalities for both residents and management staff can be established.

4.5.1 Software Requirements

Software requirement will describe the required applications, platform, software and tools to be use in developing, deploying and executing of Shamelin Star e-Reporting System. All of these characteristics are necessary so that the system works optimally, supports all features and is compatible with currently known required technology to be used.

1. Android Studio

Figure 4.5.1: Android Studio (Android Developers, 2023).

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA.On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as: A flexible Gradle-based build system.Build variants and multiple APK generation.Code templates to help you build common app featuresTesting tools and frameworksLint tools to catch performance, usability, version compatibility, and other problems C++ and NDK supportBuilt-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App EngineLike getting the wind at your back

To get started with Android programming using Android Studio. You have powerful tools to manage your workflow in the form of built in emulators, build automation via Gradle and an array of performance profilers. This application allow Shamelin Star e-Reporting System to run smoothly on Android Tablets through the easy integration between back-end & instant DB connection. Material Design support in Android Studio Material Design is also supported by (Android Developers, 2023) to design visually consistent and intuitive end-user interfaces.

2. Java

Figure 4.5.2: Java (Oracle, 2023).

The programming language which the mobile application is developed in is Java; because it runs on all OS (Operating System) and platforms, high scalability and library support. It offers an object-oriented environment for developing web applications and this is important to represent information related to complaints (we mean complaint data) where user verification process become a most critical issue. Java's flexibility provides for easy maintenance and high performance of the system, upgrade versions are also possible. Additionally, its years of stability in Android development have made it a favourite for mobile solutions that demand the reliable and structured coding standards (Oracle, 2023).

3. Firebase

Figure 4.5.3: Firebase (Google Cloud, 2023).

Firebase is a cloud-based backend for web and mobile that provides real-time database, authentication and cloud storage capabilities of the system. It enables the synchronization of management and resident data, so that complaints update instantly and so that information flows smoothly. Firebase improves security of an app by providing a secure way for users to authenticate to the system and handling data back-end requests via database rules, encrypting access to user data so that unauthorized users cannot read or modify in any way. It also scales and can be integrated with Android Studio It is suitable for apps that require real-time data processing & analytics. The project is benefitted by using Firebase to guarantee seamless, secure, and convenient data management (Google Cloud, 2023).

4. HTML5

Figure 4.5.4: HTML5 (W3C, 2022).

In the management dashboard application, HTML5 is implemented as core in the structuring of responsive and accessible web pages. It also supports multimedia integration and cross browser consistency without the need of additional plugins, which helps to make sure the reporting dashboard works seamlessly across all types of devices. HTML5 also enhances semantic structure and searchability so that management users can access and interact with data more women nature. The standardization of HTML5 supports the compatibility and performance over time in web applications (W3C, 2022).

5. CSS3

Figure 4.5.5: CSS3 (MDN Web Docs, 2023).

The Web Dashboard uses CSS3 for displaying and layout, helping to make the user interface aesthetically pleasing, uniformed and intuitive. It enables advanced design elements such as transitions, animations, and responsive grid systems for screens of all shapes and sizes. 'Through the use of CSS3 a business elegant yet accessible user experience can be achieved for both occupants and management. Its modular and efficient nature helps reduce code duplication and provides maintainability to the interface as a whole (MDN Web Docs, 2023).

6. Java Script

Figure 4.5.6: JavaScript (Flanagan, 2020).

JavaScript, as the scripting language forms the foundation of dynamic interactions and real-time responsiveness in, web dashboard. It supports data updates without page reloads, which is convenient to manager who filter and view complaint records. JavaScript also provides support for input validation, so that user-provided data can always be converted and treated safely. Giving HTML5/CSS3 fluent experience with powerful front-end interaction. Flexible and heavily community-based nature of the language makes it a prominent factor when creating dynamic web applications (Flanagan, 2020).

4.5.2 Hardware Requirements

The hardware system requirements specify the physical components that are needed to ensure acceptable performance, stability and functionality of a software application. Such requirements often include, for example, what operating systems the user's computing device is to have, the type of processor of the computing device, how much memory and storage space are available in the user's computing device. This has to be fulfilled in order that these elements can perform efficiently during development, testing and operation (Sommerville, 2016).

1. Laptop

Figure 4.5.7: ASUS ROG G15 (ASUS, 2022)

Table 4.5.1: Hardware Specifications

Computer Brand	:	Rog Strix G513QC
Processor	:	AMD Ryzen 7 5800H
Ram	:	16GB
Graphics Processor	:	RTX 3050 4GB
System Type	:	64-bit operating system, x64-based processor
Windows	:	Windows 11 Home Single Language

2. Smartphone

Figure 4.5.8: Realme 6 (GSM Arena, 2020)

Table 4.5.2: Hardware Specifications

Brand	:	Realme
Model	:	Realme 6
Chipset	:	Mediatek MT6785 Helio G90T (12 nm)
RAM	:	8GB RAM
Storage	:	128GB
Operating System	:	Android 11

4.6 Conclusion

This chapter had elaborated on the fundamental requirements in developing the Shamelin Star e-Reporting System, which is to computerize condominium complaint reporting management. The functional requirements WEREAR set straight out is to deliver features like digital complaint, track on real time basis, report panel, AI Based Chat Bot assistance, so that both residents and management staff get very good facilities for transparent communication. On the other hand, all the non-functional requirements handle significant aspects as system reliability, usability, security and performance that aims to guarantee application works fluently and securely in a real world.

The chapter additionally covered the ways of collecting data such as interviewing the condominium management and surveys handed out online to users, enabling identification of user activities and needs. Moreover, the system requirements in terms of software and hardware were also well defined, to facilitate the design and implementation of both, system development and deployed components. Tools like Android studio, Firebase, Java, HTML5, CSS3 and Javascript were considered appropriate technologies in order to develop and maintain the mobile application efficiently.

In summary, this chapter provides a good base for future work which will focus on design, implementation and testing. But if functional and non-functional team have already been defined, the project is set up for success in delivering a tasks-driven, safer, and user-friendly application platform that helps to communicate effectively and efficiently between residents and condominium management.

5 ANALYSIS

5.1 Introduction

Analysis is the process of dissecting all the functions and processes by which Shamelin Star e-Reporting System operates and represents it in such a way that everything that comprises the whole working (the functionality and non-functional) is properly understood for efficiency4. The analysis aims to capture community sentiment, patterns of delivery around reporting issues, and the speed at which staff responds to management, as their perceived system requirements and customer needs. The objective is to move the existing manual complaint process into a formalized, data-informed system that will facilitate communication between residents and management. In a domestic setting, where both time sensitivity and communication is key this analysis helps to create better transparency in the service process, fast enchained issue resolution but also that users comfortably report accurate information.

5.2 Data Gathering Analysis

Interview and online questionnaire were used to collect data for the Shamelin Star e-Reporting System. Condo management staff were interviewed to determine how they currently operate in the paper-based system, and the barriers that exist regarding processing/responding to resident reporting forms on incidents, as well as what they might vision a digital system should look and act. Further more, a questionnaire through Google Forms was sent out to the residents of Shamelin Star Condominium to interview them regarding their satisfaction towards the existing form of reporting system and level of transparency as well as preference on features available such as real-time tracking complaints and photo attachment.

The management interviews showed that the primary problems are misplaced forms of complaint, late responses and non-availability of patterned reports to analyze. One day later, residents complained they were unable to monitor their own complaints and an absence of quick communication between them and the managing office. By just exposing findings from both groups, the information collected contributes to elucidating what are the essential system requirements and features for developing an effective, transparent and easy-to-use e-reporting mobile application that fulfils the need of both residents' and operator’s staff's parties.

5.2.1 Questionnaire Analysis

The form for the Shamelin star e-reporting system was developed to enable feedback from residents on their experiences with paper-based complaint lodgment. It was disseminated for easy access and completion via Google Forms on October 3, the year 2025. The primary goal of this survey was to seek feedback on how residents currently report facility and maintenance issues, if they are satisfied with the

time it takes for issues to be resolved, and if they would like a new way to report using a mobile base system.

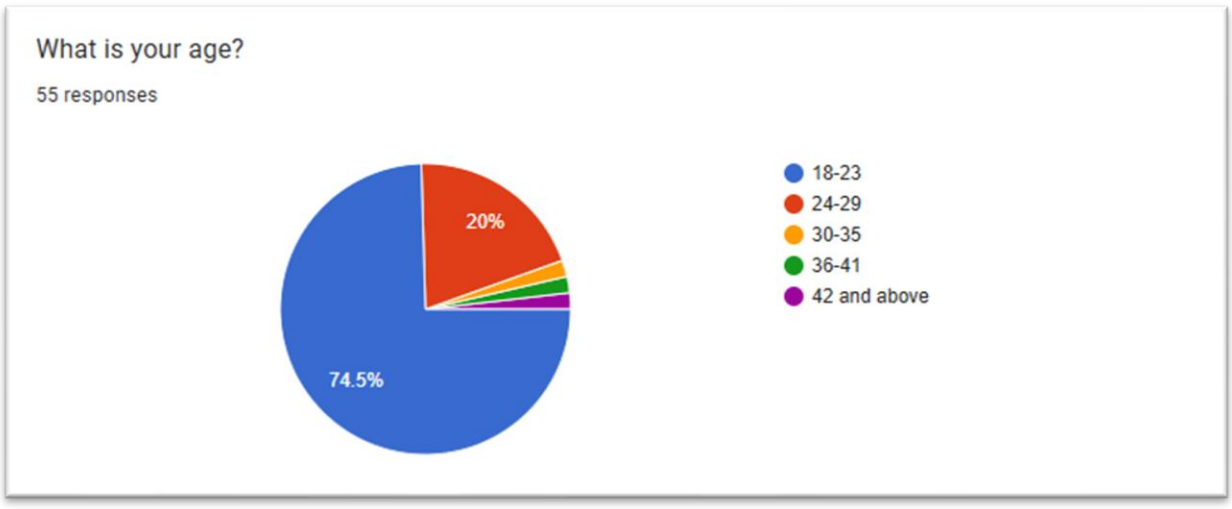


Figure 5.2.1: Question 1

It shows 74.5% 41 of the participants are '18-23'. 11 participants or 20.0% (‘24–29’). '42 and older' represented 1.8% of the sample, or one participant. For 1, who were '30-35', the percentage was 1.8%. '36-41' accounted for 1 subject (1.8%). We received 55 responses to our questionnaire and this breakdown of the demographics is informative. This dissection has brought some essential information about what the respondents believe. It shows that more respondents (74.5%) are between 18 and 23 years of age, which could potentially impact the design and usability development of the system.

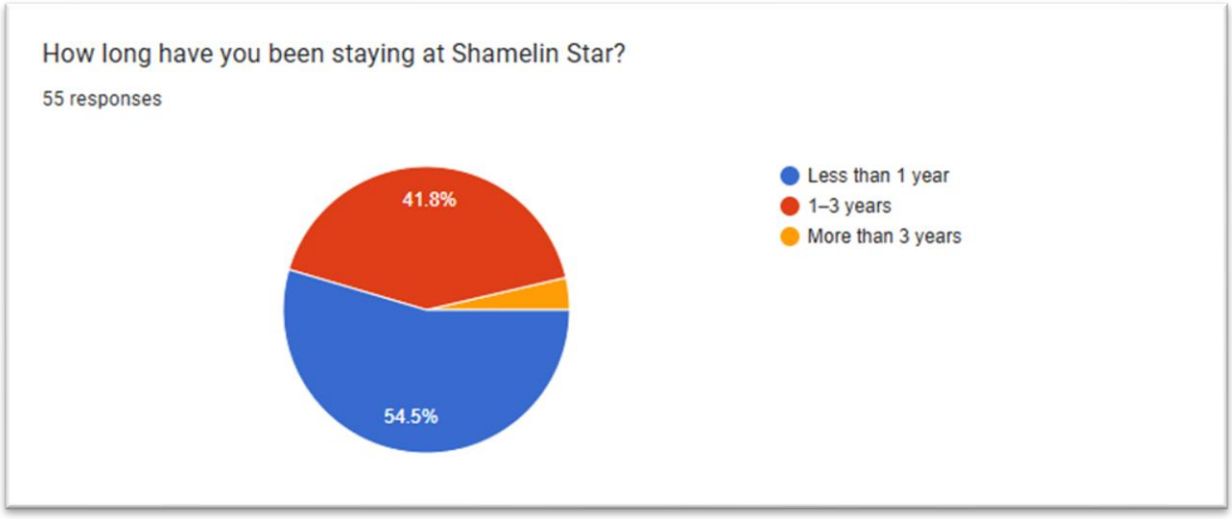


Figure 5.2.2: Question 2

54.5% of respondents were 'Less than 1 year'; proportion being 30 persons. The 1–3 years had highest percentage (41.8%) with 23 respondents. Only two participants were in the category 'More than 3 years' (3.6%). There were 55 employees who in total answered the questionnaire, and this distribution is highly interesting when analyzing their answers. This statistics can help us get a more complete feeling about

the respondent's point of view. It suggests that often (54.5%) participants chose “Less than 1 year" which could affect the decision about the system’s design and usability factors.

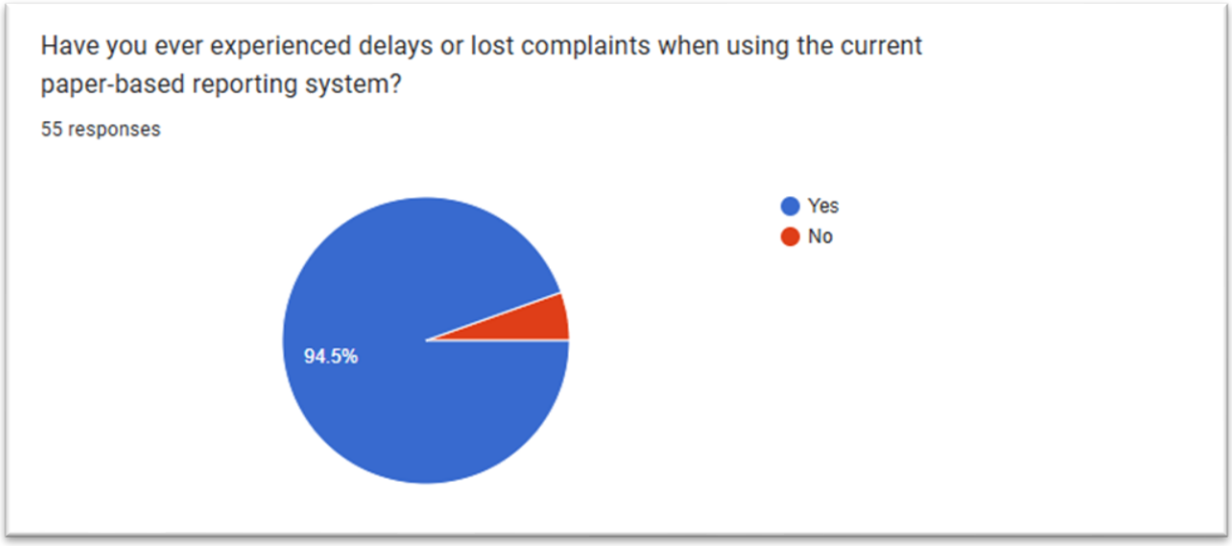


Figure 5.2.3: Question 3

“Yes” was reported by 94.5% respondents 52. 5.58% of the responder “No” (3 persons). Although the number of respondents to the questionnaire was 55 in aggregate, demographic distribution is interesting. This is an interesting breakdown on the views of respondents. It shows that an overwhelming majority (94.5%) chose 'Yes' and this could affect system design and usability issues.

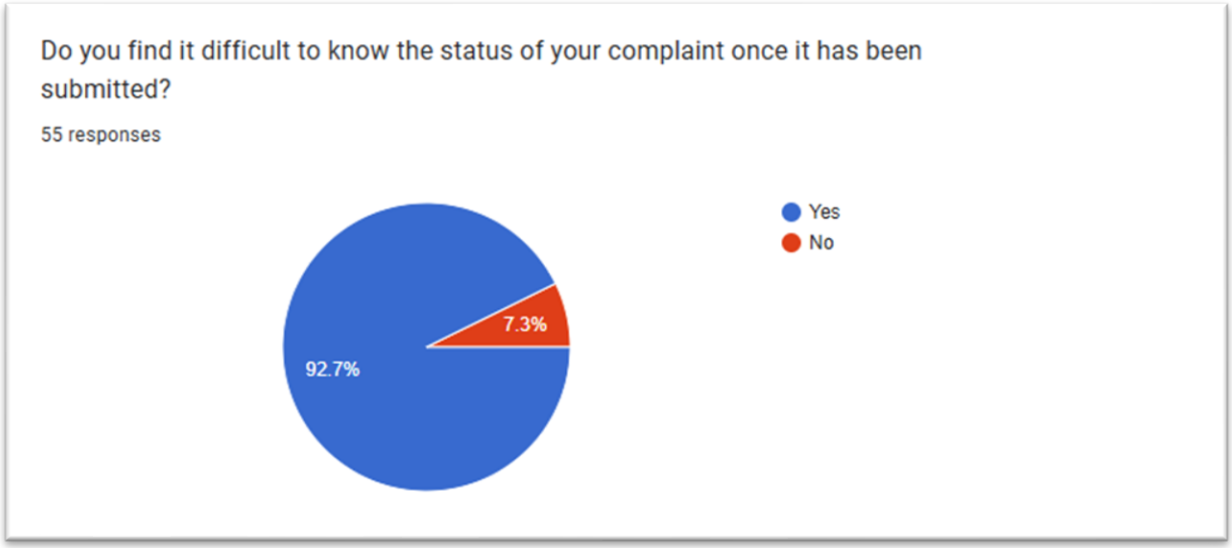


Figure 5.2.4: Question 4

Yes, 92.7% 51 of participants did so. The 7.3% of the participants “not at all”, 4 participants. A total of 55 people responded to the questionnaire, and this demographics is worth noting for an interpretation of the responses. This part-by-part breakdown is quite informative about the attitudes of respondents. The

majority (92.7%) responded "Yes," which might affect system design, and usability aspects in systems are possible to consider.

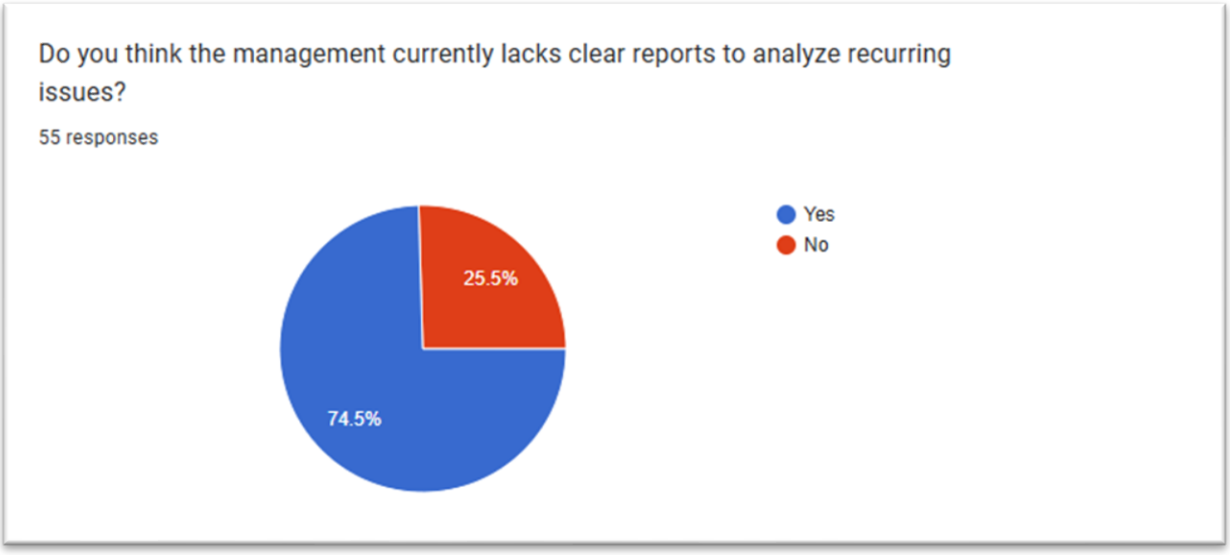


Figure 5.2.5: Question 5

Yes inches 74.5 percent or 41 people. 25.5% of the respondents 'No' representing 14 participants. Overall, 55 people responded to the survey. This is a demographic profile of respondents that should be noted as you read through responses: This analysis is revealing of what the respondents believe. This reflects that most of the users (74.5 %) answered 'Yes' and this information can have implication on design and usability aspects for the system.

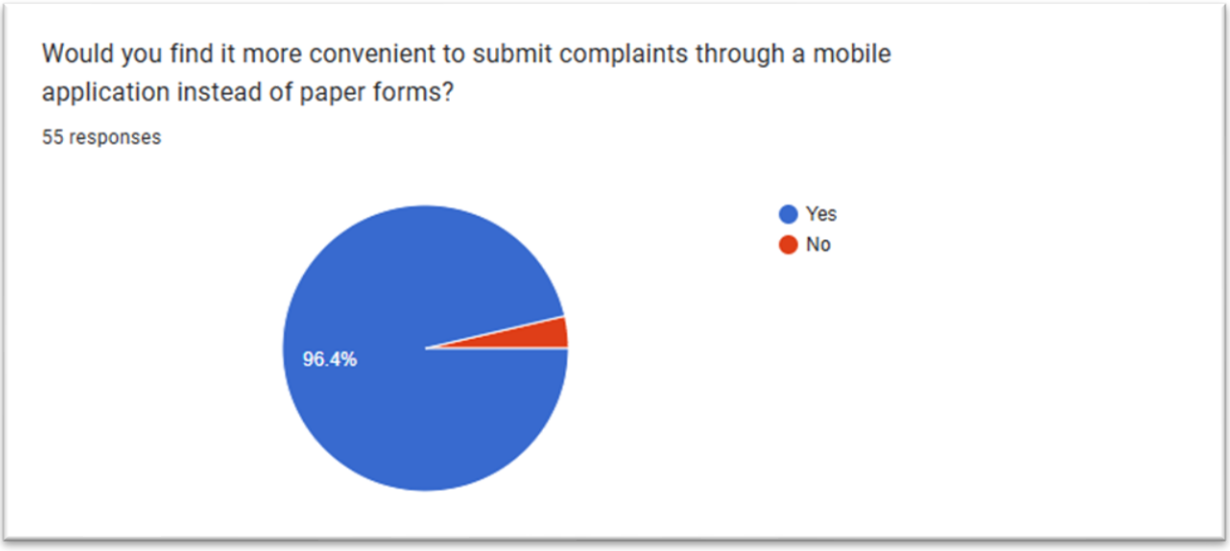


Figure 5.2.6: Question 6

'Yes' was also replied by 96.4% of the respondents, that is 53 persons. 3.6 % refused to answer and also indicated 'No' 2 people. There were altogether 55 responses to the questionnaire, and this profile brings interesting information about the respondents. This analysis is instructive about the respondent's frame of mind. It is also worth noting that 96.4% selected 'Yes', which might affect the design and usability of the system.

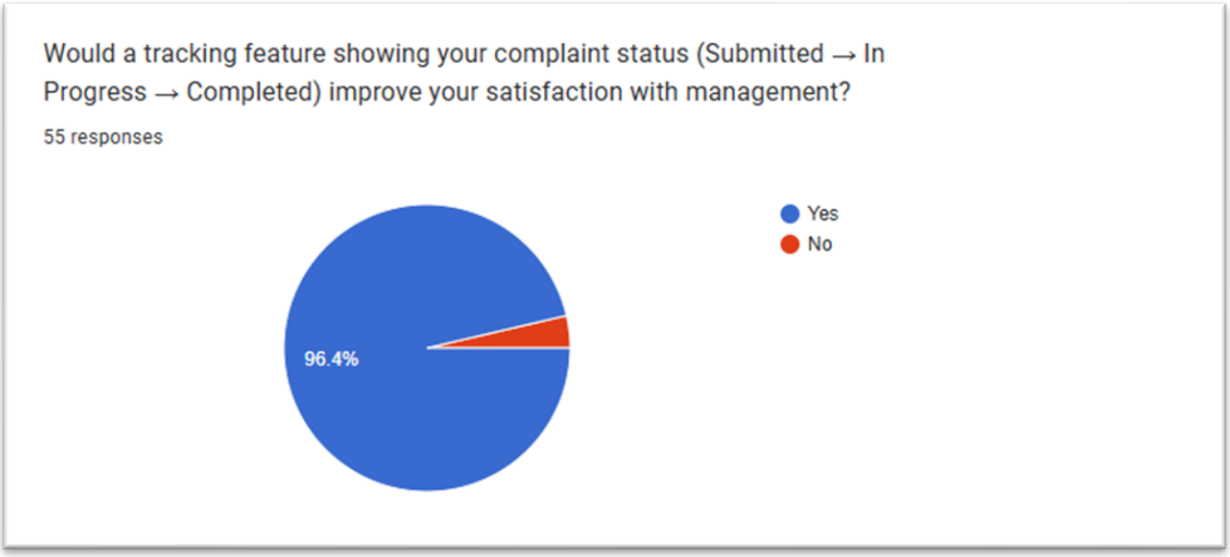


Figure 5.2.7: Question 7

96.4% said 'Yes', which included 53 respondents. Only 2 respondents still think that 'No', which was 3.6%. There are a total of 55 respondents to the survey, and based on this demographic profiling, it is possible to understand accordingly to its answers. This breakdown is enlightening about how the surveyed feel. It implies that more than 96.4% voted as 'Yes', which can be useful for the design and usability point of views of the system.

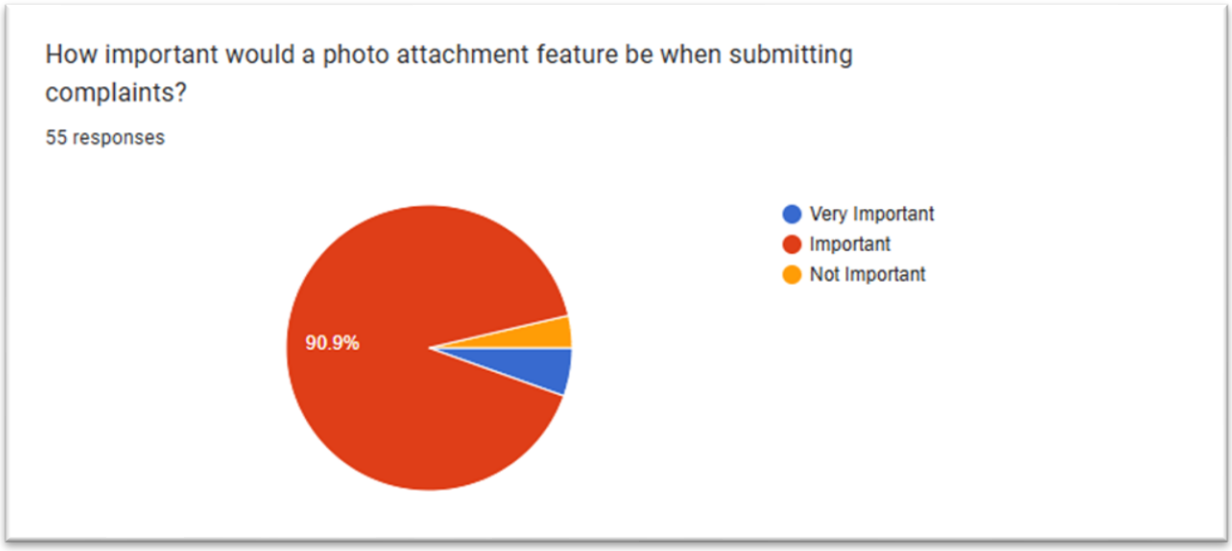


Figure 5.2.8: Question 8

90.9% (50 people) Respondent "Important". Very Important' represented 5.5% of the sample, with 3 subjects. 'Not Important' as an option was selected by 3.6%, representing 2 respondents. Overall, 55 participants filled in the questionnaire and this demographic profile is useful for interpreting results. This breakdown has the useful side effect of revealing how respondents feel. It suggests that most (90.9%) choose ‘Important’ which might be significant for the system’s design and usability considerations.

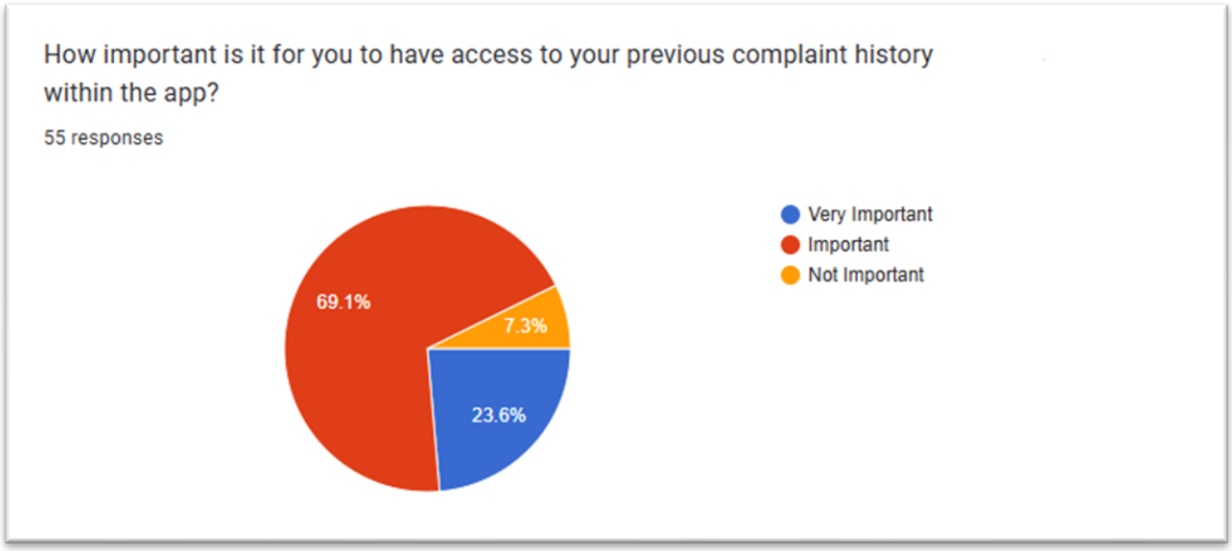


Figure 5.2.9: Question 9

69.1% 38 of the sample rated this anchor as important. 13 people 23.6% with the greatest number of responders; 'Very Important'. 7.3% of the participants "Not Important" 4 people. Note that we had 55 responses in total to the questionaire, and this gives us some useful information as to how the response

demographics are divided. This split of opinions is quite useful information. This suggests that the majority 69·1% chose ‘Important’, which will inform design and usability considerations for the system.

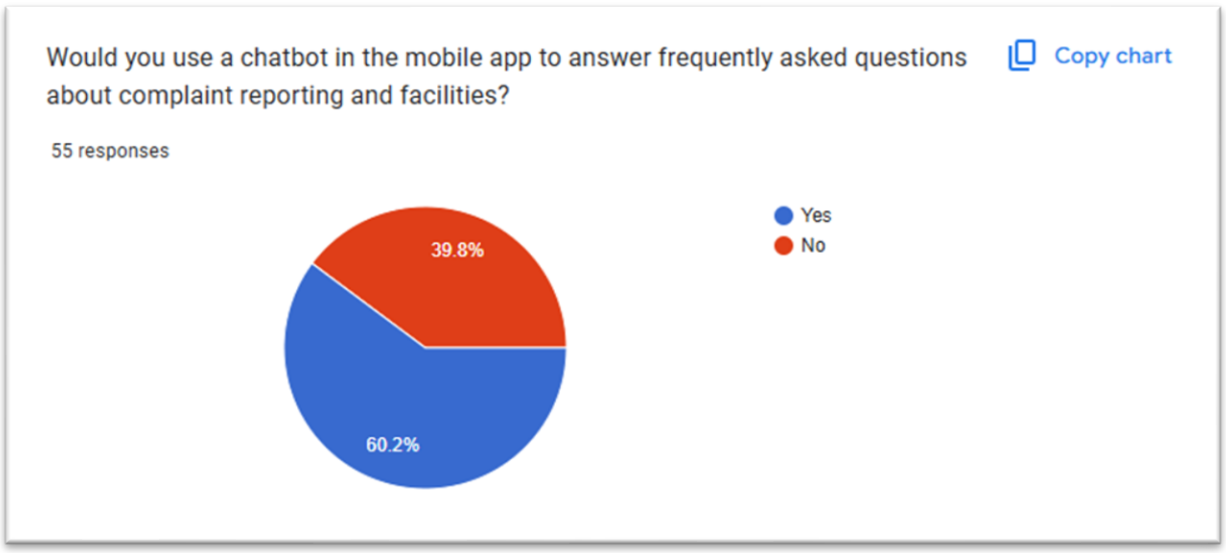


Figure 5.2.10: Question 10

'Yes' 60.2% of the respondents 39. The amount of those who answered “No” was quite high, 39.8% 16 persons. In all, the questionnaire was answered by 55 respondents and this characterization of the people wearing hearing protection is useful to understand the answers. This detail adds important flavor to the perspective of the responders. It suggests that a majority 60.2% would prefer 'Yes' which may impact system design and usability considerations.

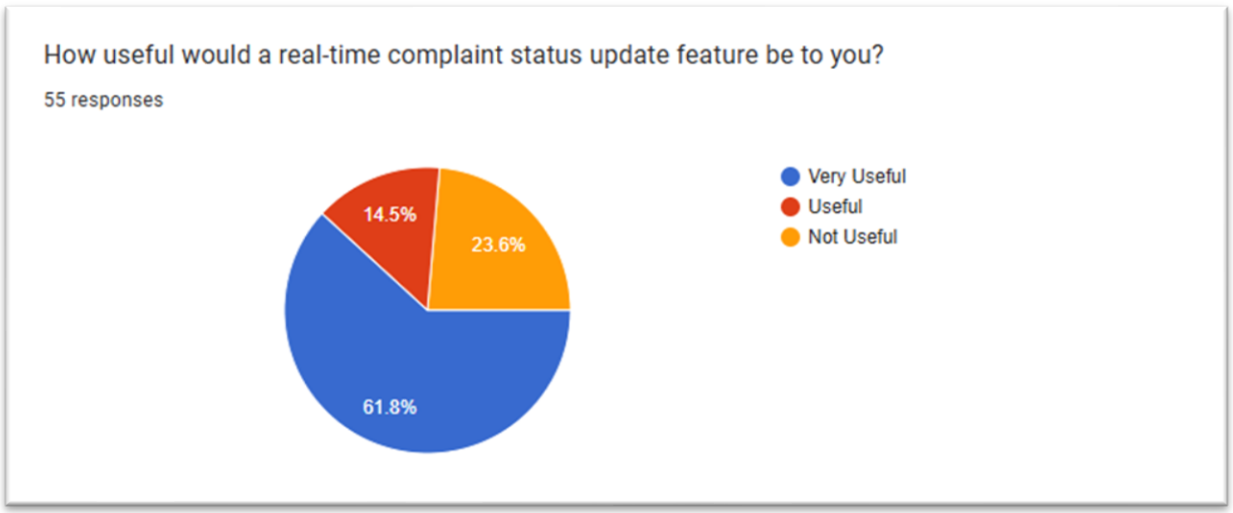


Figure 5.2.11: Question 11

61.8% of all respondents found this 'Very Useful', comprising 34 people. 13 members of respondents (23.6%) labelled it 'Not Useful'. 14.5% of participants ‘find them Useful’, 8 people. 57 people filled in the

questionnaire altogether; I think the following demographics might cast some light on these 57 responses is quite worthwhile. The chart gives a sense of how the respondents feel. It reveals nothing more that the (58.4%) of them choose 'Very Useful' which could be helpful in system design and usability issues.

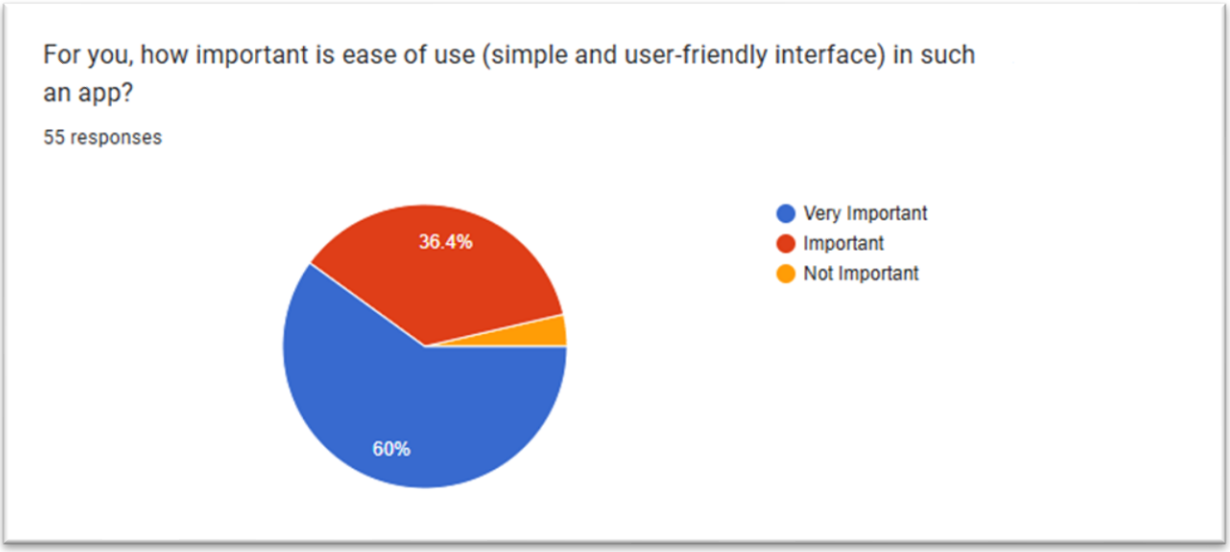


Figure 5.2.12: Question 12

60.0% of respondents considered 'Very Important', 33 individuals. Important 20 respondents 36.4%. 3.6% 'Not Important', 2 people. All in all we had 55 questionnaire participants, both of which should give you perspective on the answers. This dissection says a lot about the attitude of the respondents. With all respondents selecting 'Very Important', this result may have an impact on the design of system and aspects of usability.

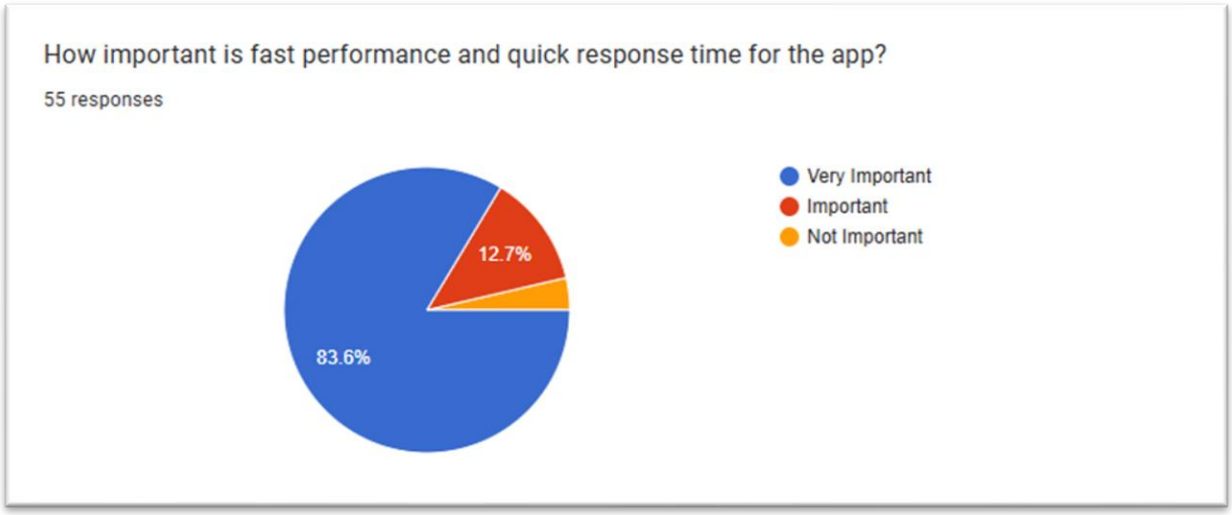


Figure 5.2.13: Question 13

83.6% of survey takers who answered this question (46 people) chose 'Very Important'. 12.7% 7 'Important' with 2 men 3.6% of the participants 'Not Important'. Altogether, there were 55 students who answered the questionaire and this gender breakdown can give us new information about responses. This attitudinal decomposition will be useful in understanding the different attitudes of respondents. This implies that a large proportion of such subjects (83.6%) finds coder-roles highly important, which can affect the design and usability decisions of a system.

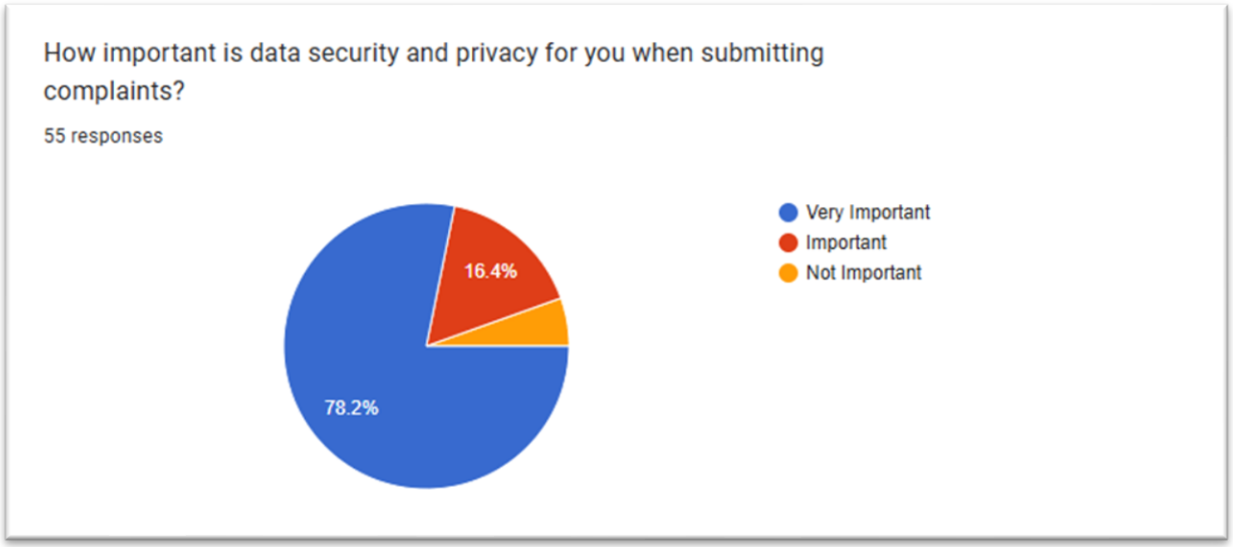


Figure 5.2.14: Question 14

This represented 78.2% of the respondents who responded ‘Very Important’ 43. Important: 16.4%, representing nine persons. 5.5% 'Not Important', with 3 people. The respondents to the questionnaires were 55 in all, and this composition has certain significance with regard to the responses. Such a

bifurcation adds insight to the views of the respondents. It puts out the fact that the large majority (78.2%) rated ‘Very important’ and it could have implications on system design and usability concerns.

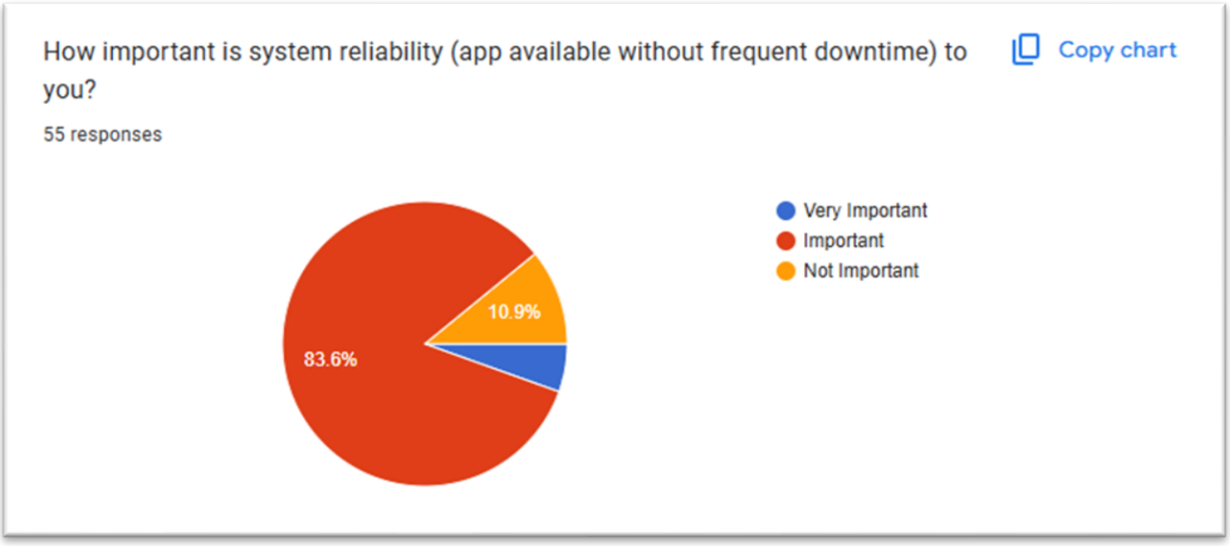


Figure 5.2.15: Question 15

83.6% of participants 46 rated 'Important'. The number of percent from the participants 10.9% 'Not Important' to counted 6 people. 5.55% of participants reported 'Very_Important' 3 patients. Altogether, 55 people answered the survey and all the information adds up to some interesting findings. This reduction is particularly useful in comprehending the opinions of the respondents. It suggests the fact that most (83.6%) have chosen “Important” as selected option, this could potentially affect the architecture of system and usability perspective.

5.2.2 Interview Analysis

The management of Shamelin Star Condominium gave interviews for qualitative inputs on existing complaint process and issues arising out of it. Qualitative interview analysis with transcripts is commonly reviewed and interpreted for themes and repetitive categories (Sheppard, 2020). According to (Puri, 2023), interviews can be used to gather in-depth information from those directly involved in the process, giving a deeper insight into their needs and views.

The interview with the Shamelin Star Condominium management revealed several issues and requirement of its current complaint-handling process. Dilemmas related to the current paper-based system of reporting and value of developing a digitalized system formal questionnaire responses uncovered high level issues in relation to the current paper based process and also pointed towards efficiency, transparency and improvement in communication between residents and management that could be achieved by shifting from the existing papers based process to a digitalized one.

Table 5.2.1: Interview Questions

Questions 1	How do you currently receive and manage residents' complaints about facilities or maintenance issues?
Answer	At the moment, most of the complaints are submitted through physical forms at the management office or occasionally via phone calls and WhatsApp messages. Once received, the staff manually records the details in a logbook or spreadsheet and assigns the task to the maintenance team. Follow-ups are usually done through verbal communication or phone calls, which can make it difficult to keep track of multiple ongoing complaints at once.
Analysis	According to the interview data, management now depends on physical complaint forms, phone calls and WhatsApp messages to take stock of complaints from occupants. Although this conservative method has the advantage of direct communication it is extermely inefficient as there is no organised recording mechanism.

Table 5.2.2: Interview Questions 2

Questions 2	What are the main challenges you face when using the current paper-based/manual reporting method?
Answer	The biggest challenge is the lack of organization and traceability. Physical complaint forms can be misplaced or damaged, especially when there are a large number of submissions in a short time. It is also time-consuming to manually search for past complaints or check the status of an issue. Additionally, there's often no proper record of when a complaint was submitted or resolved, making it hard to measure performance or ensure accountability.
Analysis	The interview also disclosed that manual process's problems, are bad data organization, time consuming of finding past records and hard to provide the situation to the citizens. This inefficiency not only causes residents to be unhappy, but it also limits management's ability to oversee the maintenance work. The management stressed that a more organized and transparent system is required to ensure complaints are received, recorded, tracked and addressed in a predetermined period.

Table 5.2.3: Interview Questions 3

Questions 3	Can you describe situations where complaints were delayed, lost, or not followed up properly?
Answer	Yes, there have been several occasions where residents submitted complaints about leaking pipes or faulty lighting, and the forms were misplaced before being passed to the maintenance team. In some cases, complaints were received verbally but not recorded immediately, leading to delays in action. Residents would sometimes come back after a few days to ask for updates, but since we had no structured tracking system, it was difficult to provide clear answers or confirm the status.
Analysis	Problems such as lost forms, late updates and verbal complaints that have not been recorded are common since no central place exist. Therefore receiving a slow response to complaints and little reward for submitting them.

Table 5.2.4: Interview Questions 4

Questions 4	From a management perspective, what features would make complaint handling faster and more efficient?
Answer	A digital system that allows residents to submit complaints directly through their mobile phones would save a lot of time. Features such as automatic complaint logging, real-time notifications to staff, and a dashboard to monitor pending and completed complaints would make the process more efficient. It would also help if residents could attach photos to their reports, as that would make it easier for the maintenance team to understand the problem before arriving on-site.
Analysis	The interview evidence robustly supports the sense of a digital e-reporting mobile application as a means of improved communication, transparency and data-informed management that is required.

Table 5.2.5: Interview Questions 5

Questions 5	How useful would a complaint tracking system be for you in terms of transparency with residents?
Answer	A complaint tracking system would be extremely useful. It would allow both residents and management to view the current status of each complaint, which would help reduce the number of follow-up calls and

	misunderstandings. This transparency would also improve trust between residents and management since residents could see that their complaints are being addressed promptly.
Analysis	Regarding system capabilities, the management showed high interest in receiving a digital complaint submission systems which includes features for real time tracking for reporting persons, automated notifications and resident's ability to upload pictures of reported matters.

Table 5.2.6: Interview Questions 6

Questions 6	Do you see value in having structured reports that allow filtering, downloading, and analyzing complaint data?
Answer	That would indeed be most useful. If we were to receive structured reports, this could also be integrated with operational maintenance planning - particularly if there is a local recurring problem occuring that requires preventative action; for example an excessive number of water leaks in a specific block or lift break downs etc. It would also be useful for resource planning and monitoring of staff performance, etc., as well as preparing monthly reports to committees. In general it would be better decision-making and planning for long term maintenance.
Analysis	I thought one of the other interesting things that came up was sort of the potential beneficial role of structured reports. Management agreed with this recommendation and noted that downloadable, filterable reports could be helpful in identifying repetitive equipment failures, scheduling preventive maintenance, and monitoring staff performance. This evidence-based decision-making would dramatically enhance operational decisions and future facility planning.

5.3 Use Case Model

UML Use Case Diagram is a model used to define a system border, the set of actors and their behaviors in regards with the subject in hand for example what the system should do, focusing on what users expect from an implemented system, user goals, not on how they are satisfied (Visual Paradigm, 2019). For the Shamelin Star e-Reporting System, this user case model assists in understanding how residents, management and system interact with the complaint reporting and tracking process. The system applies principles of user-centered design and systems thinking to visually describe the functional requirements such as submitting a complaint, tracking status, structured reports, chatbot assistance.

5.3.1 Resident

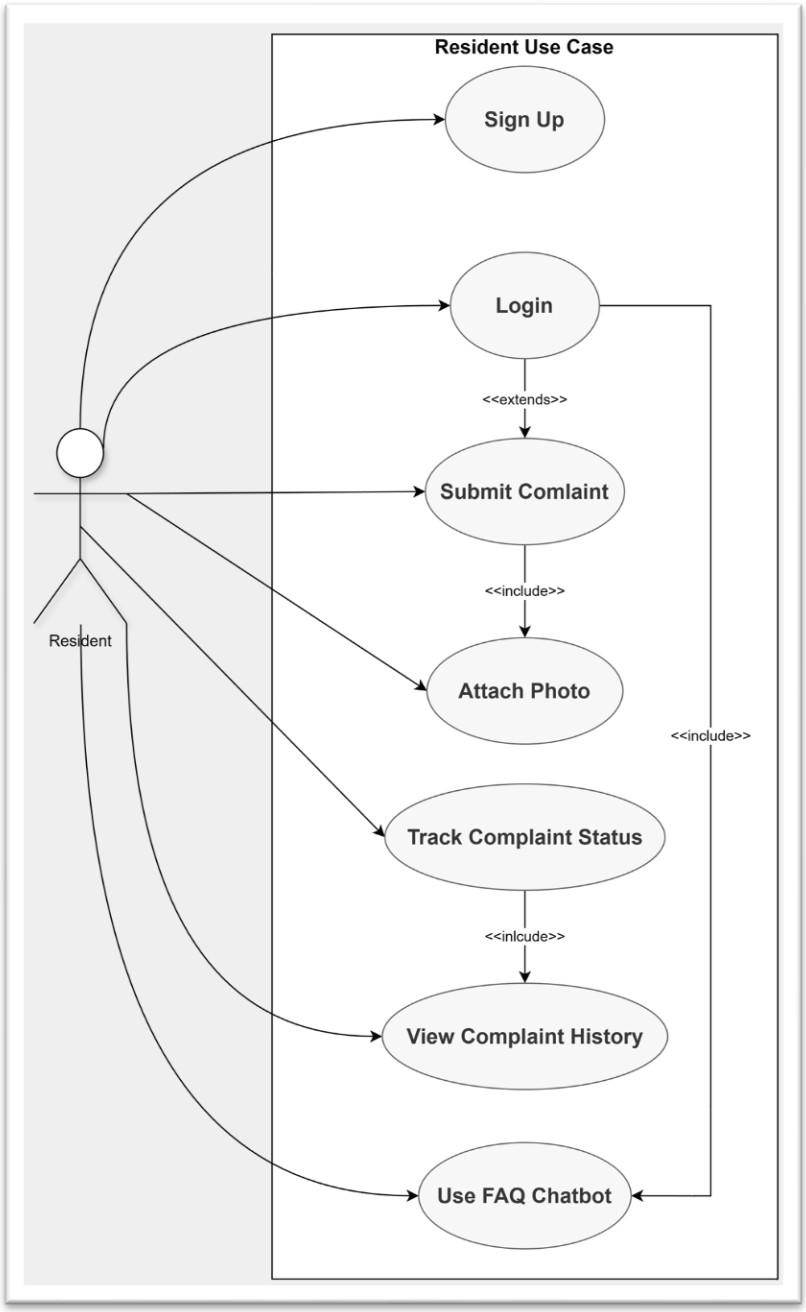


Figure 5.3.1: Resident Use Case Diagram

As per Figure 5.3.1, the Resident Use Case for Shamelin Star e-Reporting System enables residents to register and login to the system providing them an efficient way of handling their facility or maintenance complaint. Upon clicking on the link, residents will also have the ability to file a new complaint in which they attach photos for clarity; residents can monitor their submission in real time as status updates follow from Submitted to In Progress then Completed. Resident can also follow up on their complaint history to track re-occuring issues and even message the management directly through the platform. Features like these

help maximize the convenience, transparency and communication between residents and management; giving them greater autonomy to play an active role in making their housing better.

5.3.2 Management

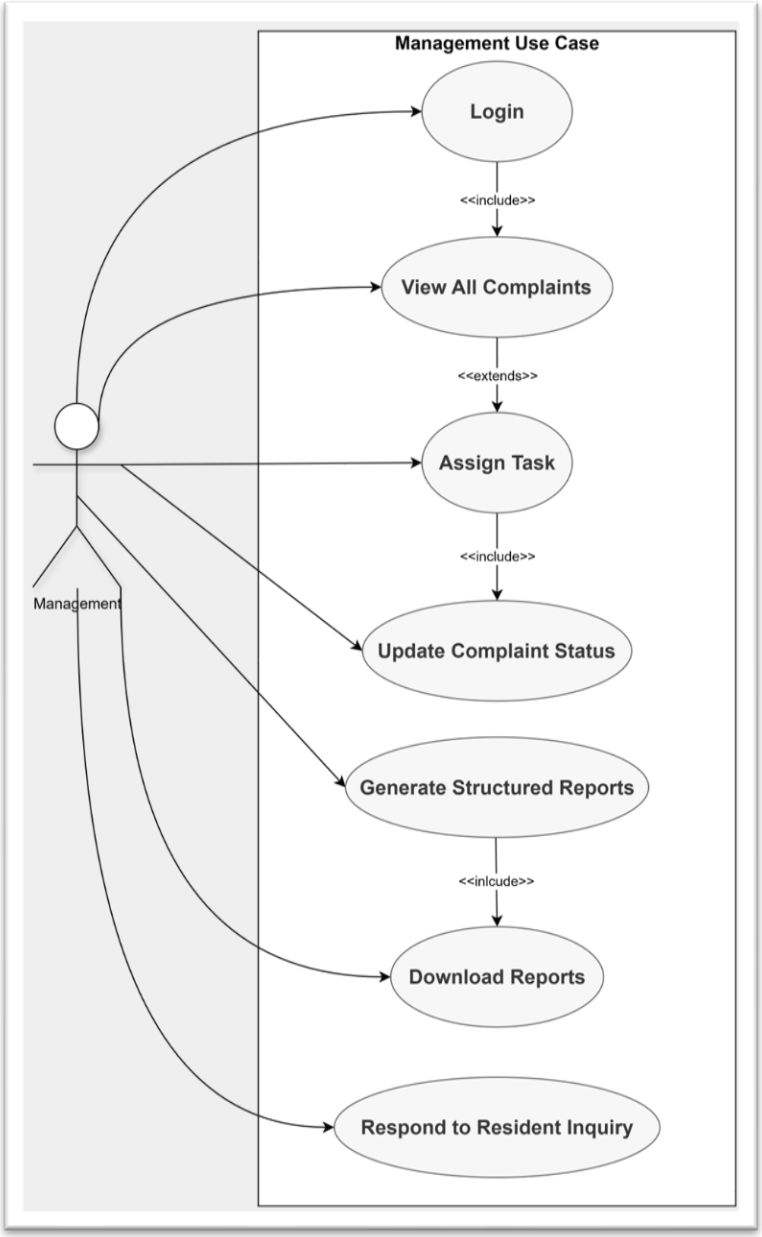


Figure 5.3.2: Management Use Case Diagram

From Figure 5.3.2 the Management Use Case of Shamelin Star e-Reporting System also represents that condominium staff will be able to well-manage resident's complaints in an orderly manner using digital platform. Managers can login to access submitted complaints, assign tasks for the maintenance team, and change complaint status to notify residents. The system enables management to create filtered and downloadable reports of a structured nature, for performance assessment and maintenance scheduling. In addition, integration with the chatbot backend allows management team to automatically address common

resident questions and therefore minimize repetitive communication efforts. In general, this feature also allows for a more organized, accountable and responsive management of facility issues.

5.4 Flowchart

Flowchart A flowchart is a type of visual diagram that provides the sequence of steps or processes within an entity, allowing complex workflows and relationships to be easily understood between functions (Lucidchart, 2019). For the Shamelin Star eReporting System, flowcharts are developed to help organize and analyse how data flows from one part to another in the complaint management system.

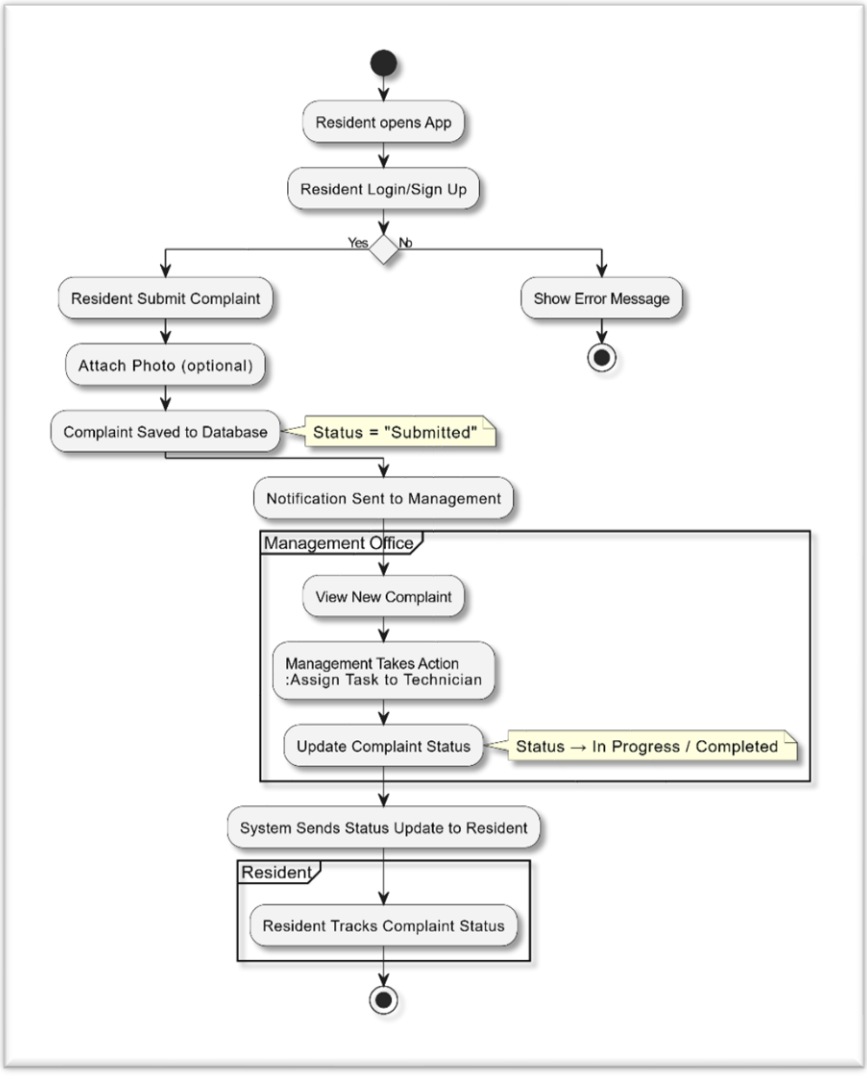


Figure 5.4.1: Flowchart

On the basis of Figure 5.4.1, residen flowchart is drawn to represent a process that depicts users interact with Shamelin Star e-Reporting System. The user opens the app, logs in/signs up and goes to homepage. After they log in, residents can file complaints by providing descriptions and attaching pictures. Once the

complaint is filed, it will be retained in the system's database and its status can be tracked by the user in real-time. Notifications are issued upon any status change, providing transparency until the problem is solved.

Figure 5.4.1, a management diagram demonstrating the process of which the administration deals with complaints. The staff can view new complaints as soon as they login into the admin dashboard and assign them to maintenance teams or update their status. Upon finalizing the maintenance request, mark the complaint status as Completed and automatically notify the resident. It allows the management team to produce structured reports for performance and early intervention analysis, increasing efficiency and documentation.

5.5 BPMN (Business Process Modelling Notation)

Visual Paradigm (2010) indicates that Business Process Modeling Notation (BPMN) is a graphical language for specifying and representing the precise steps required to achieve business processes. It aims at making complex flowcharts more straightforward to understand straight away, by implementing common symbols like events, activities, gateways and threads which can be used to indicate how information and tasks move amongst participants in a process. BPMN diagrams are particularly valuable as everyone, technical or non-technical, from web developers to the management team and local residents alike can understand how a system works and communicate that.

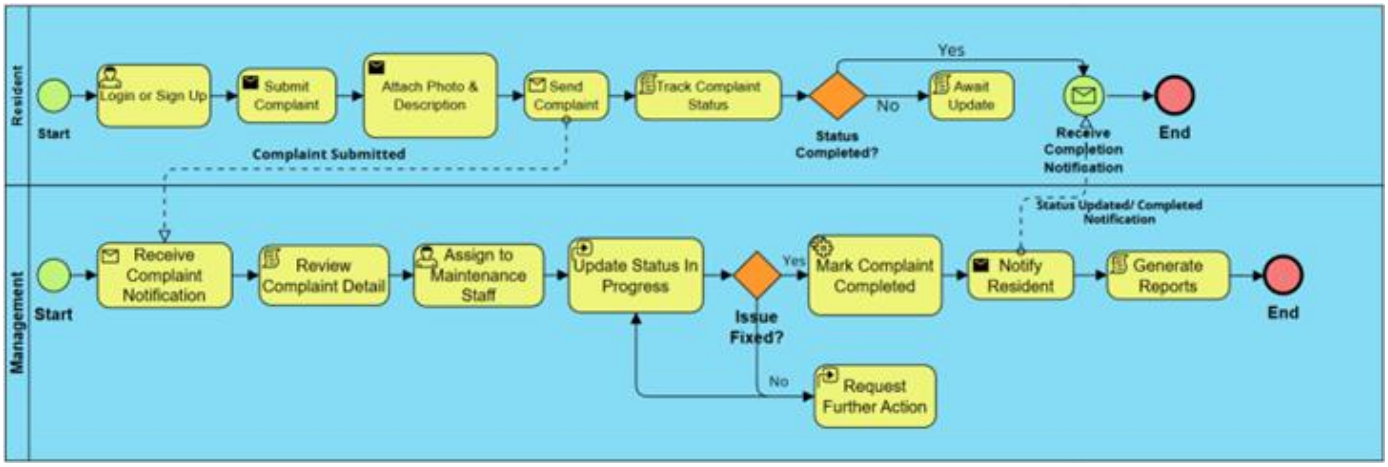


Figure 5.5.1: BPNM

According to Figure 5.5.1, the BPMN notation of the Shamelin Star e-Reporting System exhibits the end-to-end workflow. The resident would first log into the mobile application, log a complaint and if necessary attach photos to illustrate their concern. If the management office receives a complaint, it will read the details of the claim and assign it to someone responsible for maintenance issues.

The system changes the status of the complaint from “Submitted,” to “In Progress” and ultimately, “Completed,” enabling residents to track progress live. citizens are then automatically notified of a resolved issue, ensuring transparency and accountability. Furthermore, administrative personnel are able to create customized reports and identify problems before they occur resulting in more efficient preventive maintenance.

This BPMN model serves as a uniform and easily comprehensible overview of how digital complaint management positively improves communication and efficiency between residents and management.

5.6 Conclusion

Conclusion In this chapter, we have provided a thorough investigation of the Shamelin Star e-Reporting System through different modeling and analytical tec hniques. Information collected from the residents and management, revealed major loophole of problems such as non-effective manual reporting with unavailability in terms of complaint tracking mechanism combined with scope for further report processing being absent. Findings from both the questionnaire and interview were used to identify important user requirements and expectations, which provided a solid platform for system design and development.

Moreover, the application of modelling instruments like Use Case Diagram, Flowchart and Business Process Modelling Notation (BPMN) have brought to light how diverse components of a system and users interact during complaint management process. These models provide a visual representation of the workflow and relationships to aid in guaranteeing that they system being designed matches how actual users work and what is actually required.

The results and models presented in this chapter will lead to next phase of the project System Design which would help to develop application that is according to the needs of user while keeping in mind throughout a model-based approach. Ultimately, this research phase guarantees the Shamelin Star e-Reporting System will successfully facilitate communication, increase transparency and simplify complaint processing for residents and management staff.

6 DESIGN PHASE

6.1 Introduction

The design phase is a key time for Shamelin Star E-Reporting System development since it tells people how the various system components are to be structured and connected or presented to users in order that they may function effectively (Pressman & Maxim, 2020). It 's the real beginning of the construction process, which gives abstract system needs a real structure that is appropriate for this mobile and web designs fillings complaint (Sommerville, 2016).

Residents can make complaints through an application on their mobile phone. Administrators, after these are submitted, can manage that complaint from the web-based management system. (Laudon & Laudon, 2020) Also using the mobile app, residents can enter details and load program scans and photos concerning their complaints in terms of residential facilities. All these are stored at a centralized Firebase Cloud Fire Store database so that the mobile app and management control panel at all times keep their records accurate to each other in realtime (Moroney, 2020).

A secure web-based dashboard platform allows administrators and other management staff to view complaint statuses, query data and update it where necessary. (Sommerville, 2016) This platform also provides detailed statistical reports for all aspects of this particular problem from the total number of complaints registered through to those still pending today or over time, progress rates and cases completed. This design allows management to monitor the condition of complaints regularly, set their priorities and ensure that residents' problems are dealt with as promptly as possible (Laudon & Laudon, 2020).

System design emphasizes ease of use, fault tolerance and the capacity for all system elements to be on line at all times (Pressman & Maxim, 2020). A systematic approach to system design like this will help your residents to work and prosper on interpersonal communication between apartment dwellers and building management (Sommerville, 2016).

6.2 Interface Design

For the Shamelin Star E-Reporting System to practicably operate its Interface Design is a crucial part of the whole. With a well-made user interface, a system may user-friendly and achieve better results in output. But however it is deprived of this facility (Nielsen, 1994).

The user implements functions on a workstation inside which several Resident Client system functions are spread out. As for the interface design, its aim is to make system functions logical so people can use them easily, with minimal effort or skill. This requires that any two parts of the same function be combined into one and data must be consistent both among different parts of a single function as well next time you come back. To achieve these ends without making further demands upon either the user's time or expertise (Pressman & Maxim, 2020).

There are two front lines for resident and administrator in the system Therefore, a web management dashboard was designed for administrators an app intended for the resident. The difference between these interfaces naturally involves functional requirements: in this case users are supposed to address and manage complaints (Sommerville, 2016), which means that the two processes for submission and management can then be run independently.

Management personnel can monitor and manage residential complaints through the administrator interface dashboard from a centralized viewpoint. For example with this interface layout, the person who administers can see complaint details, complaints status, filter the complaints by their stages with a click to resolve them. The central theme of the dashboard conveys clarity on important but not yet addressed facts, which administrators should then ideally attend to Web technologies (Laudon & Laudon, 2020).

Each resident can register complaints via mobile app interface. Users can key in complaint details of one sort or another on the mobile interface, Rhetorical Analysis depending on their needs. The design is simple for residents to accept and easy to operate by brandishing an interface which may guide them through in a "click-and-answer" mode (Nielsen, 1994).

The two interfaces adopt a consistent modern styling, and their menu-driven navigation, responsive designs, and secure authentication methods provide a seamless user experience on any device dispelling the discomfort users of internet great facilities may experience as described in Chapter 1. Below, Section 3 will elaborate on how to design management dashboard interfaces and what they do for instance using Web technologies too now to provide user ease of use and overall convenience throughout (Pressman & Maxim, 2020).

6.2.1 Management Dashboard Overview

The management dashboard is a Web-based administrative tool that acts as the back-end response system of year one houses carrying within the Shamelin Star E-reporting System for handling residential complaints. With this dashboard, managers who have been designated by a system manager gain access to complaint information on a directory basis in order to keep tabs on user-submitted complaints, the current status of them and progress for resolution.

Controlling of complaints and their categories, as well as users; assigning batches of complaints to users for action. All designed via structured user interfaces that allow authorized administrators at different levels to do this work. We designed the dashboard so that it can shift between functions seamlessly while still keeping details readable and plain.

I've created the following key pages and features including an administrator login page, dashboard overview, complaint management interface for submission into a Web-based platform. Access to the dashboard is safeguarded and authenticated to enable only approved personnel use of view and edit complaint data. This highest-level access control ensures data consistency and prevents misuse of the platform by malicious users.

To achieve this aim, each page within the management dashboard is designed to be as neat and efficient as possible. The following sections explain details of how the mother administrators can make various major pages in brief with their functionalities.

6.2.1.1 Admin Login Page

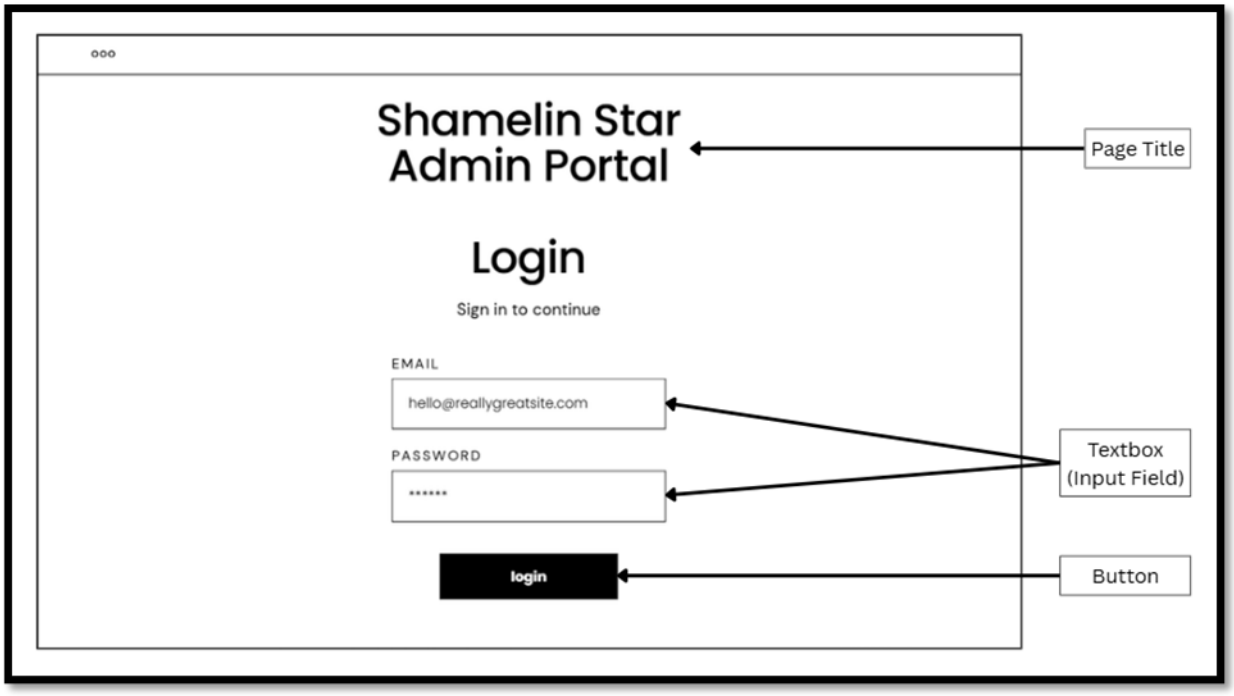


Figure 6.2.1: Wireframe Admin Login

Figure 6.2.1 As an administrative user, you use it to log in. Of these two fields both holding system title and email address when the page loads (with input focus on whichever field is selected), admin password Once entered, the user can hit the Login button. Data is then processed: does with his or her right mind have access rights through this dashboard? of course. A slide-window Login box is provided for the user? The management dashboard is entered by pressing Enter. In a convenient and uncluttered way, the simple interface simplifies use while assuring system security. It only allows authorized individuals to enter.

6.2.1.2 Dashboard Page

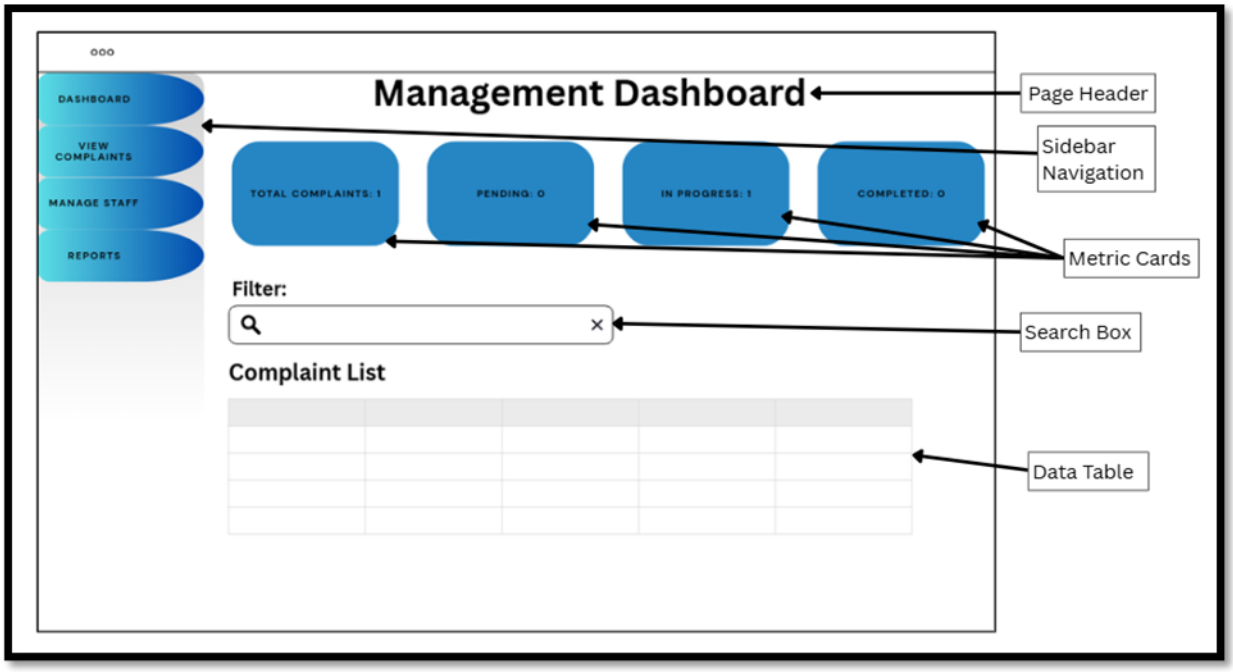


Figure 6.2.2: Wireframe Dashboard

Figure 6.2.2 At present, the company's Dashboard Page is an overview for administrators of complaint activities inside system. A sidebar navigation menu on the left side leads hackers to various management functions, and the page tells us basically where exactly we are within this interface which is with one good header. There are a few metric cards at the top of the page. Number of Complaints, Cases Pending Currents, and Now Being Processed Complaints. Completed cases are also displayed along with any successful cases completed in recent business units put through last month's end to completion this month. At the right side bar if users have had enough of reading about corn and Soybean note that there is a Filter or Search Box where Administrator may wish to find specific complaints rapidly. Then there is a data table of complaint records in an organized way suited for human reading. This type modularization of dashboard layout is good for management to get the basic idea quickly and lay his decisions at once. It goes without saying that information was streamlined: Just glance over it with no words to read.

6.2.1.3 Manage Staff Page

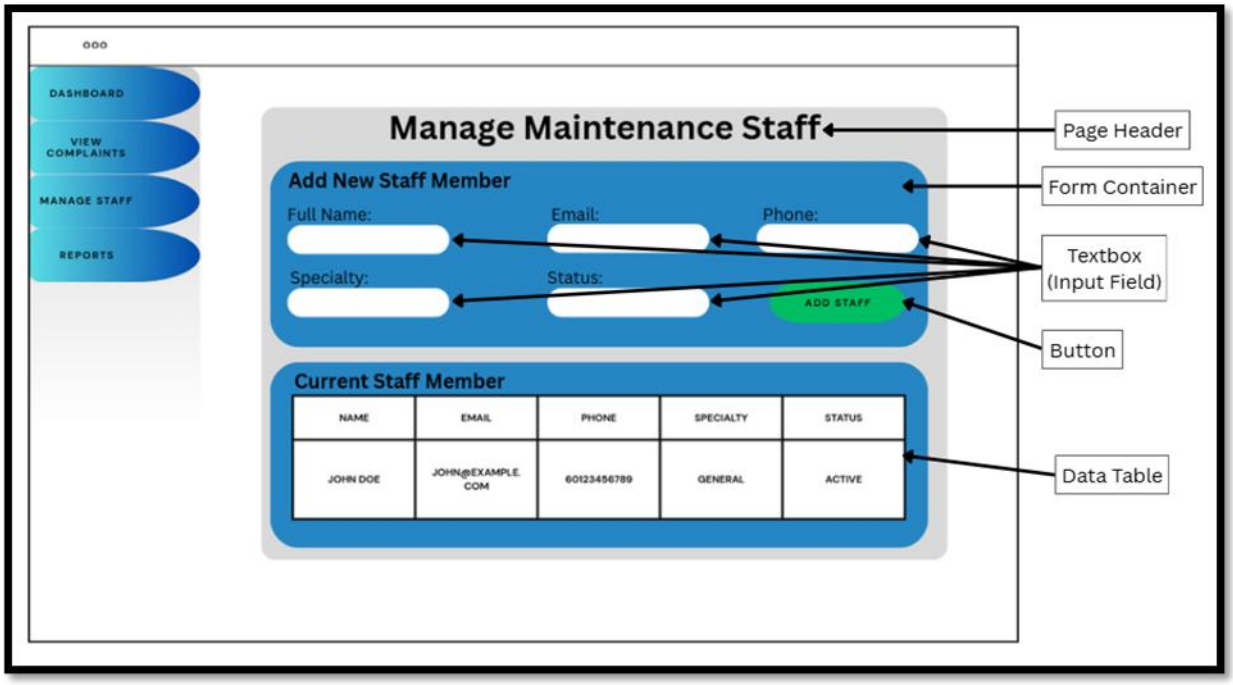


Figure 6.2.3: Wireframe Manage Staff

Figure 6.2.3 Manage Staff Page The rest of the page has a form section, which can be used to add new staff members with information like Full Name, email address, phone number, specialty and status. There is an action button to submit and store the staff information. After the form, there is a data table showing the existing members of the staff along with their information to ease administrators in listing and managing staff. This allows staff management in an organized manner along with storing maintenance personnel information centralized on this page.

6.2.2 Resident Mobile Application

The resident mobile application is the main platform for residents of Shamelin Star to submit and manage complaints in a quick and hassle-free fashion. This role is also for the software which is easy and simple has been made targeting mainly both older generation as well as those who are not very tech-savvy users to start using this application. It lays out information in a manner that is conducive to effective navigation and the fostering of a reasonable user experience.

After logging in, users will be taken to the home dashboard. This provides an overview of the complaints they have submitted up to now, as well as direct access to this application s most-used features. However, simple oversight leads to easy mistakes-being sent back with new Complaints keeps those from ever reaching management Residents also require a structured way to transmit such mistakes.

From this dashboard, users can easily move to the Complaint Submission Page, and Alt-Tabbing is restricted. After all, telling people not to Alt-Tab is easy enough but the whole process of and Tabbing out does not give the user what they want either. By giving them a standardized format for submitting complaints, we ensure that they are easily understood and quickly responded to by our team. In summary, it directly manages quality Residents can inquire at the Complaint Status Page of their complaints' progress, and see whether each complaint is still pending, in process or complete. Furthermore this feature offers openness that helps residents follow things like those moves being taken by management. Additionally, the app also provides a Profile and Settings Page where users can edit their profile as well as logout securely from the system.

The main goal of the resident mobile application is to be simple, fast and safe for all residents and management staff to be able to communicate. Subsequent chapters will introduce the design of each application page in detail and a schematic of the functions implemented.

6.2.2.1 Resident Login Page

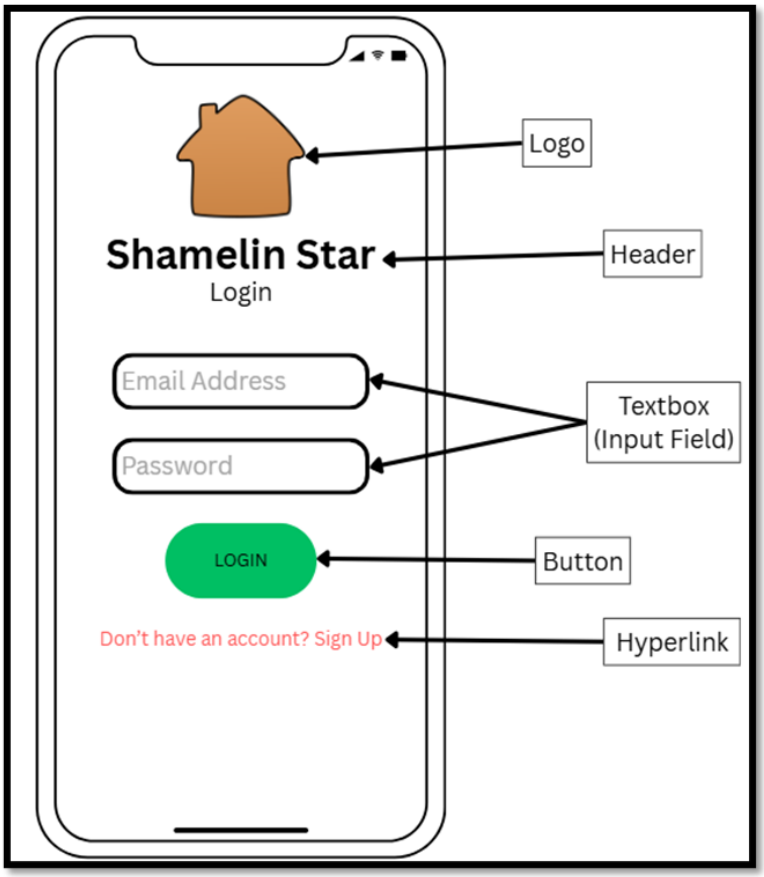


Figure 6.2.4: Wireframe Resident Login

Figure 6.2.4 For residents, Resident Login Page is their first try at correct identity. Users can input their registered email and password at this page to access the basic functions of the application. The login button will confirm whether the data you entered is in fact correct, so that only resident authorized to access this

system can enter and change information. Residents should enter into their own account quickly in order that any sensitive data such as tie-up codes or financial numbers are not made public for fear that they may subsequently get inputted and stolen by others.

6.2.2.2 Register Page

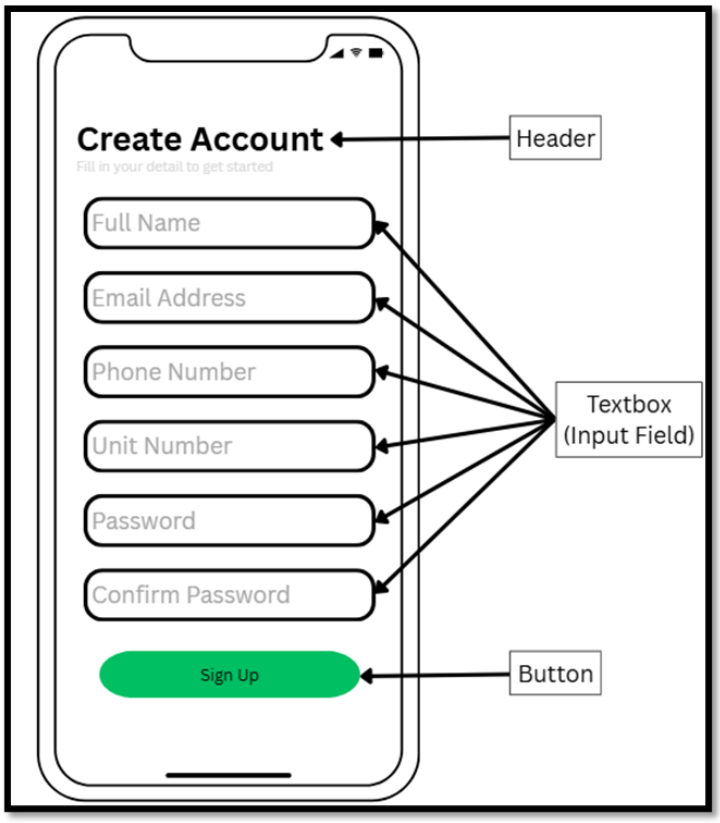


Figure 6.2.5: Wireframe Register

Figure 6.2.5 New users that want to log into Shamelin Star resident system. You need to register at the registration site of this user mobile application. This page offers input fields for residents and asks residents for their details such as email address and passwords so that they can register into this system. The registration button permits one's user data to be saved in system completion of the sign-up process automatically. It also ensures your registration data is secure and easy to get at for people, just by making a simple interface who registers users.

6.2.2.3 Home Page

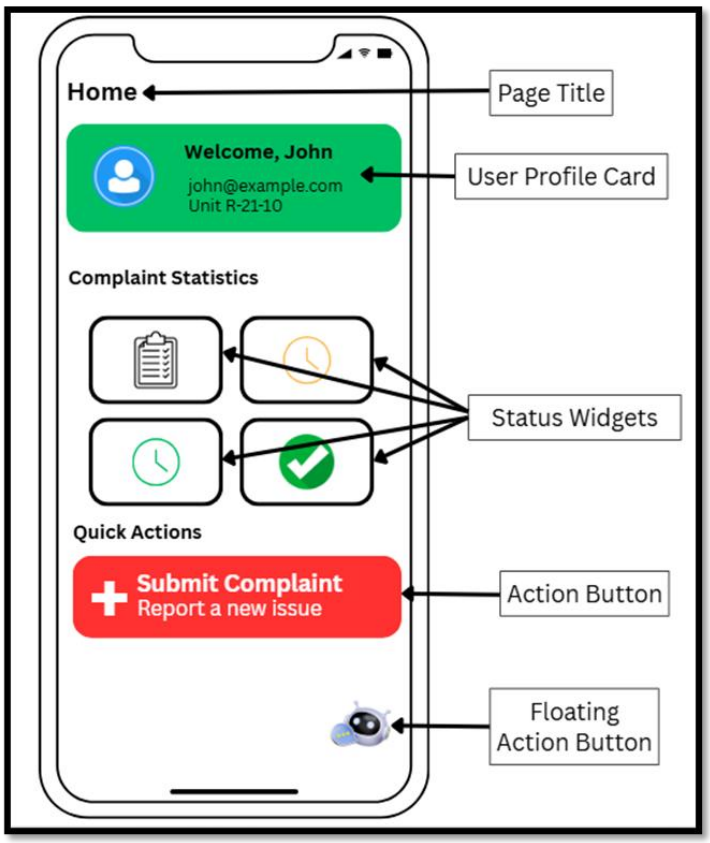


Figure 6.2.6: Wireframe Home

Figure 6.2.6 The Shamelin Star resident mobile application will drop into this Home Page (main display) upon successful login. This is the resident home page, offering an overview of the application and elementary modules such as creating new complaints as well displaying existing complaint records. The basic and recent activity information in this user account is straightforwardly presented. The core functions are directly available at Home Page, for a user-friendly interface.

6.2.2.4 Profile Page

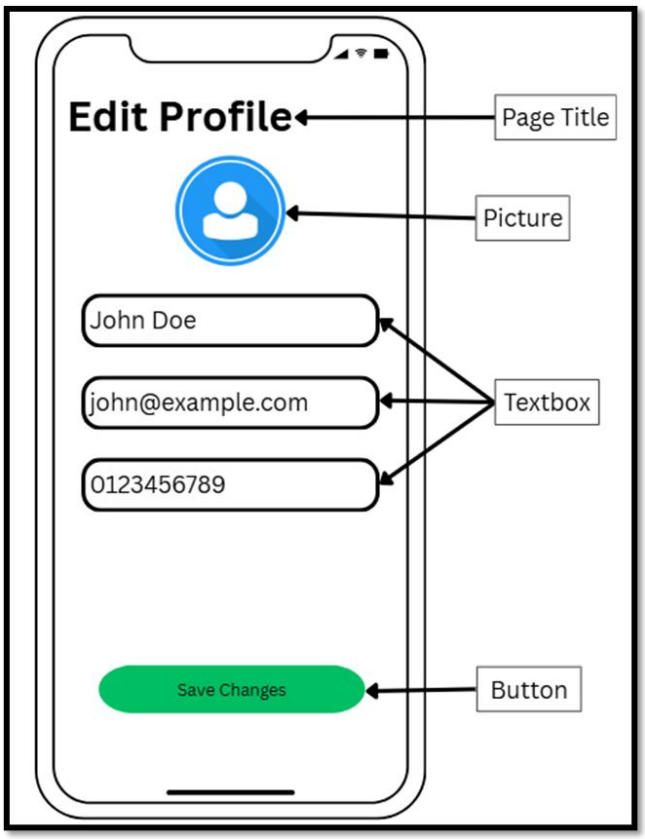


Figure 6.2.7: Wireframe Profile

Figure 6.2.7 If you're interested, the resident mobile apps Shamelin Star also has its own profile page. There users can check out and manage their personal data. It's on this page that residents can provide crucial user information (including their profile details and account settings) to help others better understand them. They can revise it as necessary. Profile Page Sparse Layout keeps everything in the right place and up-to-date while managing account settings and app security.

6.2.2.5 Submit Complaint Page

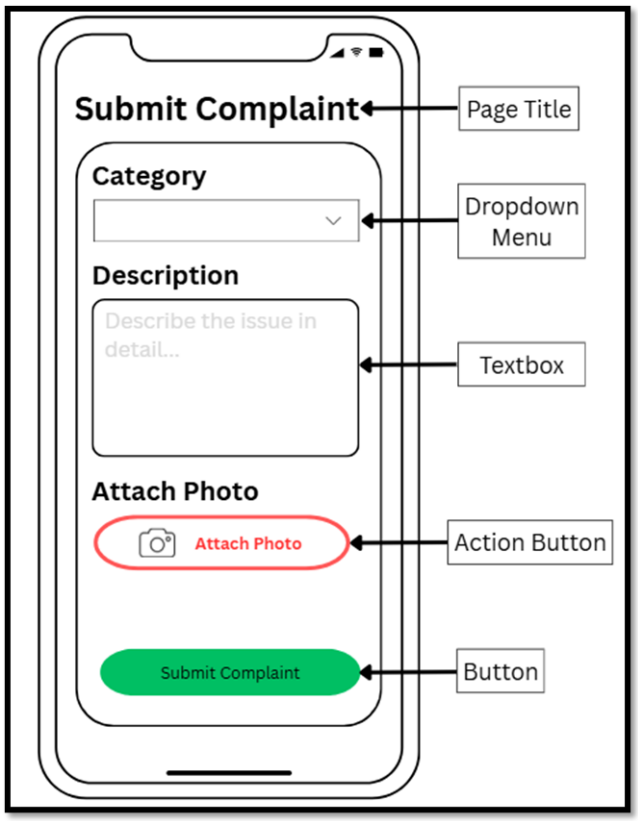


Figure 6.2.8: Wireframe Submit Complaint

Figure 6.2.8 The page Submit Complaint is the user to report any issues and where, this will put the complaint straight also receive complaints directly into management mailboxes. Choose the type of complaint from drop-down menu, then explain problem in a text box and add photo evidence if necessary. There is a submit button at the bottom of this page which sends your information into system. This page has a layout that helps to ensure complaints are clearly recorded as well as giving a simple and efficient way for residents to report them.

6.2.2.6 FAQ Chatbot Page

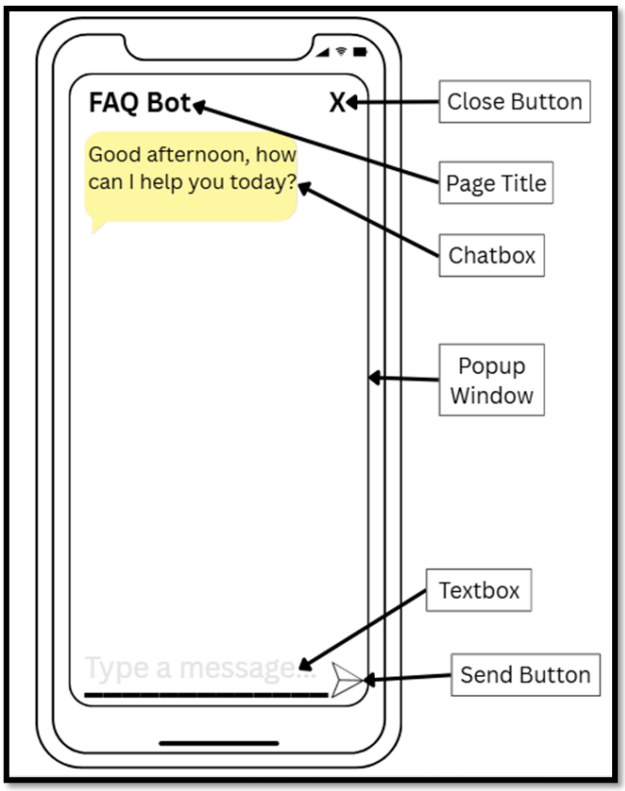


Figure 6.2.9: Wireframe FAQ Chatbot

Figure 6.2.9 the FAQ Chatbot Page of Shamelin Star resident mobile application is capable of providing users with instant assistance through automated chat interface assistant (Khan et al., 2023). It displays the clear page title and a close button so that users can easily leave from the chatbot since it pops up window. The chat area shows responses generated by the system for the user and a text box for input and send button to submit questions. By allowing residents to quickly get answers to their frequently asked questions, this chatbot interface helps provide user support without the need for manual assistance.

6.3 Database Design

In developing the Shamelin Star E-Reporting System, the database design is an important aspect (Sommerville, 2016) since efficient storage, organization and retrieval of data will be required for both resident mobile application and management web dashboard. Residents information, complaints, staff members, complaint state (Moniruzzaman & Hossain, 2013) are managed using a cloud-based NoSQL database system. A well-designed database improves system performance, enables real-time data synchronization, and preserves data consistency among various program components (Pressman & Maxim, 2020).

Different from conventional relational databases, a document-based database is used to structure the system, in which data is saved as collections and documents (Moniruzzaman & Hossain, 2013). Each

collection defines a particular entity of the system like Users, Complaints and Staff records. Thanks to this flexible structure, the application can process dynamic data and real-time updates continuously so that complaints entered in the mobile application instantly show up on the management dashboard (Moroney, 2020). The database is designed for scalability and maintainability with the result that the system can incrementally expand without a major reorganization. (Sadalage & Fowler 2012)

Database design concerns security and integrity of data. Authentication mechanisms allow only authorized users to access system data while role-based access lets local residents and administrative personnel each play their respective roles. (Laudon & Laudon 2020) Each screen that the user interacts with has a suggested behavioral pattern of its own based on the data. But if you look at everything that Mock and others are saying essentially I found the concept of geometric algorithms very useful in connecting different accounts across platforms, and building behaviors to build paths at certain events from one medium to the other for users themselves travelling forward between different sorts mobile and desktop electronically, or anywhere else people need us.They are also talking about something called Two Way Data Binding (where when one model or view updates, it could make other models or views also update if done correctly) indeed, well supported responsive platforms depend on these two processes working in harmony with each other.

6.3.1 Data Dictionary

This Data Dictionary is structured to describe the primary data entities and attributes in the Shamelin Star E-Reporting System. It offers a comprehensive view of data fields, their functions, and connections within the database, assisting developers and stakeholders in comprehending how data is organized (Sommerville, 2016).

User accounts, complaints, and staff information are among the key collections within the system. The filed complaint documents include fields for category, description, status, images and timestamps. According to (Pressman & Maxim, 2020), the data dictionary is an additional resource that increases clarity and consistency in implementation of databases, minimizes risks of redundancy, as well as facilitates easy maintenance of existing applications and upgrade to future system. A data dictionary clearly defines the structure and meaning of its data elements which provides reliable management for data all throughout system lifecycle (Harrington, 2016).

6.3.1.1 Complaints Collection

Table 6.3.1: Data Dictionary of Collection “complaints”

Field Name	Data Type	Description
assignedDate	timestamp	Date when complaint was assigned to staff.
assignedTo	string	Staff member responsible for handling complaint.
assignmentNotes	string	Additional notes related to assignment.
category	string	Complaint category selected by resident.
complaintId	string	Unique identifier for each complaint.
complaintType	string	Type classification of complaint.
complaintDate	string	Date when complaint occurred or reported.
description	string	Detailed explanation of the issue.
email	string	Resident email address.
imageBased64	string	Image attached to complaint stored in Base64 format.
location	string	Location related to complaint.
phoneNumber	string	Resident contact number.
priority	string	Complaint priority level.
residentName	string	Name of the resident submitting complaint.
stability	number	Numeric value representing complaint stability or severity.
status	string	Current compliants status (Pending, In Progress, Submitted).
submittedDate	timestamp	Date complaint was submitted.
unitNumber	string	Resident unit or apartment number.
updateNotes	string	Notes added during complaint updates.
userId	string	Reference to resident user ID.

Table 6.3.1.1 Our complaints collection table, namely stores all complaint related information raised by the residents in form of mobile application. Each document represents a complaint, which is described by category, description, status, staff that is handling it and also the image data if any. So that this framework, would let mobile application, and management dashboard to synchronise in real-time for tracking and managing complaint.

6.3.1.2 Settings Collection

Table 6.3.2: Data Dictionary of Collection “settings”.

Field Name	Data Type	Description
lastNumber	number	Stores the latest generated complaint number for reference or auto-increment logic.

Table 6.3.1.2 settings collection contains system configuration data used to support application functionality. In this system, it maintains numbering information used for generating unique complaint identifiers.

6.3.1.3 Staff Collection

Table 6.3.3: Data Dictionary of Collection “staff”.

Field Name	Data Type	Description
authUid	string	Authentication user ID from Firebase Auth.
createdAt	timestamp	Date staff account was created.
email	string	Staff email address.
mustChangePassword	boolean	Indicates whether password change is required.
name	string	Staff member full name.
passwordChangedAt	string	Timestamp of last password change.
phone	string	Staff contact number.
specialty	string	Staff specialization or role.
status	string	Staff account status (Active/Inactive).
updatedAt	timestamp	Last update timestamp.

Table 6.3.1.3 staff collection stores maintenance staff information used by administrators to assign and manage complaint handling tasks. Each document represents a staff member and includes authentication and contact information.

6.3.1.4 Users Collection

Table 6.3.4: Data Dictionary of Collection “users”.

Field Name	Data Type	Description
createdAt	timestamp	Date user account was created.
displayName	string	Display name shown in application.
email	string	User email address.
fullName	string	Resident full name.
phoneNumber	string	Resident contact number.
profileImageBased64	string	Profile image stored in Base64 format.
role	string	User role.
unitDisplay	string	Unit display label.
unitNumber	string	Resident unit number.
userID	string	Unique identifier for user account.

Table 6.3.1.4 users collection stores resident account information used for authentication and complaint submission. It supports role-based access control and links residents to their submitted complaints.

6.3.2 Data Flow Diagram (DFD)

6.3.2.1 Data Flow Diagram

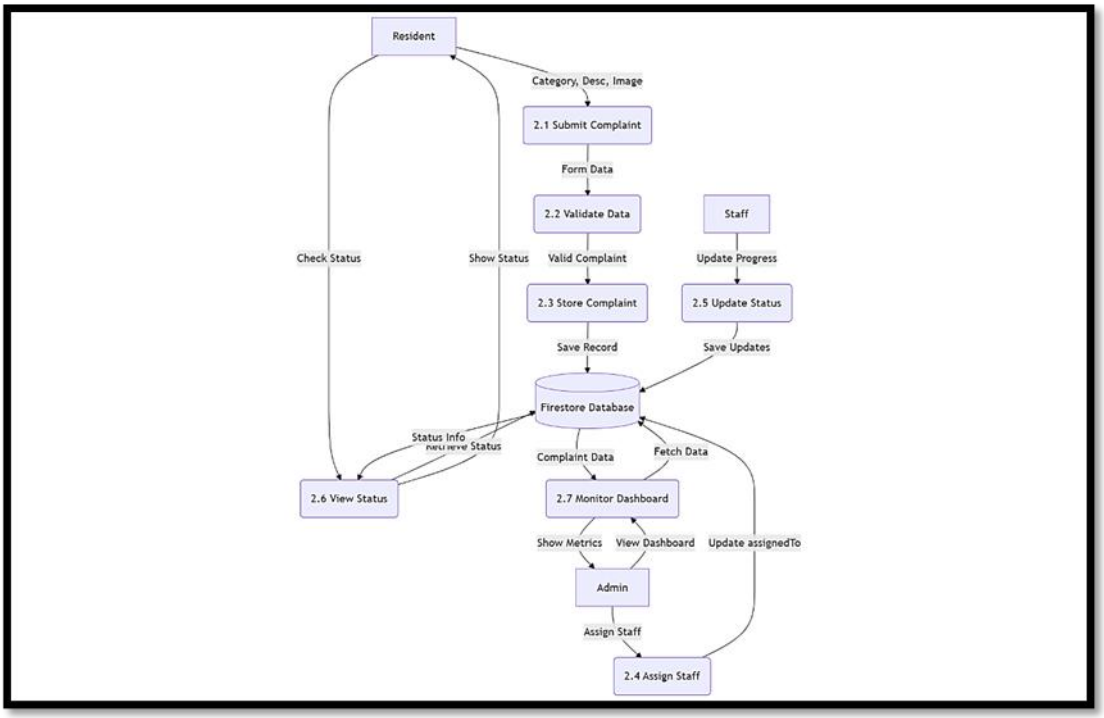


Figure 6.3.1: Data Flow Diagram

Figure 6.3.1 starts by introducing its Complaint Management process. It encompasses the essential functions of the Shamelin Star E-Reporting System: When residents use the mobile App to have an idea, theymust write and then upload their complaint of said nature. Entry from the public is checked and stored to Firestore. Since some complaint may prove unfeasible, no point in this interregnum for participants: After receiving a service request, administrators can see all coming complaints. Depending on specialization or importance level, they assign these to maintenance people who then enter their response into the complaints DB management tool on their terminals even while work is being done out in field. Such a structured workflow guarantees the proper disposition of any Report or Complaint, real-time monitoring and better communication between residents and administration teams.

6.3.3 Entity Relationship Diagram (ERD)

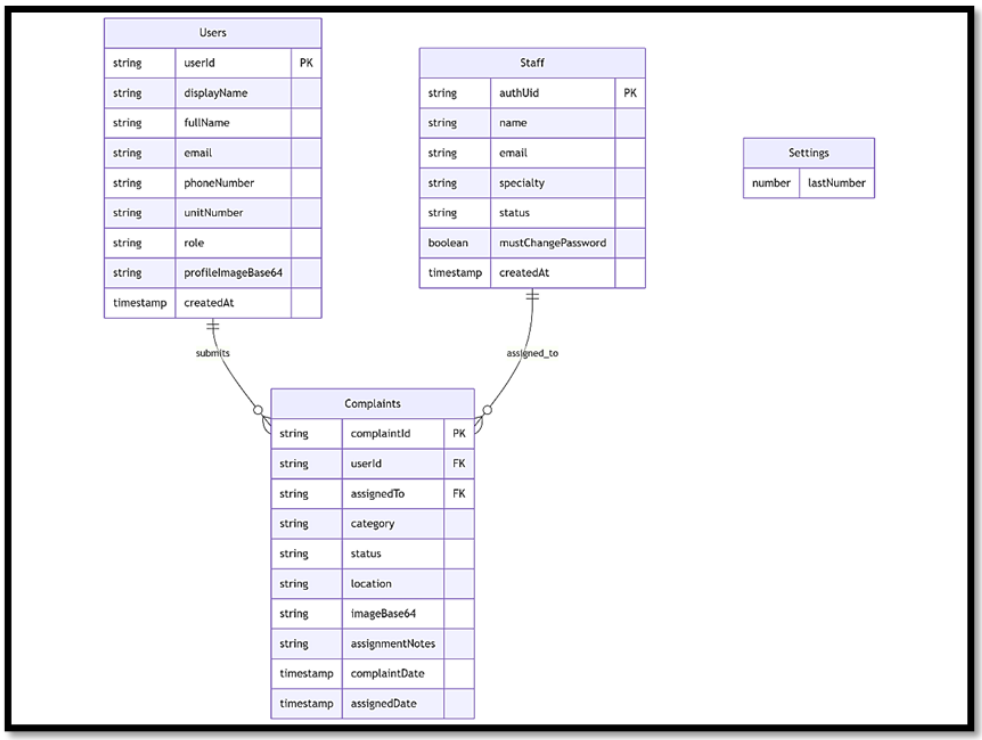


Figure 6.3.2: (ERD) for the Shamelin Star E-Reporting System

Figure 6.3.2 illustrates how we have organized and related the structured in Firebase Firestore. The complaints collection is both resident and with staff the centrepiece of this whole people system. Each comment entity records categories inhabitants send in via mobile descriptions of problems along with urgency Resident details The users collection stands for those residents who submit complaints through the mobile app, while staff collection maintains records for all management and maintenance personnel responsible for managing and dealing with reported problems. Relationships are created logically by means of identifying numbers such as userId and assignedTo. This makes it possible to connect a specific complaint database with this resident or these members of staff.

Support for operational and management of the system as a whole is also evident from the ERD. The settings collection contains code information such as system configuration of serial numbers and parameters measures that could be used in ensuring real-time collection accountability. The ERD provides clear definitions for these relationships so that data integrity can be maintained and transmitted between mobile application on one hand and web-based management dashboard. The inability of Firestore, a NoSQL database to enforce foreign key relationships did not prevent logical relationships from being maintained with reference fields such as an interface with the complainant or staff member finally. This arrangement enables the system components to be queried effectively, expanded and synchronized in real time.In brief, the ERD is a graphic representation that demonstrates the structuring of the system. It shows how complaint management, user determinants and staff assignation work together to facilitate digital reporting and administrative oversight.

6.4 Flow of the System

This passage provides an overview of the E-Reporting System at Shamelin Star. It describes how users enter an application and then the information they input is processed, saved in various repositories such as Point-to-Point Protocol (PPP) or Distributed File Systems (DFS), and eventually displayed on administration dashboard screens (Sommerville, 2016). Through Firebase services, the system consists of a resident-facing mobile application and an administrative web console which facilitate real-time communication and data synchronization (Moroney, 2020). Design artefacts such as the Data Flow Diagram (DFD), Entity Relationship Diagram (ERD) and database structure help to understand how data flows between system components as they would during actual runtime operations (Dennis, Wixom & Roth, 2015).

6.4.1 Web Management Flow

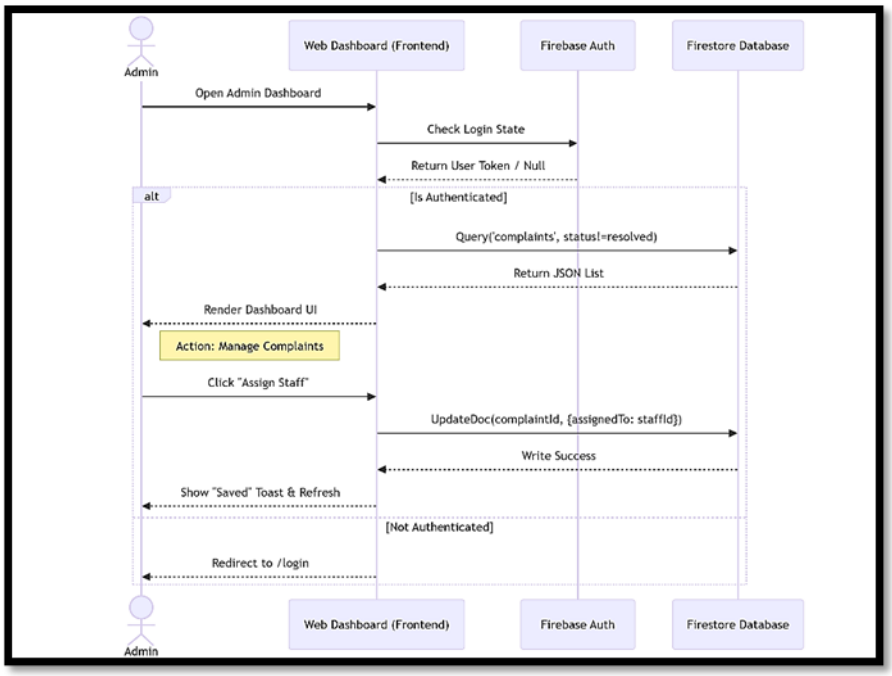


Figure 6.4.1: Web Management flow for Admin Dasboard

Figure 6.4.1 This module is for the administration staff to monitor and manage residents complaints through a web based. Management System as shown in figure below. The journey starts with the admin and when he visits the web dashboard, the system checks if his authentication session is active using Firebase Authentication. After verifying, it gets the complaint data from the Firestore database and displays this data on dashboard interface which include status of complaint, resident information, assigned staff and many more.

Management actions can be taken by Administrators to update complaint status, assign staff members as well as adding notes regarding the complaint. These updates are being stored in the Firestore Database mere seconds later, ensuring data consistency and real-time synchronization. It also provides safe access control with auto-refresh and information has already been updated on screen, while the dashboard gives full visibility for all complaint activities made by administrators.

6.4.2 Mobile Application Flow

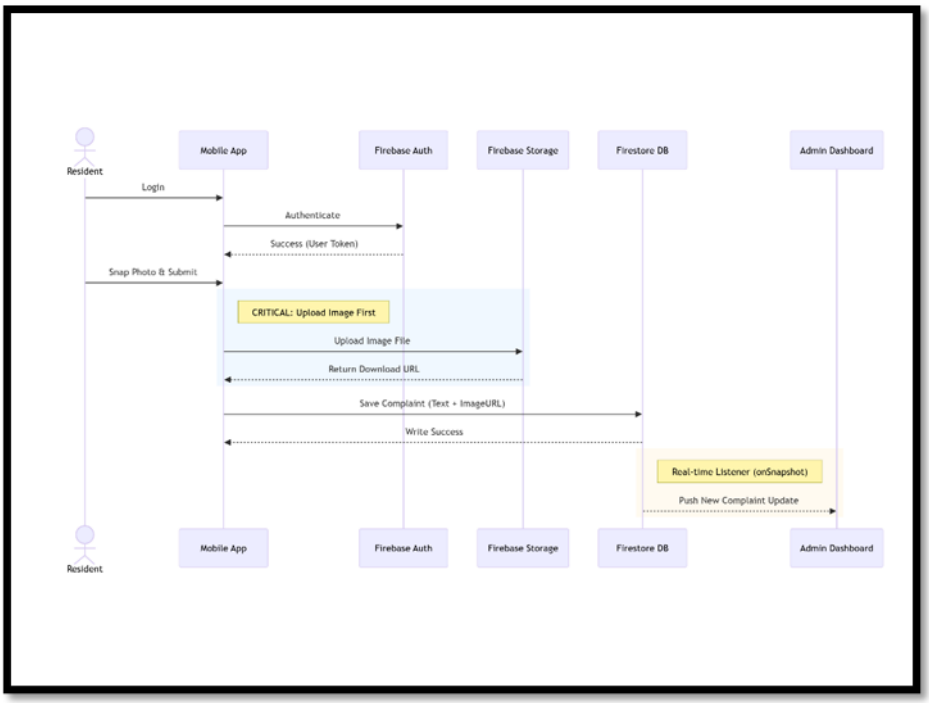


Figure 6.4.2: Mobile Application Flow

Figure 6.4.2 After residents log in to the mobile application, the authentication checks are solidified via Firebase Authentication services. For instance, in a successful login, users need to fill in the details of their complaint: The category that has its own grievance people report - description location where it happened and an image attachment in base64 size for uploading images onto web pages.

When submitted, this complaint data will be stored directly into Cloud Firestore database where it is then channeled into conversations between the mobile application and an administrator interface (dashboard). Real time chat of the brand. The administrator dashboard then displays up-to-date data in real-time, making it possible for management to see new submissions as soon they are made. Residents may communicate with administrators easily in this design. Besides allowing at no cost for data to flow between people easily while keeping a system responsive and improved usability overall.

6.4.3 POS System Flow

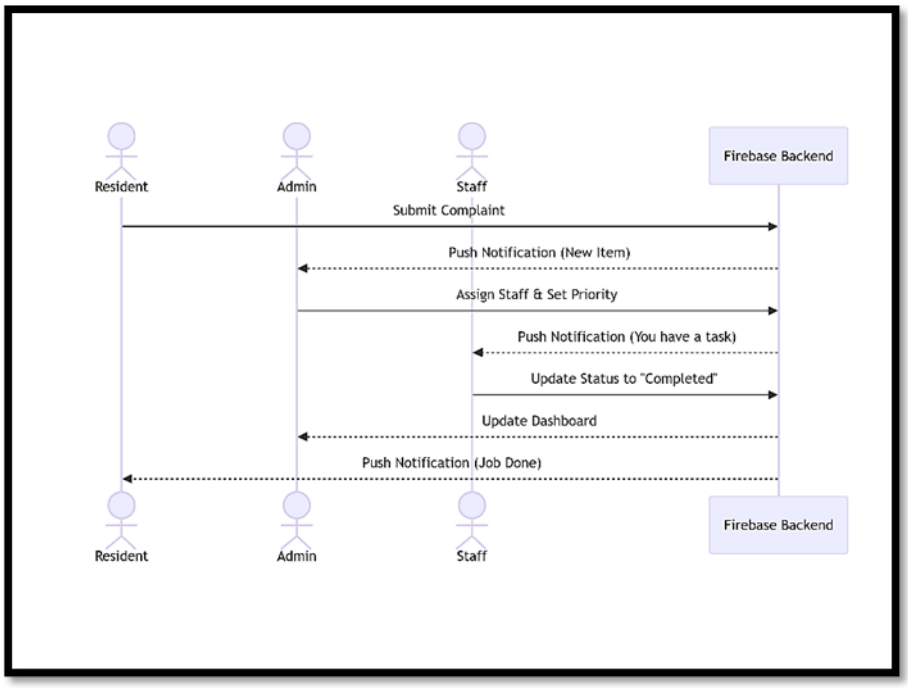


Figure 6.4.3: POS System Flow for Shamelin Star E-Reporting System

Figure 6.4.3 The setup of handling complaints by means of the Shamelin Star E-Reporting System is explained.Describes the workflow progress of managing complaints.Work order board that sends urgent and newly reported complaints to the most appropriate person for attention. Also, there are feature requests from staff which will be sent through this feature.

People with assignments can update the complaint operational solution progress is also updated to record whether a particular operation is defined or not into this system. All update are synchronised back-end so that the progress for residents and maintenance is in real time.On the basis of a structured workflow, through it comes better co-ordination between residents and management.

6.5 Conclusion

With this chapter as an introduction, I have finally presented the frontend of the Shamelin Star E-Reporting System to you. It looks at how system architecture works in detail. And completes most of its definitions. In particular. We have defined what proof mountains are, How interface designer is exactly prepared. Then project inquiry token created takes shape and is shown. Finally Project status tokens are defined, as parts of course. All whilst trying to allow users of different roles for optimal experience in process execution. This integrated, role-based system design includes both the web management dashboard and resident mobile application.

Primary design artifacts: database structure, data dictionary, DFD & ERD represent how complaint data, user information, staff assignments and system configurations get represented and processed in the Firestore. We've created an interface which is user-friendly, simple and secure through authentication. Both

admins-members and residents can easily do their jobs. Flow diagram below sketches out how data flows across operating systems. From mobile app to web back end to service end Well, these tactics have been quite successful. Due to this integrated configuration, matters are documented with much less time lag than before. And at the same time you not only get immediate feedback but people on site can also participate. In Conclusion, the system design proposed creates a scalable yet maintainable structure able to meet the functional requirements of the project.

With the precise workflows complaint creation flow, complaint & user staff assignments (including user) flow, event updates (including user flow information that is required to come into the app for them), supervision will be given throughout this system integrated entirety by administrative level Elsewhere. Future improvements, such as advanced analysis or an intelligent answering machine for questions about operations history, can be developed from the design foundation. This will be followed by subsequent step implementation, testing and deployment.

7 IMPLEMENTATION

7.1 Introduction

Shamelin Star E-Reporting System set the process in this chapter with the guiding system design finalized earlier on from previous stages. This System used a hybrid-based solution, which employed a web management dashboard admin page for the collaborative region and an Android mobile app for its citizens (Sommerville, 2016). The implementation combines contemporary Web and mobile technologies with backend services that rely on the cloud for their real-time data replication or seamless operation of this system (Pressman & Maxim, 2020).

Combined with the implementation requirements, a client-side management dashboard was designed with HTML5, CSS3 and JavaScript in ActionScript (Duckett, 2014). The work used Firebase Firestore as the primary cloud database, allowing for real-time updates to be pushed from a resident mobile application side dashboard (Moroney, 2015). When an application complaint form is submitted, we can see down on the administrative end too at once without any manual refresh or background (Sadalage et al., 2012).

The citizen's mobile app was developed primarily in Android Studio using Kotlin and/or Java for its programming language and XML to build interface layout (Phillips, Hardy & Stewart, 2019). F ire base services provide user authentication, cloud data storage and secure communication between the application itself and its back-end database (Moroney, 2015). This Cloud-based architecture, which not only does not require the conventional server-side infrastructure but also greatly simplifies System deployment complexity under this model and yet still maintains its high scalability and reliability levels (Pressman & Maxim, 2020).

All development and testing was completed in a Windows 8 environment with Microsoft Visual Studio for creating Web apps and Android Studio for creating mobile ones. This ensured that both residents and managers can easily use the System. Thanks to modular programming, real-time online connectivity and easily navigated User interfaces throughout each link in development process, we took this line of thinking from first concept to last (Sommerville, 2016).

7.2 Execution Platform

7.2.1 Windows 11

Figure 7.2.1: Windows 11

Figure 7.2.1 The Shamelin Star E-Reporting System is developed and tested using the Windows 11 operating system, It was selected with 64-bit as we claimed earlier that it is stable, modern developing tools are compatible and providing us best security. The platform enables seamless web dashboard development and Android Studio for mobile application development, allowing both environments to run at the same time without compromising performance. The web interface was tested using modern browsers (Microsoft Edge) to debug JavaScript functions and verify the real-time Firebase database updates. With the use of Android Studio's embedded Android Emulator, it was possible to run tests across various mobile device configurations without having physical devices. Apart from this, Firebase cloud services were also used to handle authentication, performing operations in Firestore database and synchronizing the data in real time between mobile application and management dashboard through this environment. Windows 11 acted as a stable and efficient environment for coding, testing, and implementation.

7.3 Implementation Tools

7.3.1 Visual Studio Code

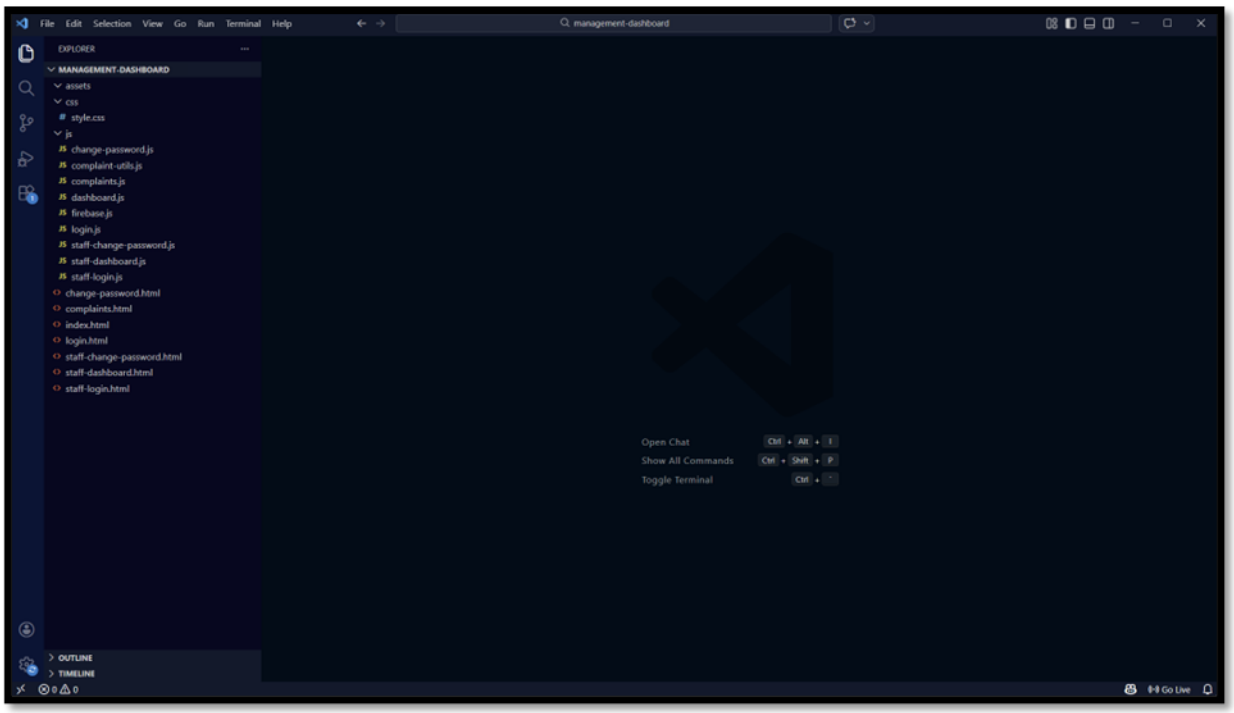


Figure 7.3.1: Visual Studio Code

Figure 7.3.1 The screenshot of Visual Studio Code on developing Shamelin Star E-Reporting System. Visual Studio Code (IDE) Visual Studio code was used as the main Integrated Development Environment (IDE) for building the web based management dashboard with HTML, CSS and JavaScript owing to its low weight performance and extensive extensions support (Visual Studio Code, 2024).

7.3.2 HTML

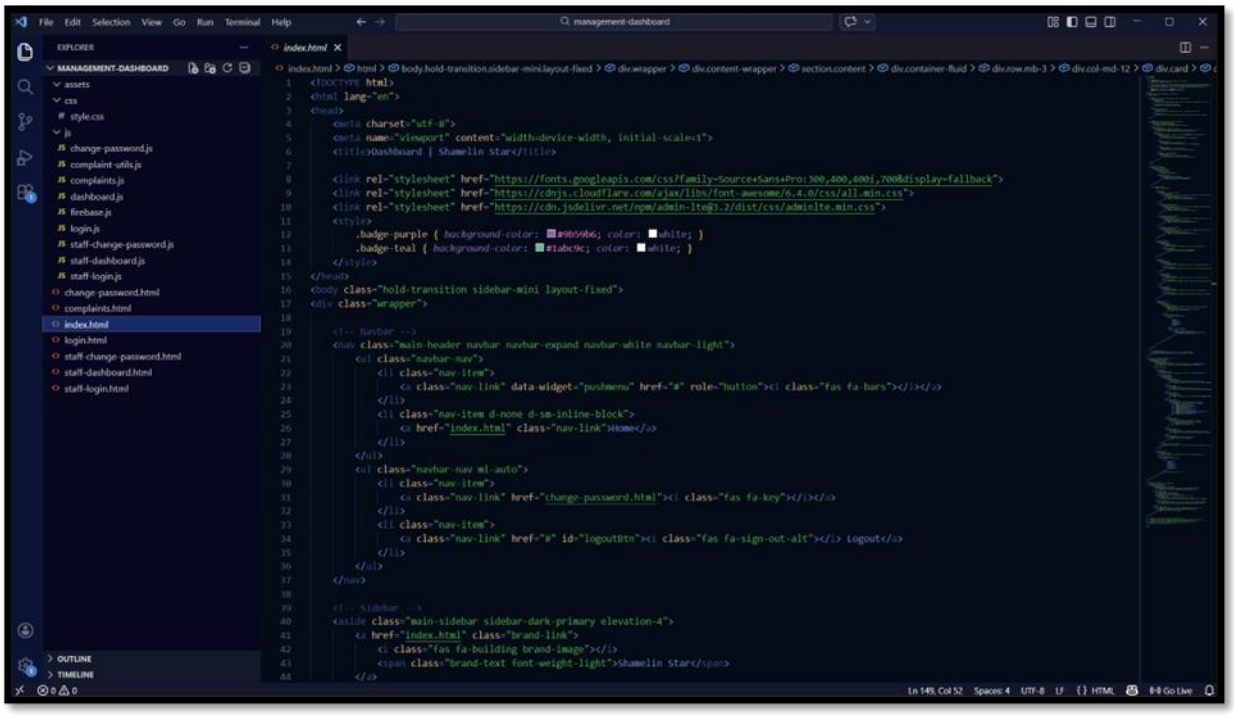


Figure 7.3.2: HTML

Figure 7.3.2 Therefore, the basic layout forms, navigation menus, tables, and content sections used HTML to construct. The management dashboard embedded in a web page with ant technology also combines all these elements by making them one. It's the primary shanghai-steel of user interface logic components and guarantees interoperable conditions across different browsers and equipment. Based on well-structured semantic HTML elements, combined with Bootstrap components that take into account usability and accessibility requirements for both management staff as well as residents alike.

7.3.3 JavaScript

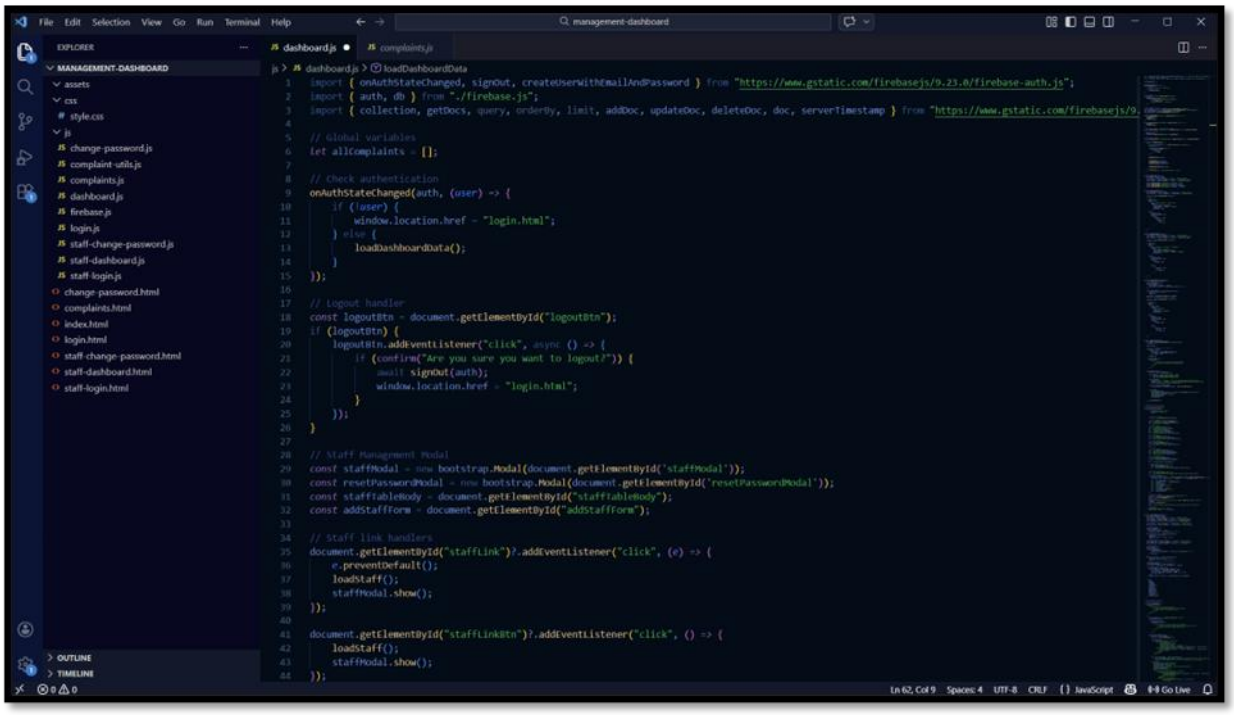


Figure 7.3.3: JavaScript

Figure 7.3.3 In comparison to most other languages with different design paradigms, the author programmed this entire course around JavaScript and Node JS only using HTML as a display platform. And for several years before it was picked up by colleges who wanted freshmen programming boot channels which featured its real-time updating capabilities accurate to within milliseconds. The heart of any project or work is always its interaction with users, and this is no exception. The business dialog will help you understand how to design a good UI that avoids or reduces language barriers. It manages complaints by asynchronous mechanism, updating their status and executing user sessions, all without any Page refreshing. This enhances the system's responsiveness and user-experience. At the same time ES modules use modularization, which can make code more organized and easier to provide overall system maintenance for future development through debugging and renovation efforts anywhere else that it may be needed or come about from new applications or added features within an existing one.

7.3.4 CSS

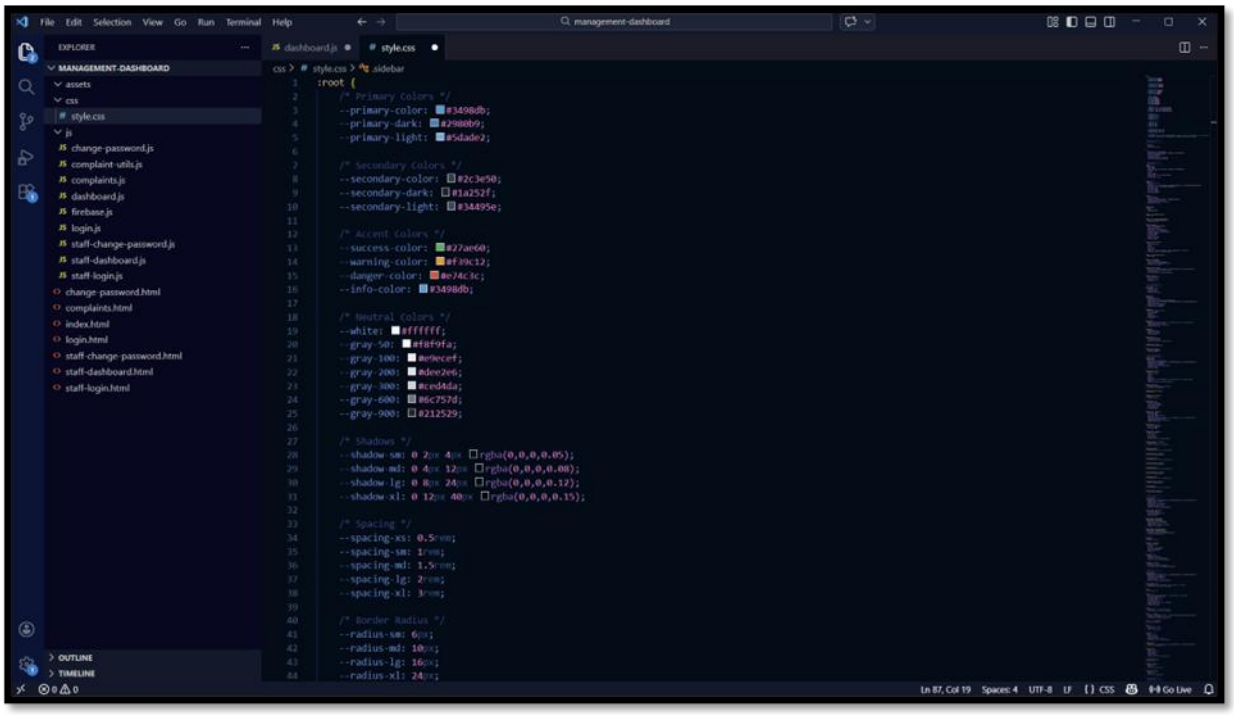


Figure 7.3.4: CSS

Figure 7.3.4 CSS was used for a clean and responsive interface. It appeared the same on all screens. In order to give the interface a user-friendly experience, it is used in conjunction with custom styles: Bootstrap components and custom styles combine to form a consistent layout, modern colour scheme, better overall readability. The layout also supports responsive design principles, so both the web dashboard and login pages are both usable and visually consistent to make for a friendly Web professional interface.

7.3.5 Android Studio

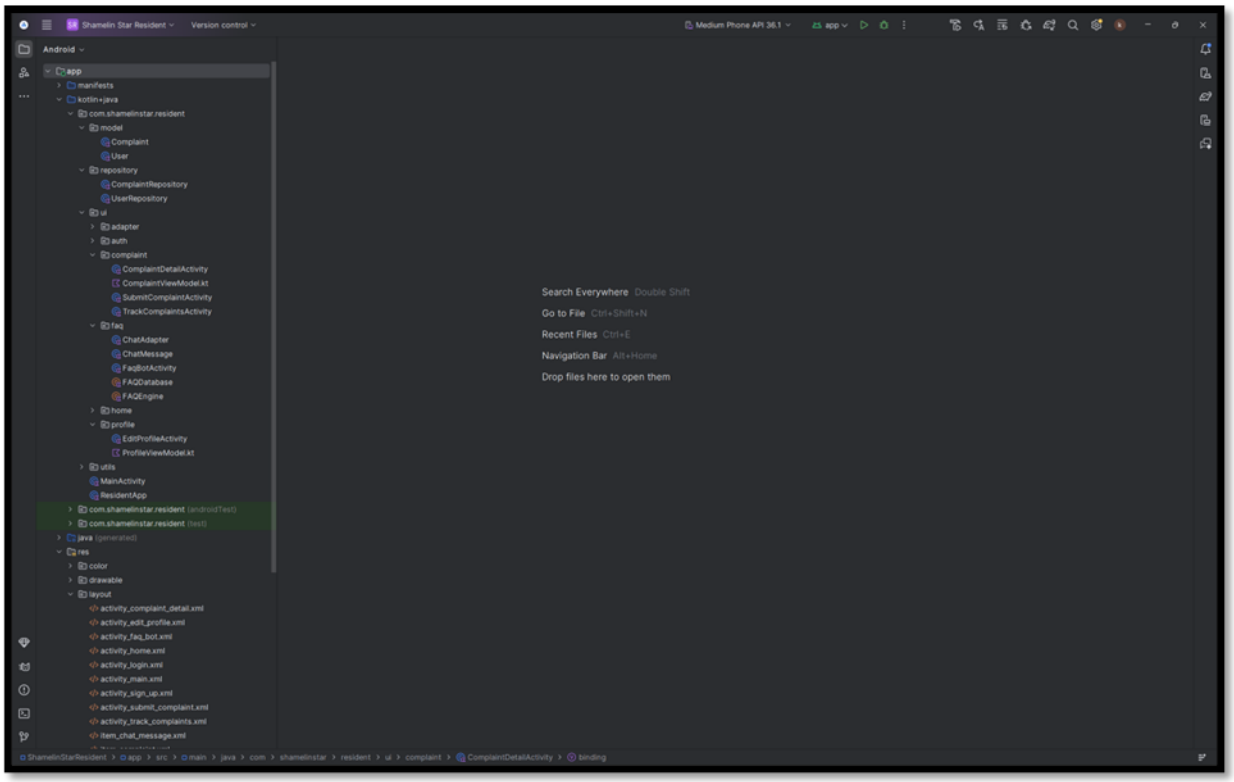


Figure 7.3.5: Android Studio

Figure 7.3.5 Android Studio environment for the development of the mobile application as part of the resident mobile application service for the Shamelin Star E-Reporting System. The Official Integrated Development Environment (IDE) for the development of application logic using Kotlin/Java or user interface designing from XML layout files is Android Studio. It includes critical components like the Android Emulator, layout editor, and debugging console that allowed for rapid testing and troubleshooting throughout the development process. Integration with Firebase SDK using Android Studio helped in authentication, real-time communication to the Firestore database and smooth syncing between mobile application and web management dashboard.

7.3.6 Kotlin

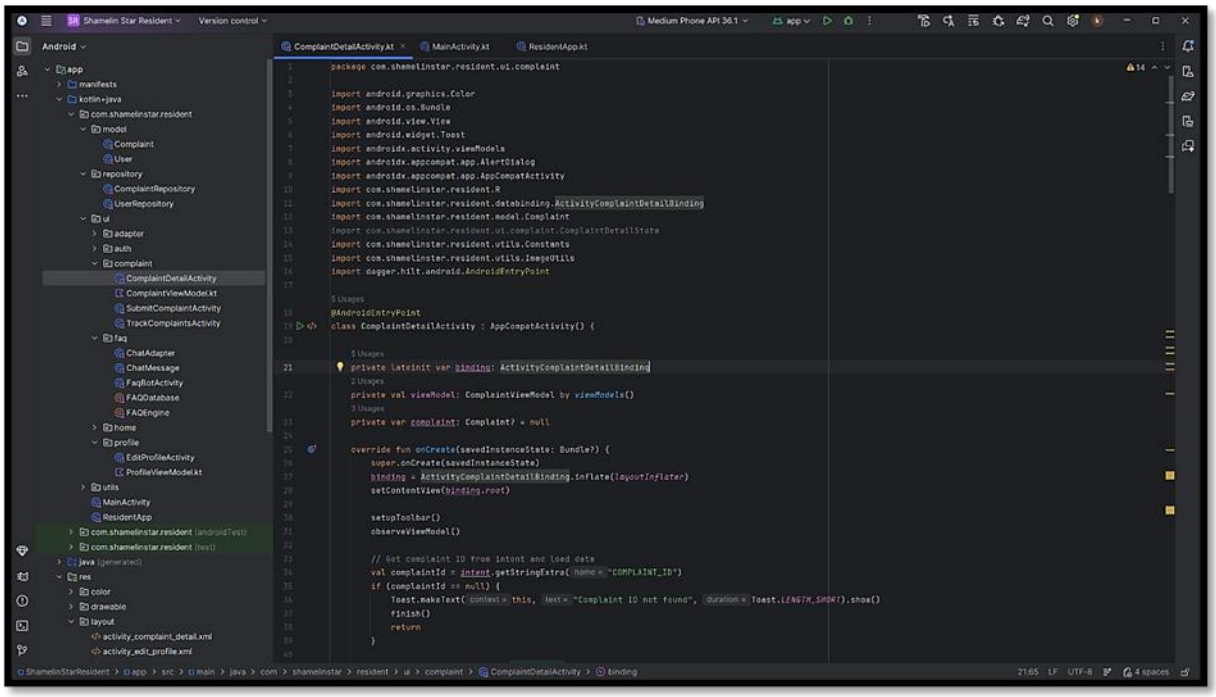


Figure 7.3.6: Kotlin

Figure 7.3.6 The Resident Mobile Application of Shamelin Star E-Reporting System is developed using Kotlin programming language. The application logic, user interactions, activity navigation and Firebase Authentication and Firestore database services were implemented using Kotlin. Its expressive syntax, null-safety features reduce prevent runtime errors which makes code readable. Kotlin and Android are being used in new Android projects as it supports contemporary Android development patterns. Real-time submission of complaints and data retrieval functionality quickly implemented in Android environment through Kotlin.

7.3.7 XML

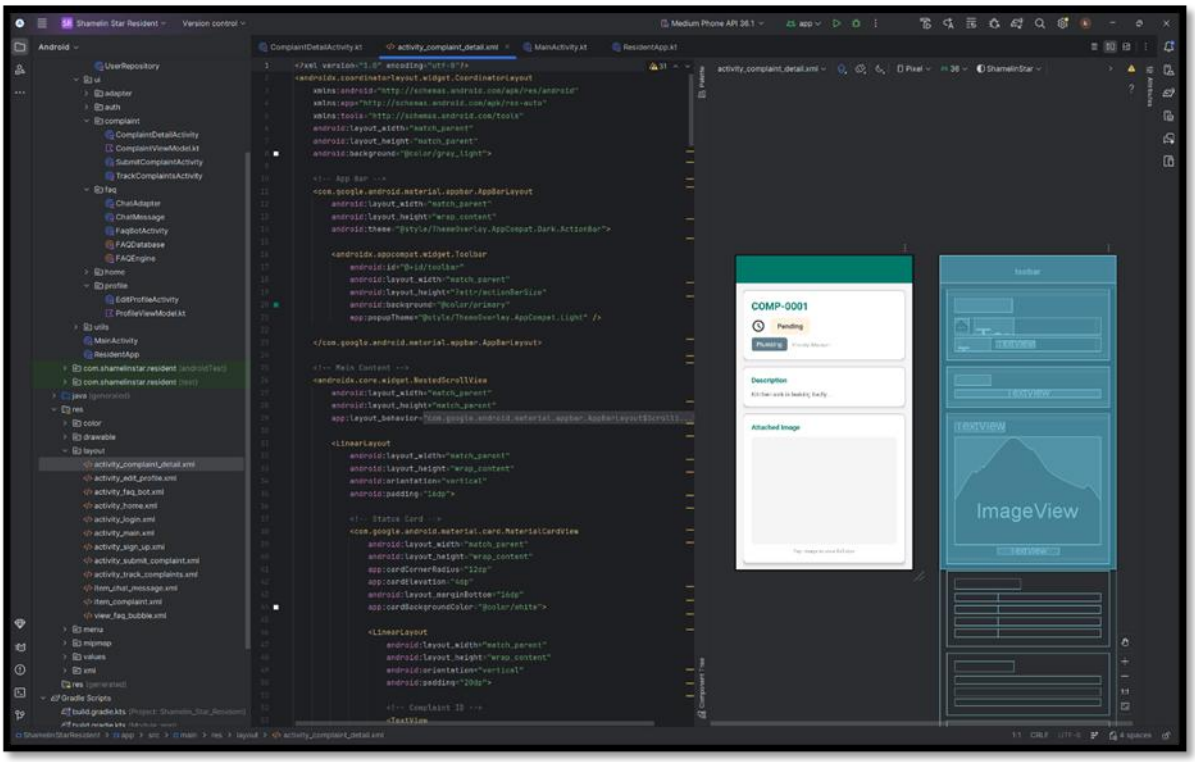


Figure 7.3.7: XML

Figure 7.3.7 Use of XML in designing user interfaces layouts for the resident Mobile Application (Shamelin Star E-Reporting System) It utilized XML in Android Studio to define the arrangement and structure of interface elements, such as text fields, buttons, image files, and navigation factors. This separation enhances code organization and maintainability, as it decouples user interface design from application logic. XML layout files utilize XML elements and attributes to define the placement of UI components on the screen, allowing for a clean separation of design from implementation Compared to other formats such as Java, XML serves better in scalability and maintainability with large projects with multiple designers working simultaneously. This modularity allows designers/developers to work on individual screens without affecting each other's changes Additionally, XML layout files enable responsive design by determining flexible constraints and layouts that translate accurately across different Android device screen sizes.

7.3.8 Firebase (Firestore)

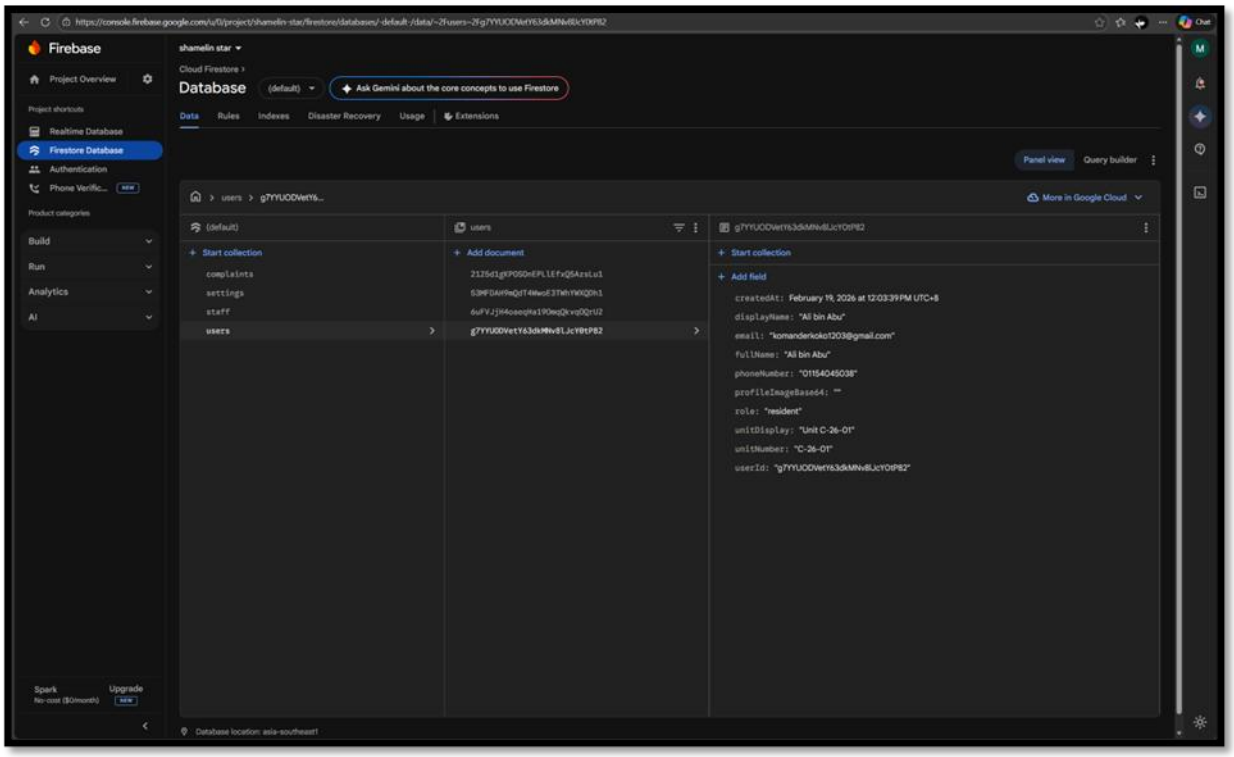


Figure 7.3.8: Firebase (Firestore)

Figure 7.3.8 The cloud database shown is the Firebase Firestore console used for the Shamelin Star E-Reporting System, Implementation Firestore (a NoSQL document-based database) was used to store and manage the collections for users, staff, complaints, and settings. Supported data synchronization in a real-time manner, no need to refresh the web-based management dashboard/mobile application complaint submission appears automatically. You can also use Firebase Authentication and security rules to have access control, so only authorized users can read or change system data. Cloud-based architecture requires no handling of a server yet also provides scalability, reliability and effective data management.

7.4 System Interface

7.4.1 Admin Login Page

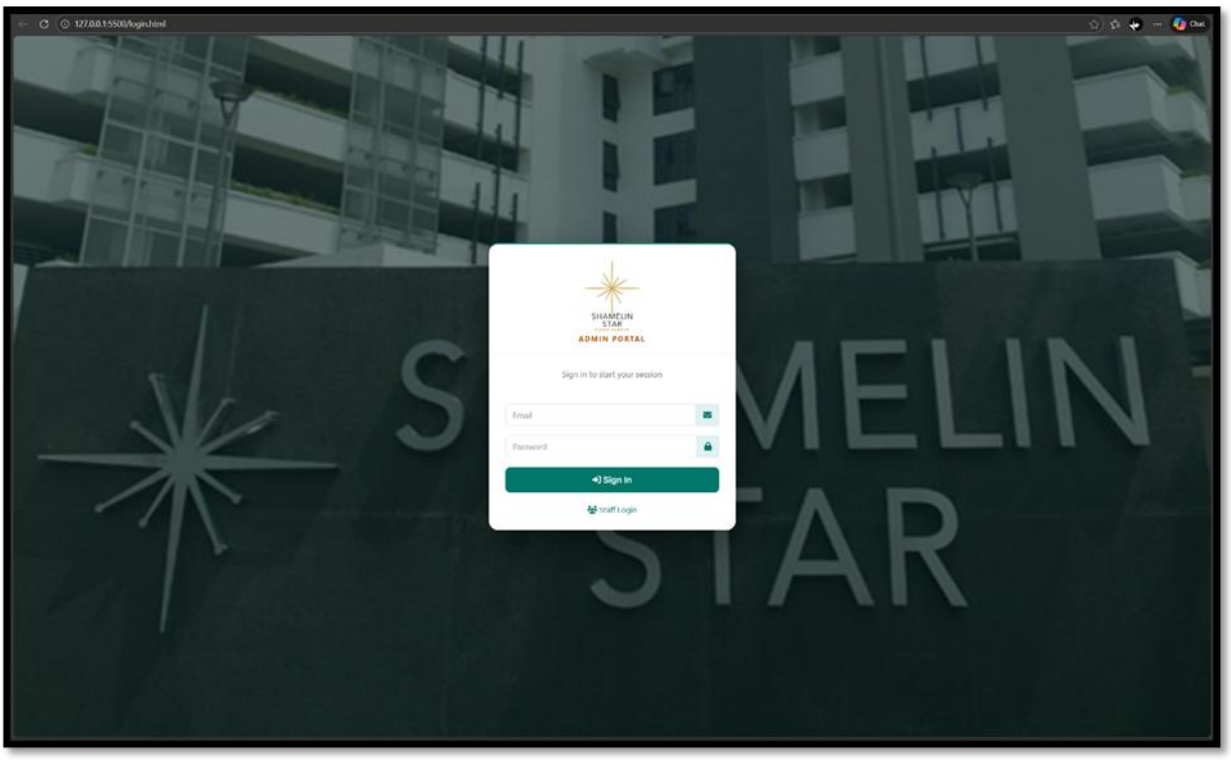


Figure 7.4.1: Admin login page

Figure 7.4.1 Shamelin Star E-Reporting System's Admin Login Page. This web based dashboard can be accessed by authorized management users given their login details. To avoid unauthorized access, an email and password input area has been provided. Firebase Authentication is used to verify the identity of these users before they can even enter the system. It provides the login screen for accessing the admin module and prohibits those not authorized from viewing complaint details of a confidential nature.

7.4.2 Admin Dashboard Page

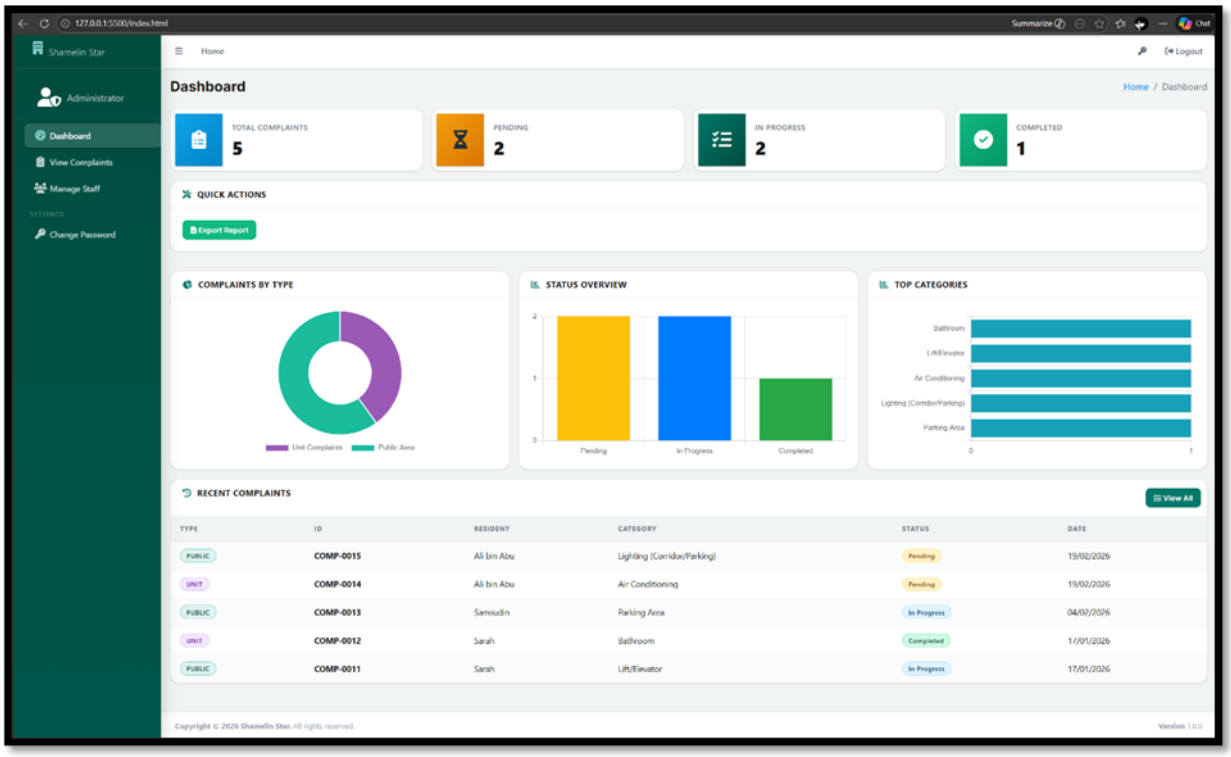


Figure 7.4.2: Admin dashboard

Figure 7.4.2 Statistics of complaints by Dashboard Dashboard display image. It includes the following three statistics on (complaints) the total number Staggered, Pending, and Completed. This allows administrators not only could retrieve complaint records quickly, but also just monitor system activity gently. Interface design adopts Bootstrap framework for clean and responsive layout. When new complaints are received `Firestore for Real-time Updates '' is used to feed this information back to the dashboard..

7.4.3 View All Complaints Page

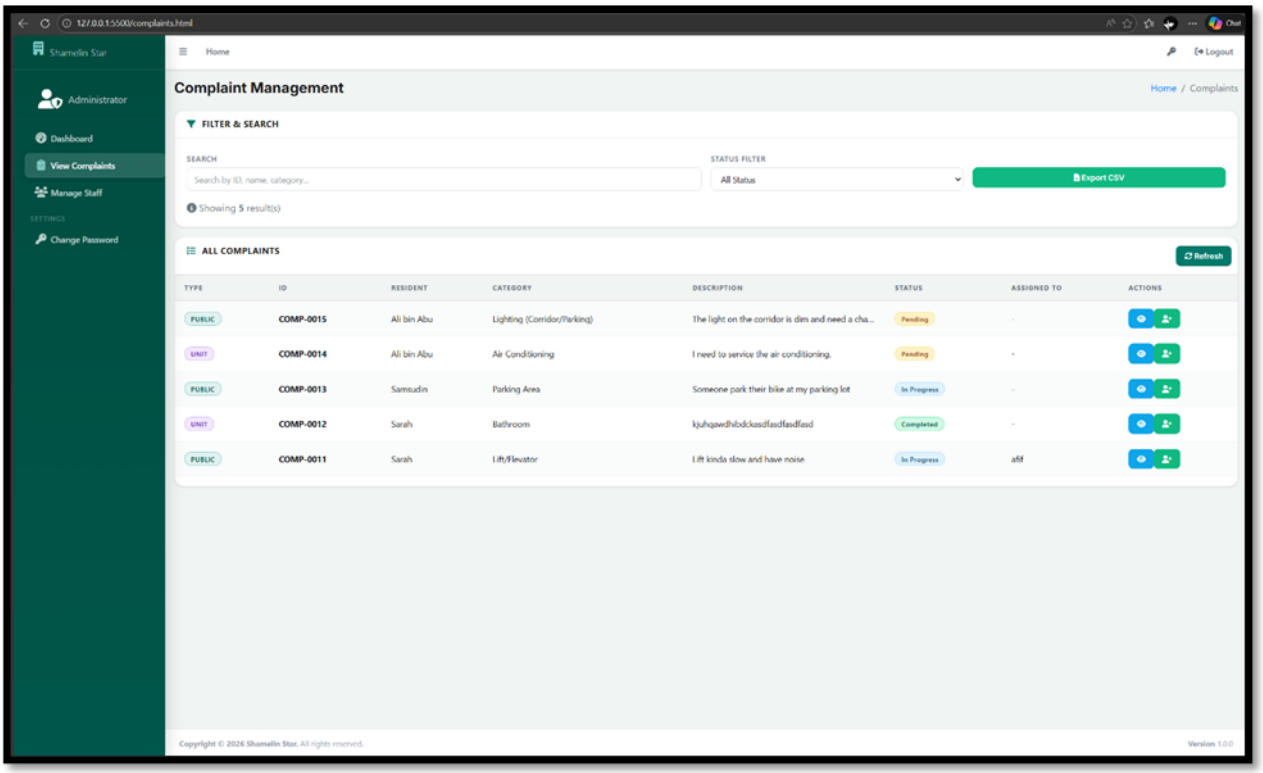


Figure 7.4.3: View all complaints

Figure 7.4.3 In web-based management dashboard of Shamelin Star E-Reporting System shows the View All Complaints page This interface shows a complete list of every complaint by residents with important details like complaint id, category, resident name, status, priority and date of submission. Complaints can also be filtered and reviewed by administrator, who can select individual complaints for more detailed information or further action such as assigning staff and updating status. Bootstrap components are used to build a well-structured and responsive layout for the page. With real-time data from Firebase Firestore, newly added complaints automatically appear in the list so that management can oversee issues and respond promptly.

7.4.4 Details Complaints Window

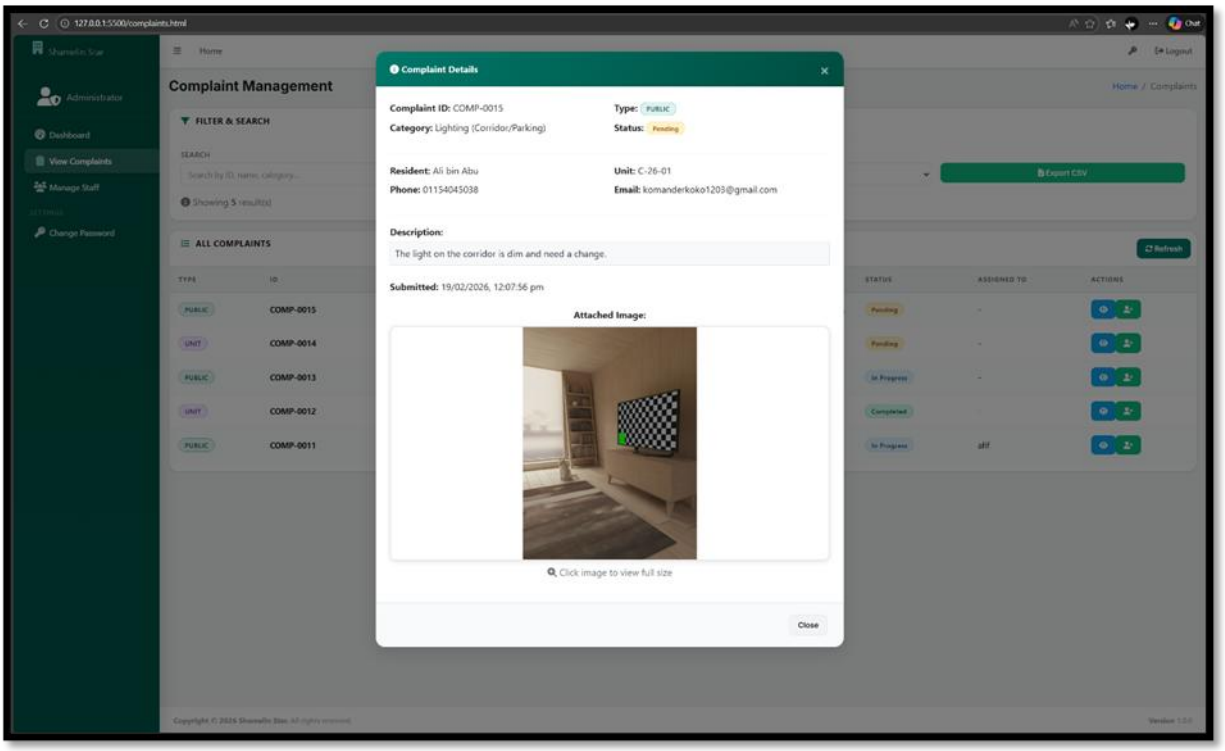


Figure 7.4.4: Details of each complaint

Figure 7.4.4 Sample window of the Complaint Details in the management dashboard of Shamelin Star E-Reporting System. This interface shows complete details about a selected complaint, including resident information and their complain category, description of the complaint, location, image evidence if attached with the post, priority level as set by user and current status. Administrators have a full view of the complaint record and can assign staff, update status, add internal notes, etc. This window pulls data directly from Firebase Firestore to guarantee real-time updates and accuracy.

7.4.5 Assign Staff Window

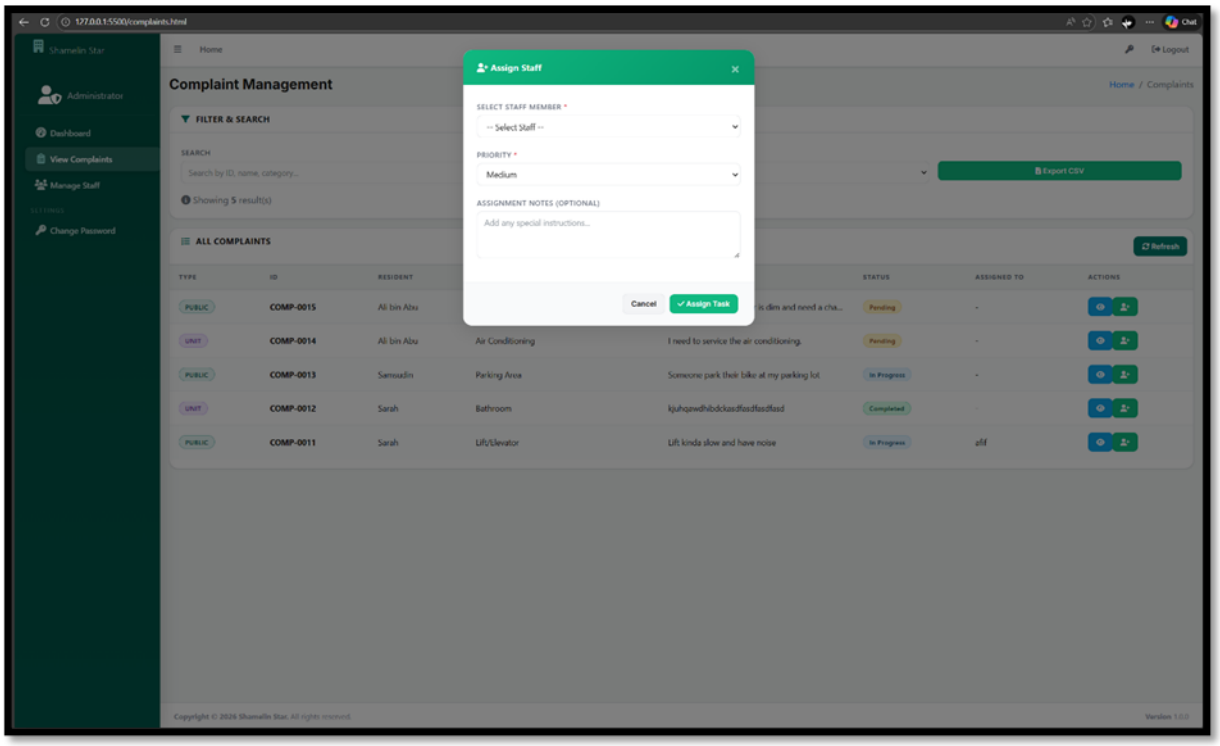


Figure 7.4.5: Assign staff to the complaints

Figure 7.4.5 Assign Mode Administrators can serve up specific articles according to department or geographic area. It also shows a ticket in the calendar so staff can tell when that particular complaint will be confirmed. This involves displaying relevant complaint information in a window that has a list of staff. From the database, it presents a list of editors who could do that item Administrators can further invoke rules to set priority levels and add assignment notes or other technology. Once confirmed, the assignment details are based on real-time database updates, examining fault tolerance over responsibility as well as efficiency of operational procedures.

7.4.6 Staff Management

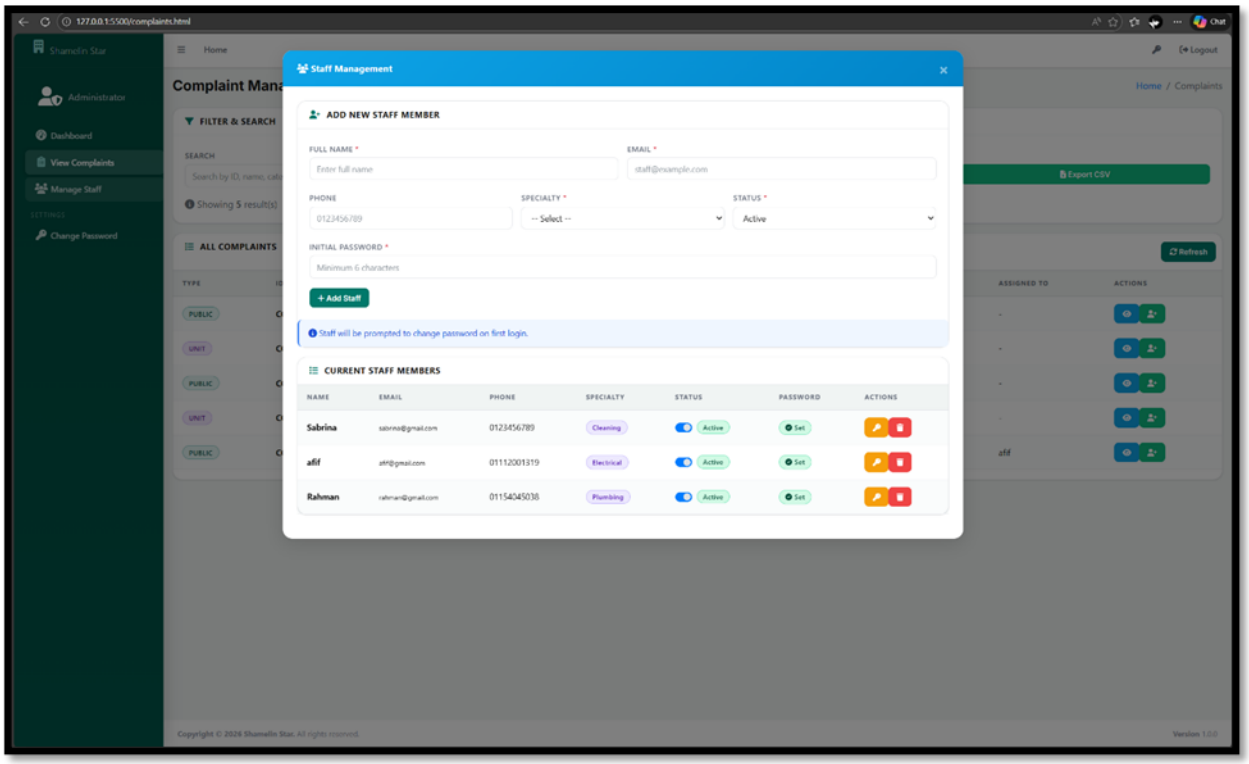


Figure 7.4.6: Staff management window

Figure 7.4.6 In Shamelin Star E-Reporting System's web-based management dashboard. The Staff Management interface. For mananging employee records the Name, Email Address, Phone numbers Specialization and Account Status a tabular way. Which makes it easy to view at a glance. We connected into Firebase Firestore the interface for staff data to be saved and retrieved. It is possible to interactively edit staff data through a third-party interface rather than from the user interface itself. This also module helps maintain a systematic approach toward workforce managment soo busy care waxes may have such information avoided even if that would eventually lead to administrative problems The appropriateness of manager is beyond question since it automatically assign each complaint to its respective party.

7.4.7 Reset Staff Password

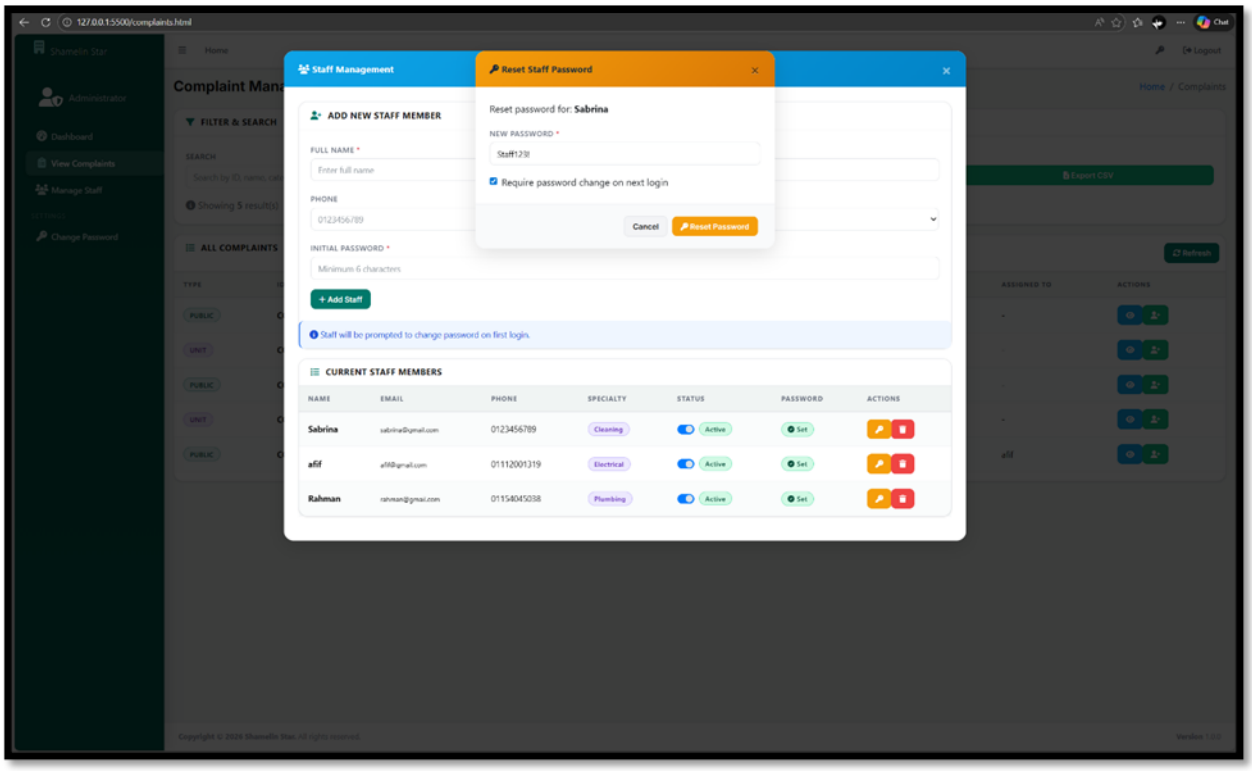


Figure 7.4.7: Reset the staff password

Figure 7.4.7 The Reset Staff Password menu as shown in the Shamelin Star E-Reporting System management dashboard. This page allows administrators to reset the login credentials of staff members who forgot their password or in case of compromised passwords. The interface restricts the action only to proper administrator level users, for system security. After confirming the reset, the new credentials are synchronized with Firebase Authentication and the staff member must change password after next login. What this does is it improves the account security, verifies that only approved access is happening to the system.

7.4.8 Delete Staff Window

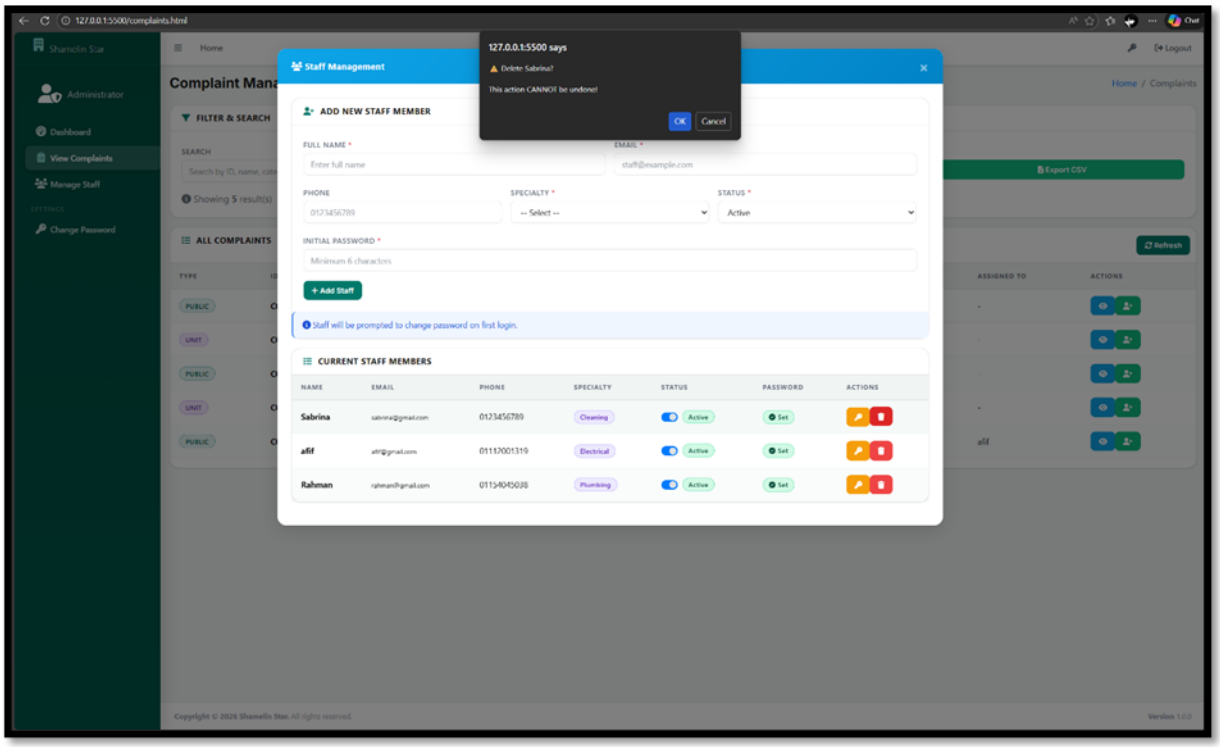


Figure 7.4.8: Delete staff

Figure 7.4.8 In the management dashboard of the Shamelin Star E-Reporting System, upon selecting the staff that you want to delete, a window for Delete Staff will appear below: This interface enables an administrator to remove the account of a staff member in case it is no longer needed because of a role update or resignation. If the staff record on Firestore is removed, then the associated authentication credentials are updated accordingly. This feature helps support proper user account management while maintaining the integrity and security of the system.

7.4.9 Staff Login Page

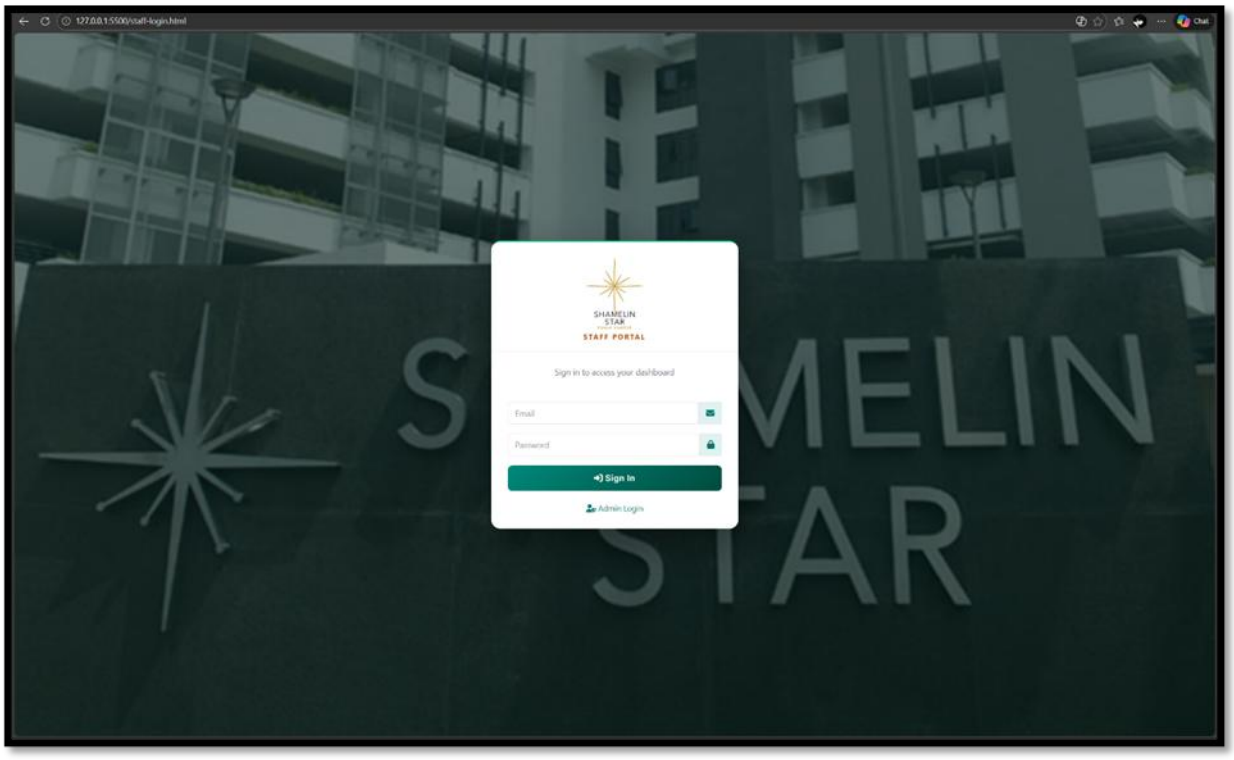


Figure 7.4.9: Staff login page

Figure 7.4.9 Staff Login Page of Shamelin Star E-Reporting System Registered staff can enter the system through this web interface using their corresponding email and password. This is made possible by implementing Firebase Authentication as part of the login process to confirm users and allow staff functionalities only. So, to stop the user from even trying: There is input validation. Once authentication is successful staff is redirected to their dashboard in which they are able to view complaints assigned to them and update the progress of complaint. This will help secure access to the operational module of the system.

7.4.10 Staff Dashboard

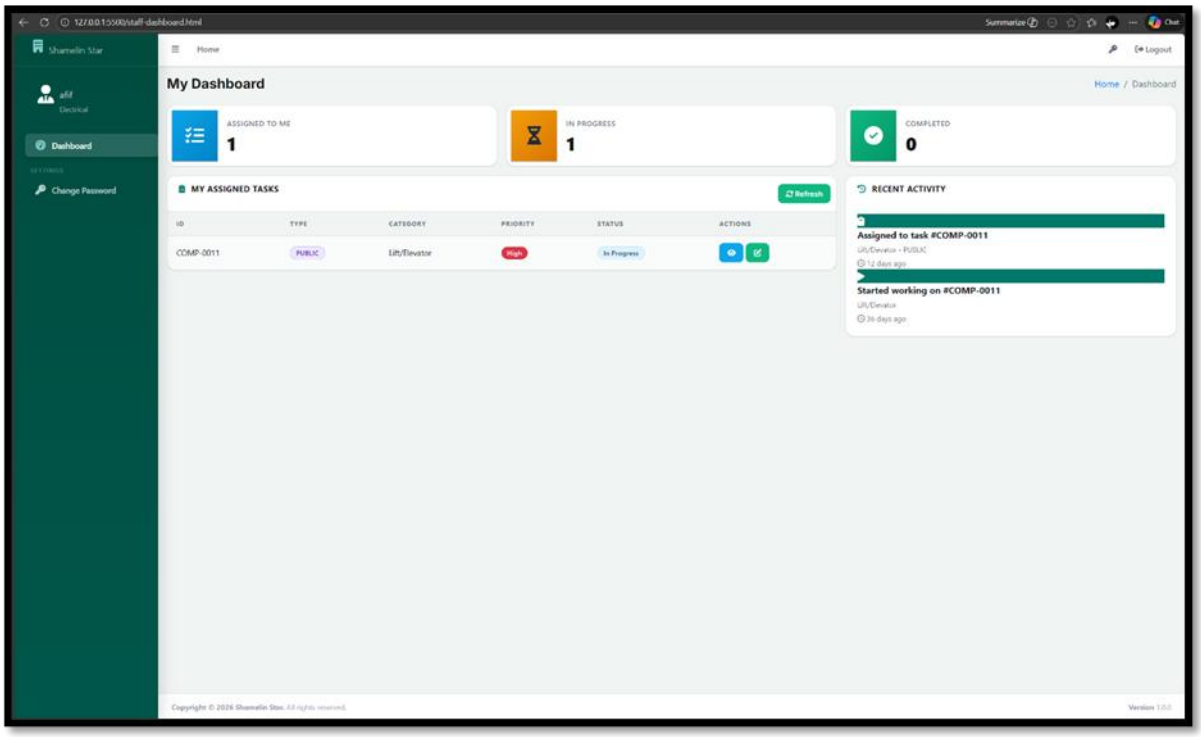


Figure 7.4.10: Staff dashboard

Figure 7.4.10 Staff Dashboard of the Shamelin Star E-Reporting System. Overview of complaints assigned to a particular staff member including complaint id, category, priority level and current status. Staff can view detailed complaint information and update progress through the dashboard. All data which is shown in this page is fetched directly from firebase firestore in real time. This will facilitate easier storage and enhance responsiveness to resident complaints with a structured interface.

7.4.11 Task Details Window

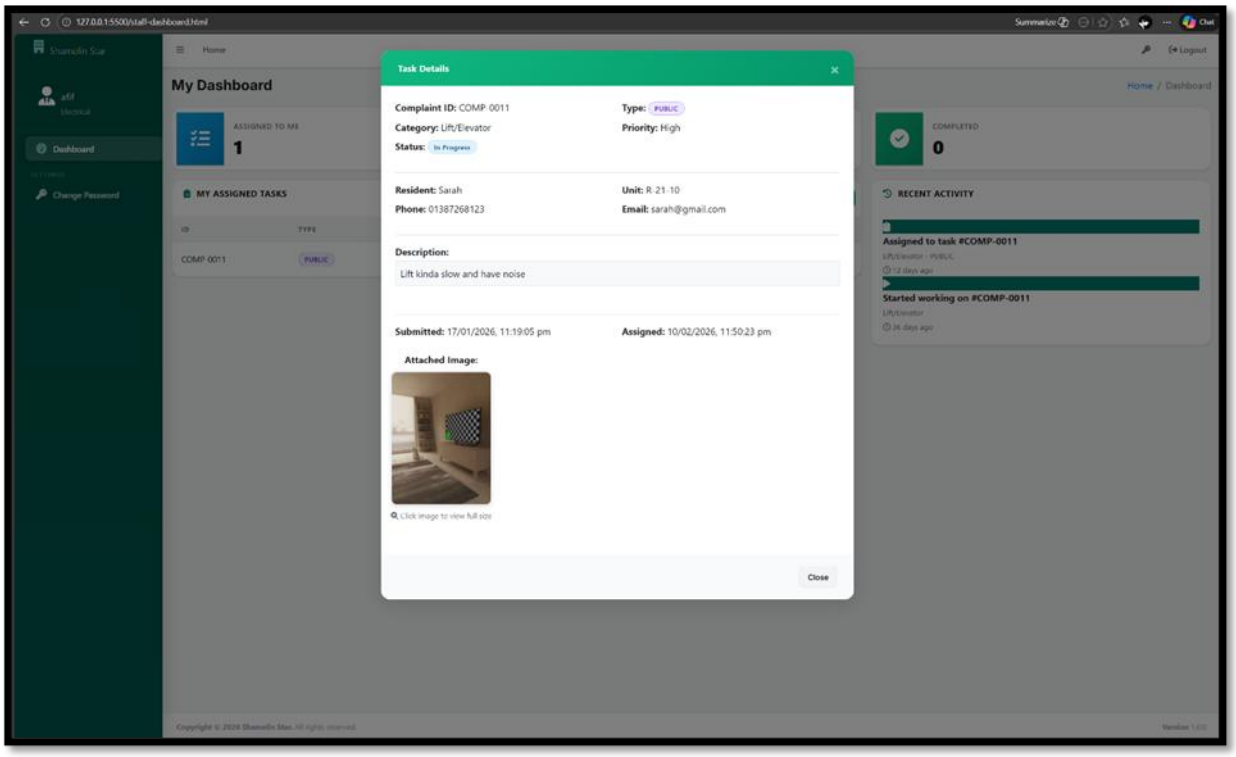


Figure 7.4.11: Task details

Figure 7.4.11 Task Details window from the Staff Dashboard of the Shamelin Star E-Reporting System. This contains detailed information about a particular assigned complaint, which consists of resident detail, complaint description, location of the complaint, and priority level as well as image evidence if attached. Staff members are able to view the complete record on a complaint and change the status of the task marking it as in progress, done. The window talks directly to Firebase Firestore reads and writes data real time. This aspect helps you track the progress of each task accurately and effectively manage their complaining resolution.

7.4.12 Update Status Window

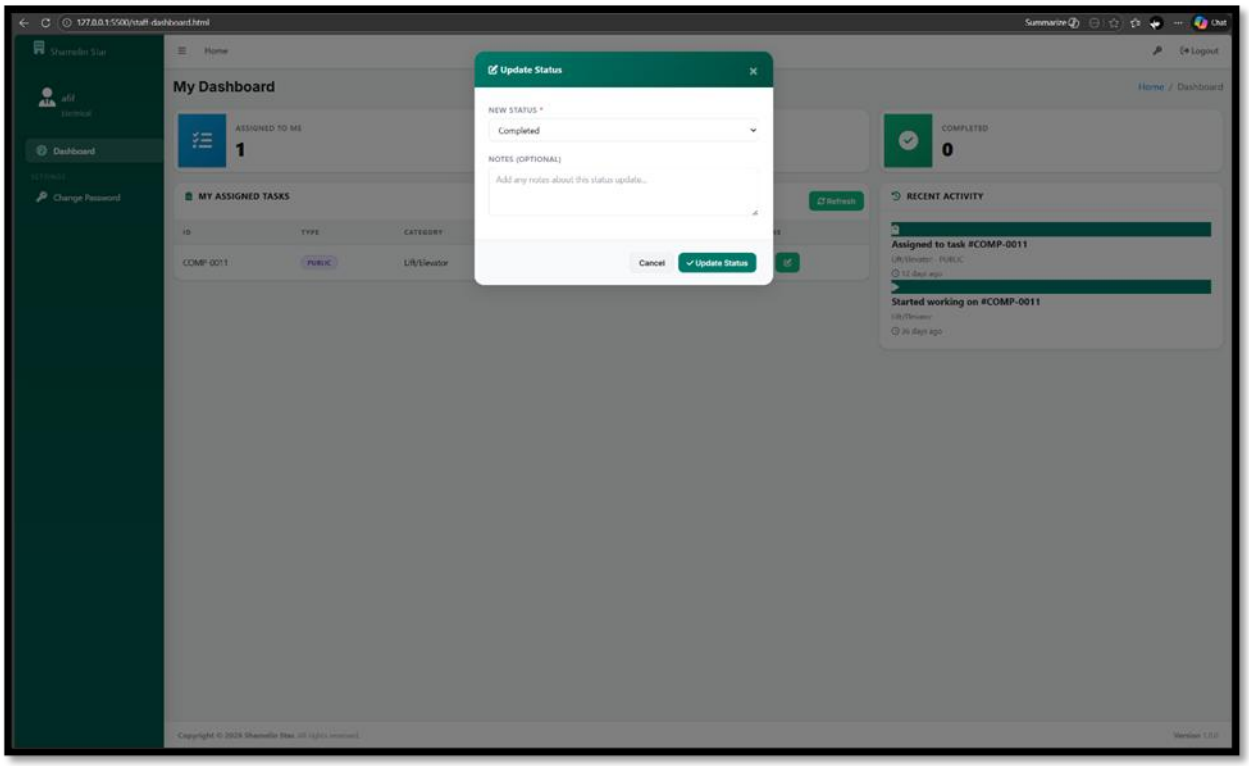


Figure 7.4.12: Update status

Figure 7.4.12 The Window Update Status in the staff module of Shamelin Star E-Reporting System is as illustrated. The Staff UI enables staff to update the state of an assigned complaint (change to in progress, complete). When updating a ticket, staff can also add additional notes to detail the resolution. To ensure real-time syncing, this was done with Firebase Firestore so that once the update is submitted, both the administrator and residents will see any changes to a complaint immediately. It improves communication, transparency in the resolution process and also keeps proper notes of complaints and progress.

7.4.13 Resident Login Page

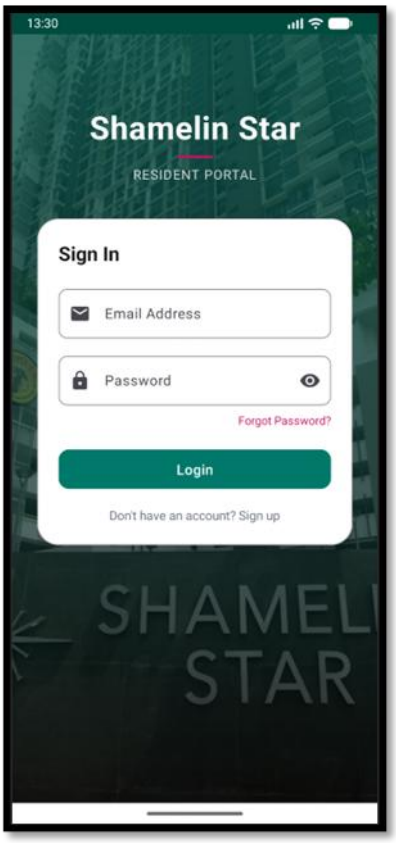


Figure 7.4.13: Resident login page

Figure 7.4.13 The Resident Login Page of the Shamelin Star E-Reporting System mobile application can be seen only authorized residents can access the application through this interface using their email and password credentials. User login is also integrated with Firebase Authentication in order to ensure user identity validation to access system features. Examples of technologies include: input validation check to make sure that the required fields are filled, incorrect logins are handled correctly. After successful login, the residents are redirected to home page that contains option to file complaint, track complaint and other system options. Provides a secure and controlled access to the resident module.

7.4.14 Resident Register Page

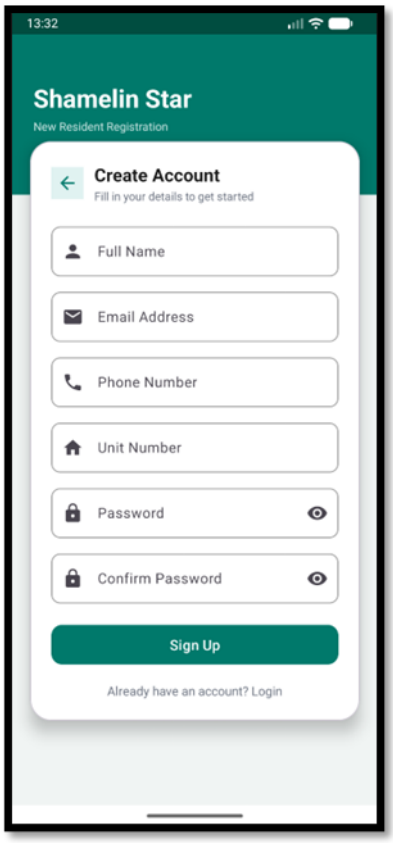


Figure 7.4.14: Resident register page

Figure 7.4.14 The Resident Register Page of the Shamelin Star E-Reporting System mobile application as shown. This interface is used by new residents to create an account with input such as full name, email address, phone number and unit number. It should validate input to make sure all required fields are completed correctly before creating an account. Once registered successfully, residents can log in to submit complaints/ track their complaints and access other application services.

7.4.15 Home Page

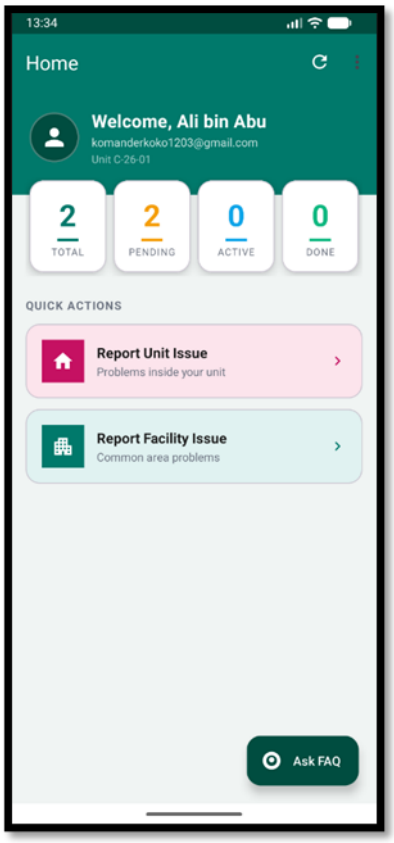


Figure 7.4.15: Home page

Figure 7.4.15 Check out Shamelin Star E-Reporting System Mobile App Home Page. This is the landing page after a resident logs in to the system. This page is the user's path into the system. Ultimately, while some of it changes for each complaint, regardless there will never be fewer than these four protocol pieces with relatively recent examples found here and here. The page uses XML design to be neat and uniform, helping users easily navigate their way through. The home page serves as master of the scenes in this mobile app system, ensuring residents can easily navigate system capabilities. At the same time it strengthens sense of order when users are looking through display screens.

7.4.16 Edit Profile Page

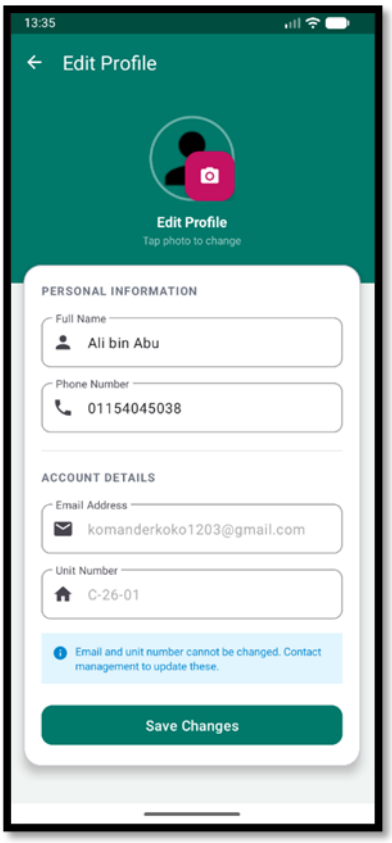


Figure 7.4.16: Edit profile page

Figure 7.4.16 Here is a screenshot of Screenshot: The Edit Profile site from Tianjin Star Electronic Report System. Workers can enter an interface to modify his full name and phone number, and upload personal photo images. Furthermore input validation checks details matched in fact are entered by the user. After the settings have been saved, changes made in information are synchronously updated with Firebase Firestore for data consistency. Using the function, the local user has full control over editing his own account details but can still be sure that accurate individual records are always retained on system.

7.4.17 Track Complaints Page

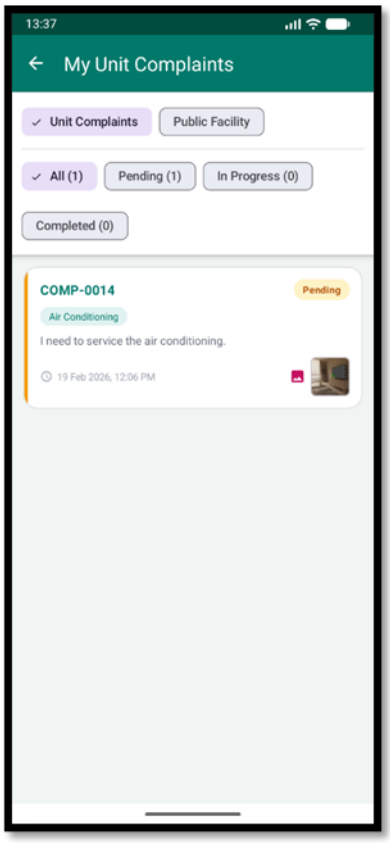


Figure 7.4.17: Track complaints page

Figure 7.4.17 The Track Complaints Page of the Shamelin Star E-Reporting System Mobile Application appears here.In the picture,Shamelin Star E-Reporting System Mobile Application's Track Complaints page is displayed. For previously filed complaints this interface will enable residents to check their status.Record includes complaint ID, category, submission date, priority level and current status.Residents can tap on an individual complaint for more detail and updates. Data is pulled from Firebase Firestore in real-time. This ensures that the upcoming information is fresh as of when you asked for it. Tapping into the platform Firebase gives real-time update system on complaint status and engages public residents around issues they flagged even if they're not privy to all technical the ins and outs.

7.4.18 Report Unit Complaints Page

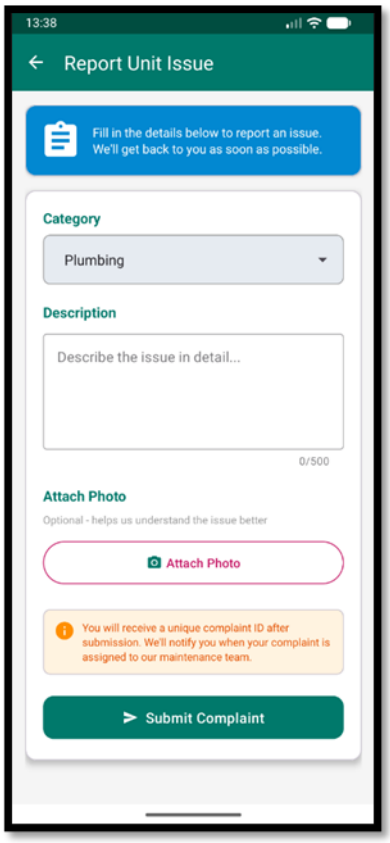


Figure 7.4.18: Report unit complaints page

Figure 7.4.18 Users can access this service on the Public Portal. Users can also report recoats from fresh units. There are entries to pick classification of complaint on the input page, input detailed description, specify place, and join any necessary image evidence. Be sure all the information required is supplied before sending in this form. At write-in submit, complaint information is saved in Firestore of Firebase, which synchronizes with the administration dashboard in real time. To sum it up, this functionality will allow the owner to report efficiently and also ensure that data collected on complaints is comprehensive and standardized.

7.4.19 Report Public Facility Page

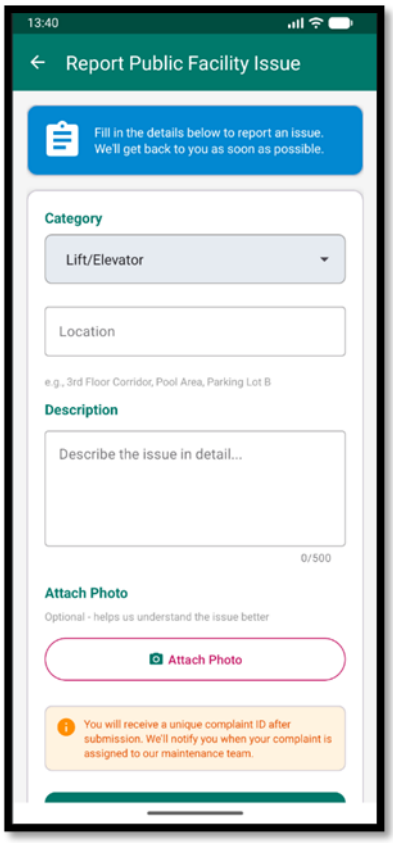


Figure 7.4.19: Report public facility page

Figure 7.4.19 The Report Public Facility Page of the Shamelin Star E-Reporting System mobile application as described. It provides residents an interface in which they can log issues with shared or common facilities where they all live for example elevators, parking areas, lighting or recreational spaces. This shows the input types used to select a category of complaint, a description for the complaint, affected locality and an image evidences for supporting. Once submitted, the complaint data is saved in Firebase Firestore and synchronized in real time with management dashboard allowing administrator to quickly review the complaint and assign a task to resolve it.

7.4.20 FAQ Chatbot Page

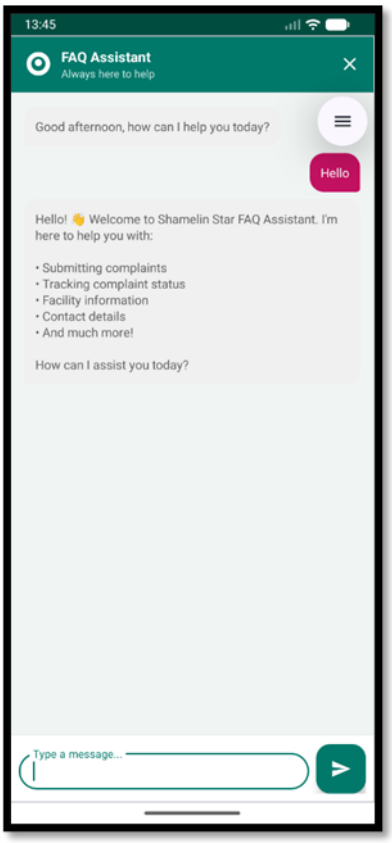


Figure 7.4.20: FAQ Chatbot page

Figure 7.4.20 shows the FAQ Chatbot Page of Shamelin Star E-Reporting System mobile application The interface provides automated assistance to its residents by answering frequently asked questions about complaint submission procedures, how to use the system, and common residential issues. The chatbot is based on an interactive messaging interface that mimics real-time dialogue which can increase user engagement. This can aid in alleviating dependency on manual administrative assistance by providing immediate answers to universal questions.

7.5 Significant Function

This portion introduces basic functions implemented of Shamelin Star E-Reporting system, which is what makes it tick. We ve added selected source code segment screenshots to show how Firebase integration and application logic was implemented in key system processes.

7.5.1 Complaint Submission Function

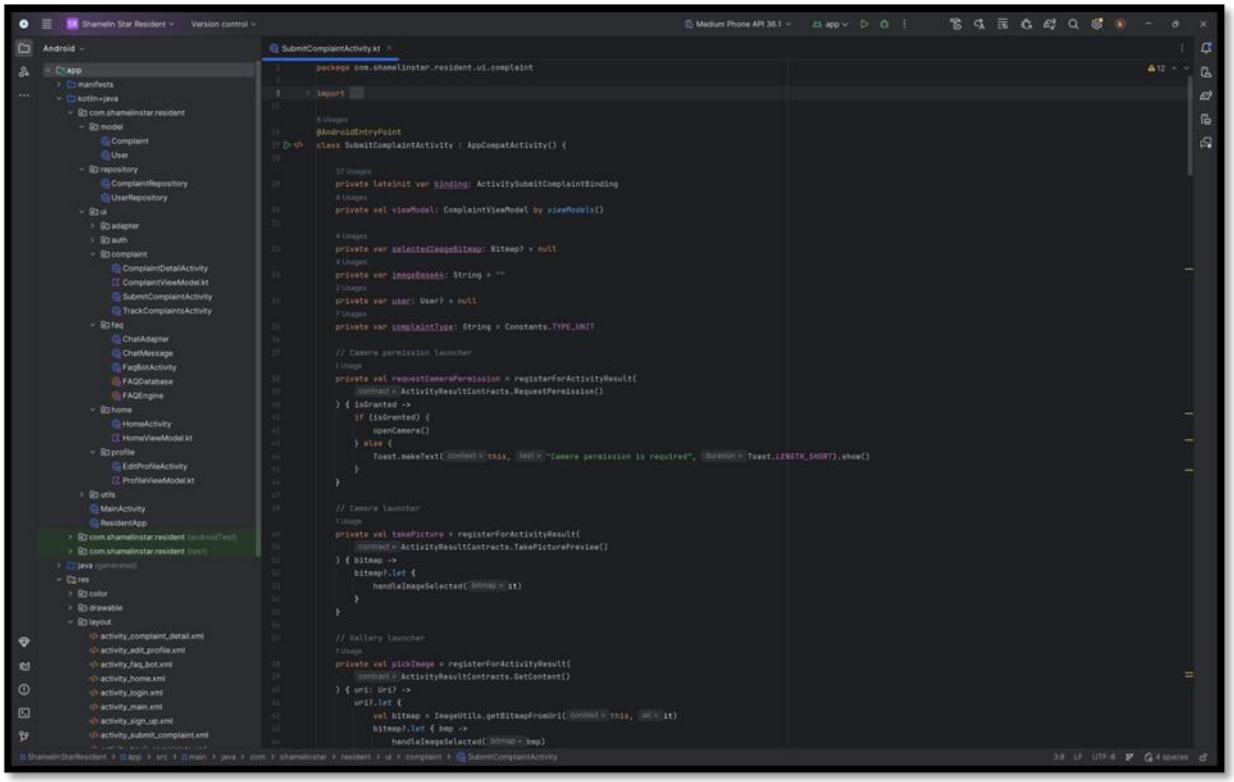


Figure 7.5.1: Complaint Submission Function

Figure 7.5.1 shows the Kotlin function used to submit a complaint from the mobile application to Firebase Firestore. First, this function retrieves the input values from the user. That includes category, description, location and priority and it downloads image files (in Base 64 format) for insertion. Before being submitted, input validation attempts to confirm that all mandatory fields are completed correctly. Once we have successfully stored the data after that firestore We also update our data in the web-based Management Dashboard in real time. This is the basic system function, enabling smooth communication between residents and management.

7.5.2 Assign Staff Function

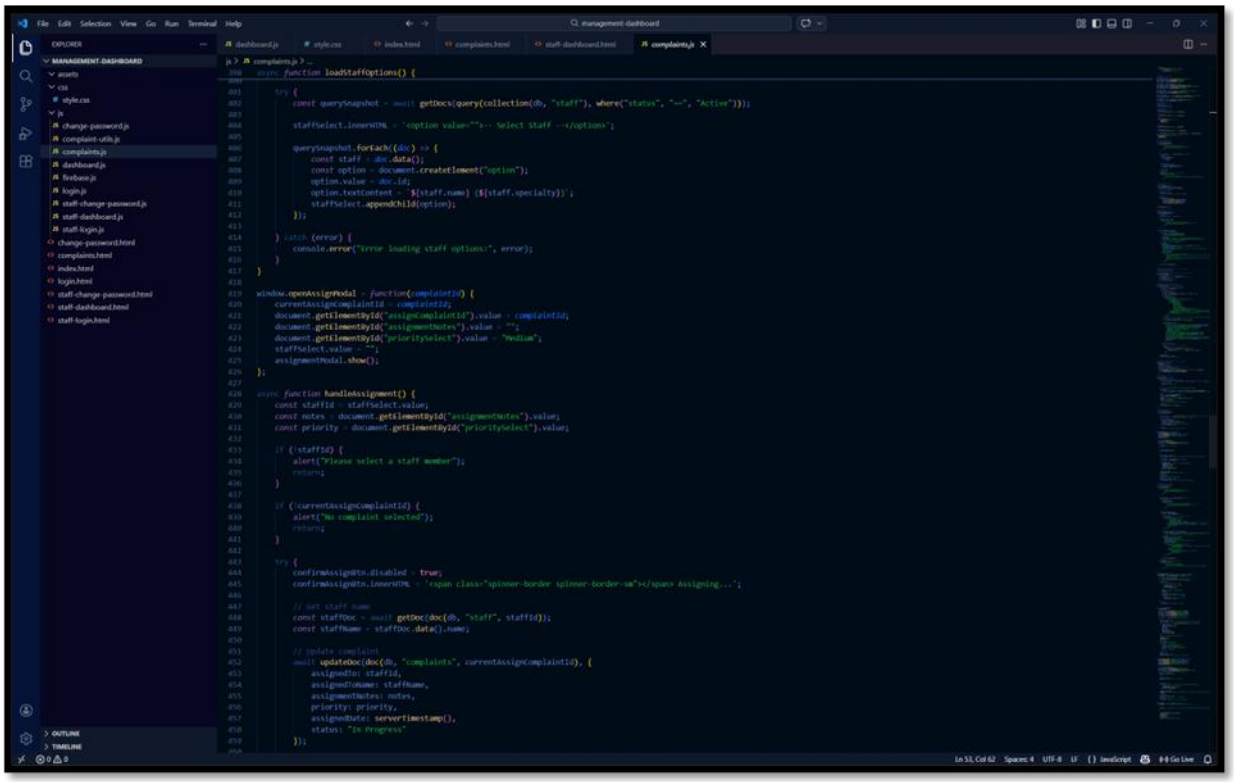


Figure 7.5.2: Assign Staff Function

Figure 7.5.2 The JavaScript code in the staff dashboard, which is used by complaint management, is shown below. It can modify the staff member assigned; but if 'doc' is object taken with Firestore, so assignedOn, status do not.The update operation appropriately updates roles in the staff dashboard and gives instant visual feedback about who is doing which tasks right now. This method demonstrates both role-based control on the system side and real-time synchronization with databases, without which structured complaint management could be sprawling and unmanageable.

7.5.3 Update Complaint Status Function (Staff Module)

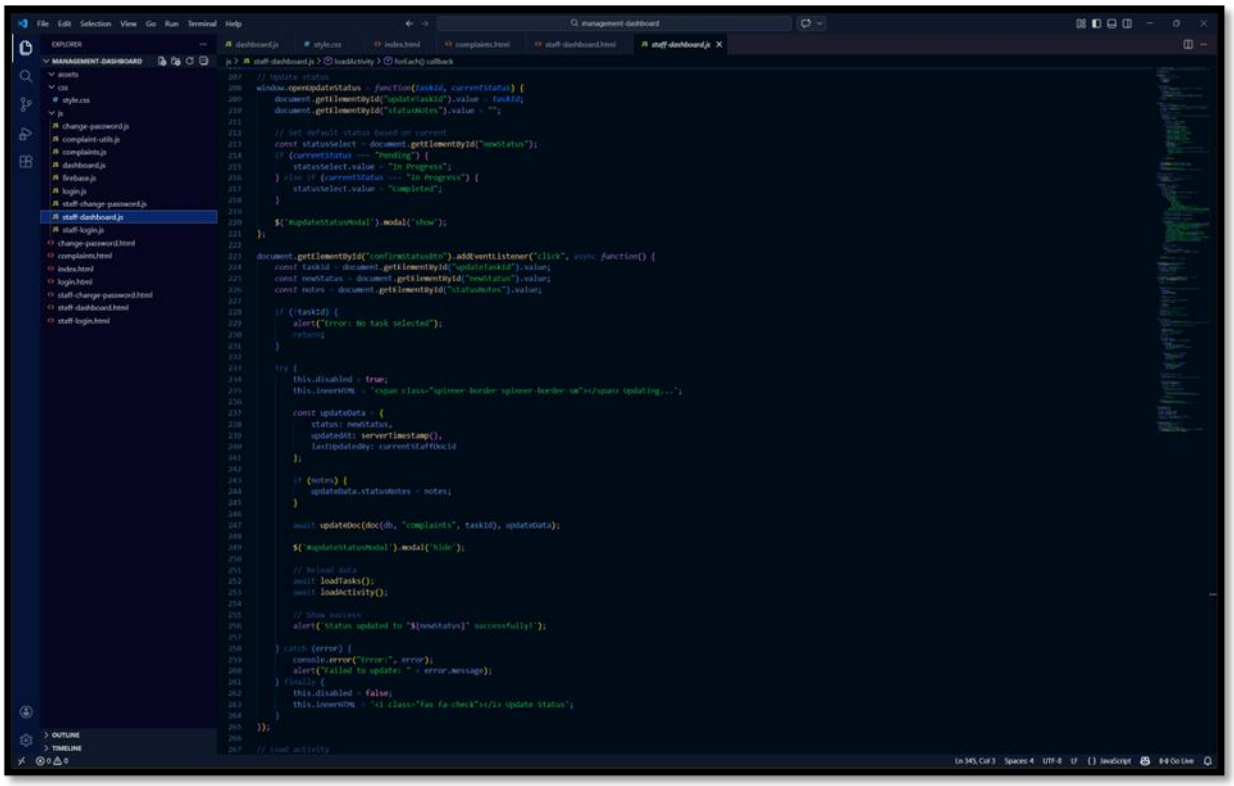


Figure 7.5.3: Update Complaint Status Function

Figure 7.5.3 which was also implemented into the staff module of Shamelin Star E-Reporting System Staff can use this function to change the current status of an assigned complaint to In Progress or Completed. Thecode utilizes Firebase Firestore's update operation to change selected fields in theselected complaint document e.g. status, updatedAt timestamp and optional staff remarks.

7.5.4 Firebase Authentication Logic

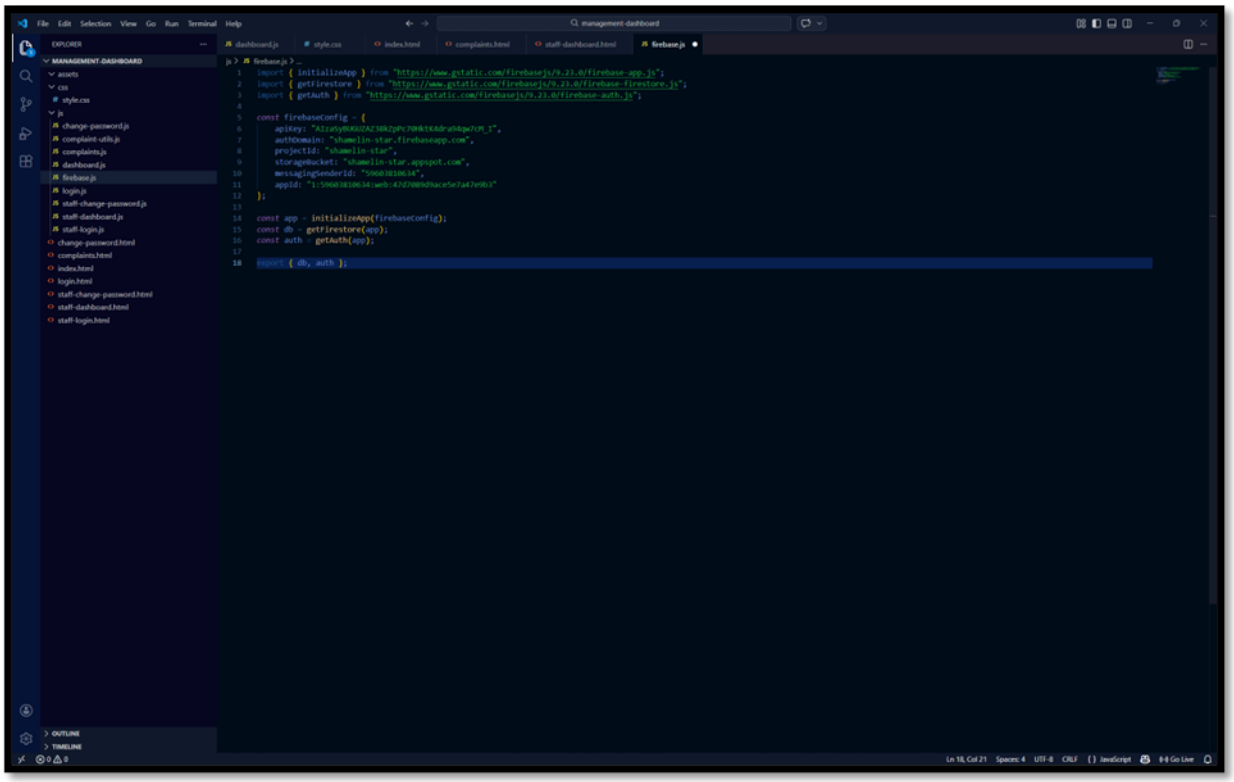


Figure 7.5.4: Firebase Authentication Logic

Figure 7.5.4 The following code demonstrates how Firebase Authentication services can be integrated with our own authentication logic. The login function verifies that the email and password are valid before allowing access to the system. As soon as staff, administrators or residents log in, Access Control allows them direct access to the appropriate dashboard according to their roles. In this way well-controlled access within a system makes it more difficult for people who do not belong to log in, adapted from word reports and year complaints data.

7.6 Conclusion

This fairly detailed chapter post we reviewed how to implement the Shamelin Star E-Reporting Method: a compilation phase following system building itself, inclusive of execution platforms, development tools, interfaces and some important functional modules. Starting from Windows 11, Visual Studio Code, Android Studio and Firebase Firestore web-based operations dashboard may interact with a resident mobile application. Be it complaint-filing, validating tickets or user status/data updates, the core functions make use of a real-time database synchronisation technology that allows each user role to speedily communicate with another all across the system. Intuitive interfaces reside in a purely instinctive fashion, the organization of task is clear, and strict access control is also observed In conclusion, the system as constructed meets the functional requirements of previous chapters and is suitable for future development.

8 TESTING

8.1 Introduction

In this chapter the testing process is achieved in order to make assured that Shamelin Star E-Reporting System operates actual according to prospection (Sommerville, 2016). Testing at the point of code, functional testing and user acceptance of the system are interested inTwo such testing methods.These processes measure performance at both the constituent system level, from end to overall ACS as well. (Pressman, 2020). We conducted tests on the web-based management dashboard and resident mobile application to ensure complaint submission flow, staff assignment, authentication and status update features are working as designed (Myers, Sandler & Badgett, 2011) In addition to regular backend integration testing, the real-time syncing of any changes made by one user is observed at every level through Firebase Firestore with varying roles (Moroney, 2020) The systematic testing of the software identified and corrected potential errors which, therefore led to greater reliability and performance of the system (Pressman & Maxim, 2020).

8.2 Unit Testing

Rooted in broader software development principles, unit testing was done to insure that each respective function worked as intended and independently of one another (Pressman & Maxim, 2020) Testing was done on the complaint submission logic to ensure that user inputs, such as category, description, location and image attachment, were being validated correctly and then stored in firebase firestore (Moroney, 2020). As the authentication processes were tested to establish that login and registration functions appropriately verify user credentials and prevent unauthorized access (Sommerville, 2016). To accomplish this, the complaint staff assignment feature was tested specifically to confirm that the complaint record updated correctly with accurate assigned staff information and changes in status (Myers, Sandler & Badgett, 2011). The update manager status feature was then tested to ensure that changes made by staff were reflected in real time on both the management dashboard and resident tracking page (Moroney, 2020). Unit testing is a basic process in programming development that aids in the detection of defects from an early stage, improving code quality, and increasing overall system stability (Pressman & Maxim, 2020).

Table 8.2.1: Unit Testing

Test ID	Component	Input	Expected Process	Expected Output	Status
UT1	Complaint Submission	Enter valid category, description, location and press submit	Validate inputs and store complaint in Firestore	Complaint successfully stored	PASS

				and appears in dashboard	
UT2	Resident Login	Enter valid email and password	Authenticate using Firebase Authentication	Redirect to Resident Home Page	PASS
UT3	Staff Assignment	Admin selects complaint and assigns staff	Update complaint document with assignedTo and status	Staff name displayed under assigned complaint	PASS
UT4	Update Complaint Status	Staff selects “Completed” and clicks update	Update status field in Firestore	Status updated and visible in resident tracking page	PASS
UT5	Registration Validation	Leave required field empty	Trigger input validation	Error message displayed and registration blocked	PASS
UT6	Complaint Tracking	Resident opens track complaints page	Retrieve complaint data from Firestore	Complaint list displayed with correct status	PASS

8.3 Integration Testing

After completing the unit testing, integration tests were done to ensure that individual modules of the system works fine when integrated and linked together (Pressman & Maxim, 2020). Where unit testing focuses on the components, integration testing verifies that modules talk to each other correctly and data flows through the system (Sommerville, 2016). Integration testing for the Shamelin Star E-Reporting System was done by validating that the mobile application, web dashboard and backend database (Myers, Sandler & Badgett, 2011) worked together. Integration-testing was performed over some key modules complaint submission, complaint tracking, and finally complaint management of administrators (Pressman & Maxim, 2020).

Table 8.3.1: Integration Testing

Test ID	Component	Input	Expected Process	Expected Output	Status
IT1	Complaint Submission to Dashboard	Resident submits complaint via mobile app	System stores complaint in Firestore and syncs with web dashboard	Complaint appears in admin dashboard	PASS

IT2	Complaint Assignment	Admin assigns complaint to staff	Update assignedTo field in Firestore and notify staff module	Staff sees assigned complaint in dashboard	PASS
IT3	Complaint Status Update	Staff updates complaint status to "In Progress"	System updates status field in database	Updated status displayed to resident tracking page	PASS
IT4	Resident Tracking Synchronization	Resident opens tracking page	Retrieve updated complaint status from Firestore	Latest complaint status displayed	PASS
IT5	Complaint Report Generation	Admin selects report filter and generates report	System retrieves filtered complaint data from database	Structured report displayed on dashboard	PASS

8.4 System Testing

System testing was performed to assess the overall performance of the Shamelin Star E-Reporting System as a fully-integrated system (Sommerville, 2016). This testing is designed to confirm that all elements of the system work in a coordinated fashion and deliver on the functional requirements set in previous chapters (Pressman & Maxim, 2020). System testing differs from unit testing and integration testing in that it considers the entire system through the lens of an end-user (Myers et al., 2011). Tests were carried out to confirm all the main features of the system such as user login, submitting complaints, tracking complaints and generating reports (Pressman & Maxim, 2020).

Table 8.4.1: System Testing

Test ID	Component	Input	Expected Process	Expected Output	Status
ST1	User Registration	Resident enters valid registration details	System validates input and stores user data in Firebase Authentication	Account created successfully	PASS
ST2	User Login	Resident enters correct email and password	Authenticate user via Firebase Authentication	User redirected to Resident Home Page	PASS
ST3	Submit Complaint	Resident fills complaint form and submits	Validate data and store complaint in Firestore	Complaint recorded and	PASS

				confirmation displayed	
ST4	Track Complaint	Resident views complaint tracking page	Retrieve complaint data from Firestore	Complaint list with correct status displayed	PASS
ST5	Admin Manage Complaint	Admin updates complaint status	Update complaint document in database	Updated status reflected in resident tracking page	PASS
ST6	Report Generation	Admin generates complaint report	Retrieve complaint records and compile report	Structured report displayed	PASS

8.5 User Acceptance Testing

User Acceptance Testing (UAT) is the last stage of software testing where actual users or system stakeholders confirm that the developed system meets their requirements and works well in a real-world environment. While UAT distinguishes the test cycle from other development tests as it is not trying to find code errors but rather it ensures that system functions properly with respect to users need and expectations.

- Showing the end-to-end journey of complaint submission for the resident, right from adding details of his/her complaints to choosing a category with images.
- Enable residents to log sample complaints using the mobile app and monitor complaint status updates in real time
- Granting access to management personnel via the web-based dashboard for monitoring complaints, assigning staff, updating complaint status, and tracking overall progress on complaints.
- Introduction to stakeholders on various system features (authentication, complaint filtering or dashboard reporting functions.
- Gather feedback regarding use of system, information comprehensibility, responsiveness and whether the complaints houses would be handled effectively.

8.5.1 Client Testing and Result

Interview Details

Name: Nurin Nabilah Binti Shamsuri

Location: via Google Meet

Date: 10 March 2026

8.5.2 Client Feedback

Table 8.5.1: Feedback Analysis of Question 1

QUESTION 1	Does the mobile app allow residents to submit complaints easily?
CLIENT FEEDBACK	The complaint submission process is straightforward. I was able to enter details and submit without confusion.
ANALYSIS	A simple submission process improves usability and encourages residents to report issues promptly.

Table 8.5.2: Feedback Analysis of Question 2

QUESTION 2	Is the complaint category selection clear and relevant?
CLIENT FEEDBACK	The categories are easy to understand and match common residential issues.
ANALYSIS	Clear categorization helps organize complaints and improves management efficiency.

Table 8.5.3: Feedback Analysis of Question 3

QUESTION 3	Does the system allow uploading images successfully?

CLIENT FEEDBACK	Image upload works well and helps explain the issue better.
ANALYSIS	Supporting images enhance communication and improve problem identification.

Table 8.5.4: Feedback Analysis of Question 4

QUESTION 4	Is the complaint status tracking feature effective?
CLIENT FEEDBACK	I can easily track the status of my complaint in real time.
ANALYSIS	Real-time tracking increases transparency and user satisfaction.

Table 8.5.5: Feedback Analysis of Question 5

QUESTION 5	Is the login and registration process secure and user-friendly?
CLIENT FEEDBACK	Registration and login are simple and work without errors.
ANALYSIS	A smooth authentication process ensures security while maintaining usability.

Table 8.5.6: Feedback Analysis of Question 6

QUESTION 6	Does the web dashboard display complaints clearly?
CLIENT FEEDBACK	The dashboard shows all complaints in an organized way.
ANALYSIS	Clear data visualization helps administrators manage tasks efficiently.

Table 8.5.7: Feedback Analysis of Question 7

QUESTION 7	Can administrators update complaint status easily?
CLIENT FEEDBACK	Updating complaint status is quick and reflects immediately.
ANALYSIS	Efficient status updates improve workflow and response time.

Table 8.5.8: Feedback Analysis of Question 8

QUESTION 8	Does the system synchronize data between mobile and web in real time?
CLIENT FEEDBACK	Changes made on the dashboard appear instantly on the mobile app.
ANALYSIS	Real-time synchronization ensures consistency and reliability of data.

Table 8.5.9: Feedback Analysis of Question 9

QUESTION 9	Is the system interface user-friendly for both residents and staff?
CLIENT FEEDBACK	Both interfaces are easy to use and understand.
ANALYSIS	User-friendly design reduces training needs and improves adoption.

Table 8.5.10: Feedback Analysis of Question 10

QUESTION 10	Does the system handle multiple complaints efficiently?
CLIENT FEEDBACK	The system performs well even with multiple complaints.

ANALYSIS	Good performance ensures scalability and smooth operation.

Table 8.5.11: Feedback Analysis of Question 11

QUESTION 11	Is the complaint filtering feature useful for administrators?
CLIENT FEEDBACK	Filtering helps me find specific complaints quickly.
ANALYSIS	Filtering improves efficiency in managing large datasets.

Table 8.5.12: Feedback Analysis of Question 12

QUESTION 12	Does the system provide clear complaint details?
CLIENT FEEDBACK	All complaint details are clearly displayed.
ANALYSIS	Detailed information supports better decision-making.

Table 8.5.13: Feedback Analysis of Question 13

QUESTION 13	Is the system reliable during usage?
CLIENT FEEDBACK	The system works consistently without crashes.
ANALYSIS	System reliability is critical for user trust and continuous usage.

Table 8.5.14: Feedback Analysis of Question 14

QUESTION 14	Does the system improve communication between residents and management?
CLIENT FEEDBACK	It makes communication faster and more organized.
ANALYSIS	Improved communication leads to better service quality.

Table 8.5.15: Feedback Analysis of Question 15

QUESTION 15	Is the system suitable for real-world implementation?
CLIENT FEEDBACK	The system is practical and suitable for residential management use.
ANALYSIS	Positive acceptance indicates readiness for deployment.

8.5.3 Resident User Survey

In this section, we're going to look at the results of end user surveys conducted in order to ascertain how user-friendly and accessible Shamelin Star E-Reporting System is.That aimed to ensurer last week the survey question was given to some object people, who are major users for software mobile applications. On the conduct surveys, it is considered that the person who does qa work must be in conformity with his or her guidelines.Feedback from the survey was directed at the experience of use, information clarity,

menus and design performance of navigation user interfaces. Overall satisfaction was scored on the basis in which a system could be used to file, send responses updates and trace complaints.

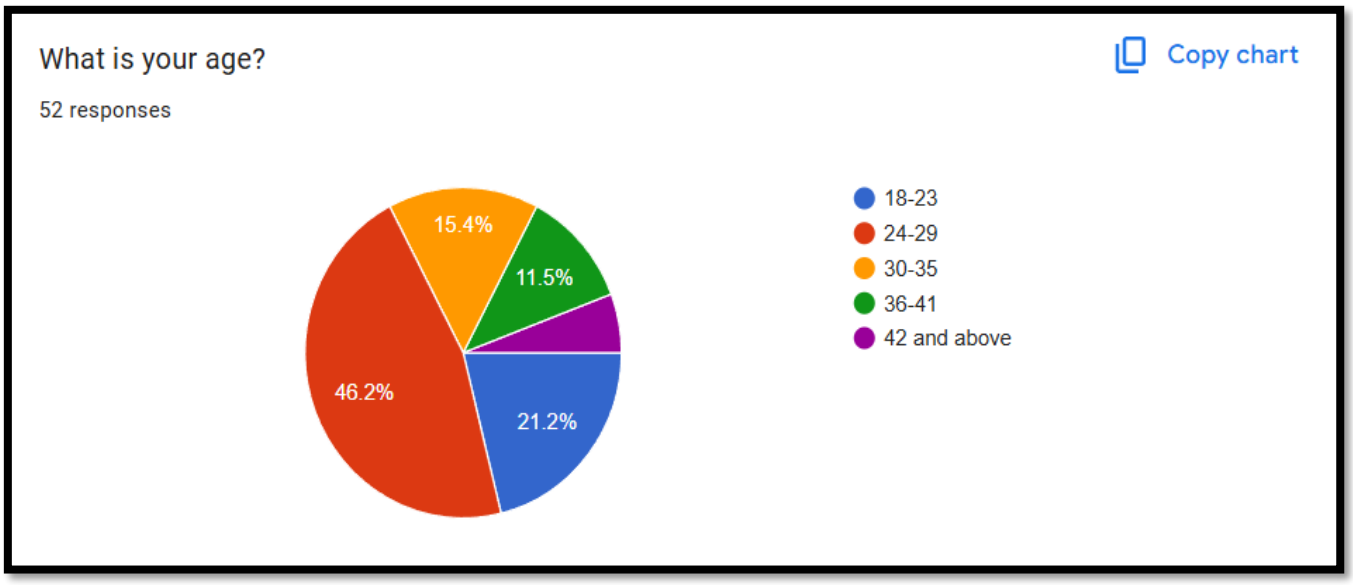


Figure 8.5.1: Question 1

Figure 8.5.1 Here the age distribution for our sample is viewed. The majority answered the age group 24-29 (46.2%), followed by 18-23. For respondents aged 30-35 some 15.4% this group was the least numerous as were its cousins aged 36-41 at just under a quarter with only 11-19% and going under messages for (42+).

The concentration of participants in the age group under 35 suggests that the survey is primarily filled out by young adults, who tend to be more adept at using mobile applications and digital systems. This is also beneficial because the Shamelin Star E-Reporting System is evaluated within a mobile application that be accessed via an online smartphone to submit and track complaints. Younger users are well distributed, which indicates that the current sample feedback is from themost relevant user group and those who are most likely to adopt and use the system.

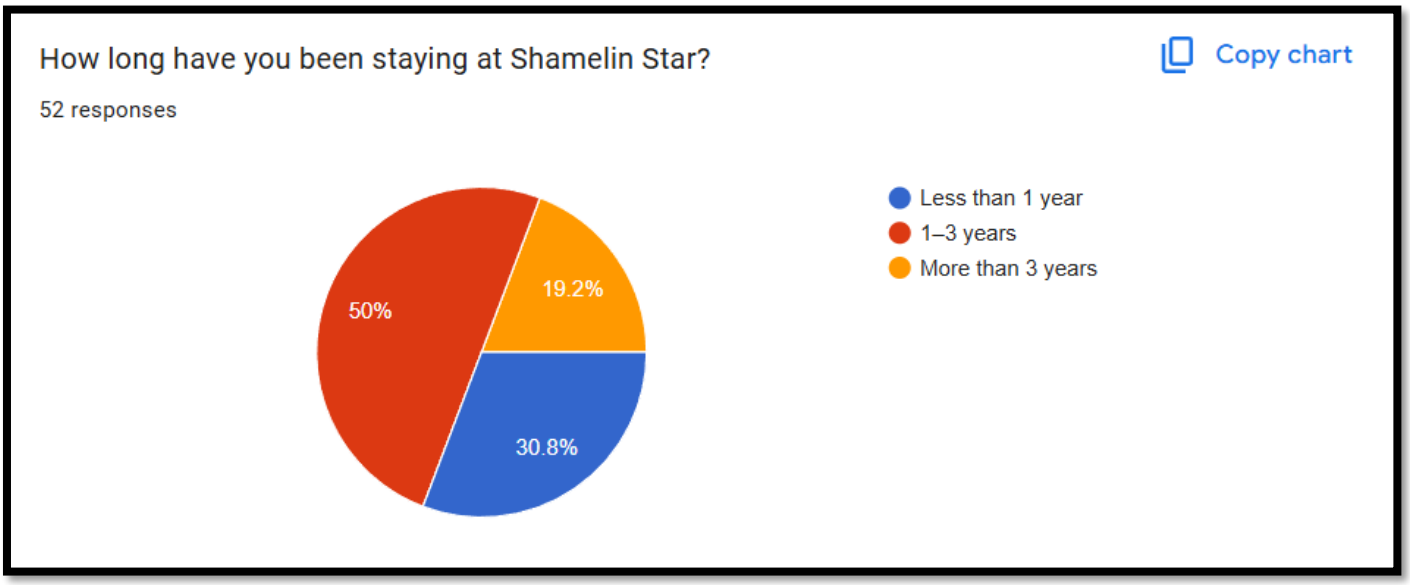


Figure 8.5.2: Question 2

Figure 8.5.2 As half of the respondents (50%) have been staying at Shamelin Star for 1–3 years as mentioned, while those who have staying less than 1 year are (30.8%), and lastly residents with more than three-year staying who is the smallest group (19.2%). Because the vast majority of unit residents are well seasoned with this courtyard's 'normal' maintenance requirements and management methods (copyright above), I trust that our new tenants will be eager and willing to add their input throughout the moving in process. In general, the chart shows that from many, many different tenures of living in an apartment house form here data was responsibly received. This e-reporting system is being graded both by people who will depend on established processes moving forward and also those that follow them dependent on you.

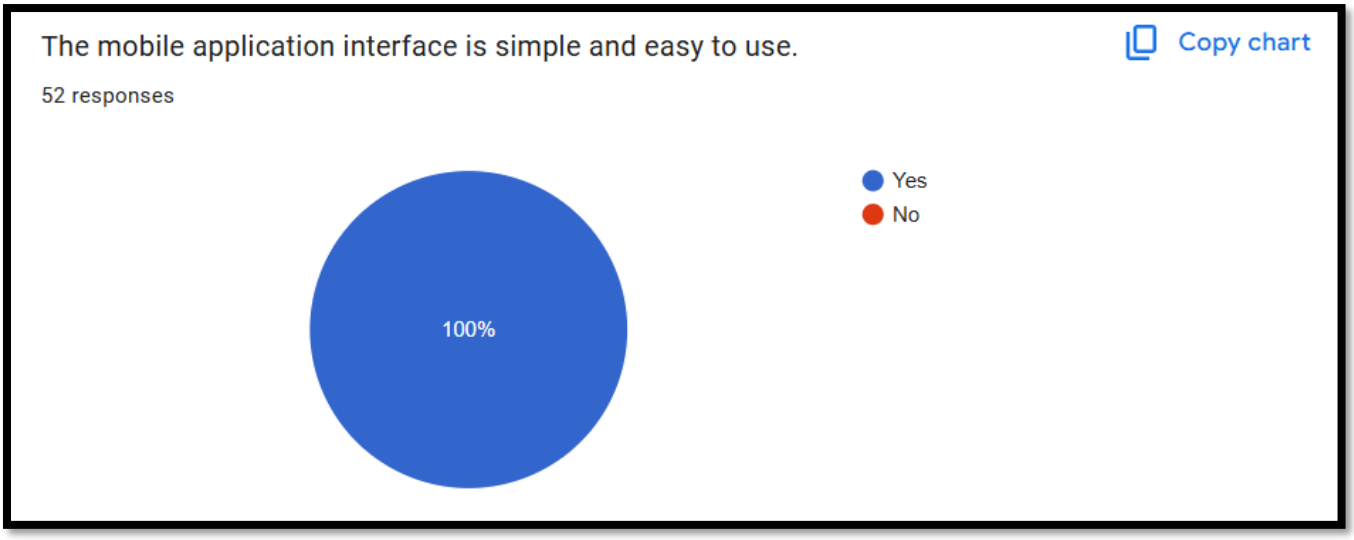


Figure 8.5.3: Question 3

As shown in Figure 8.5.3, a complete (100%) of the respondents agreed with the statement that the mobile application interface is simple and easy to use. Its evaluation is a very positive finding, because usability is key in a mobile e reporting system designed for a diverse resident demographic. An intuitive app allows

residents to engage with the tool to submit and track complaints without having to receive a lengthy tutorial or have building management standing by at their call. In summary, the chart confirms that the app accomplished what was designed to do: create a solution that is easy-to-use and within reach of everyone.

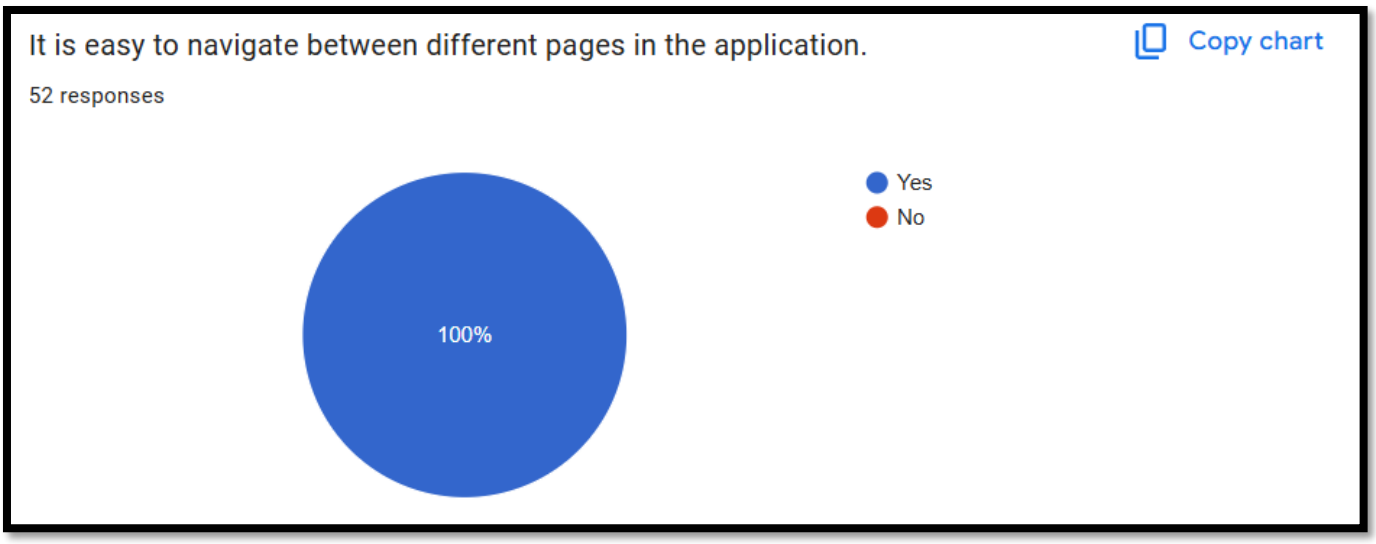


Figure 8.5.4: Question 4

As seen in Figure 8.5.4, a total of 100% of the participants responded that it is easy for the user to navigate between different pages on our application. Good navigation reduces friction and stops the user from dropping off in the reporting process. In conclusion, the chart shows that the most effective aspect of your application is its layout and user flow, since they directly respond to what your users expected from both a simple and intuitive navigation.

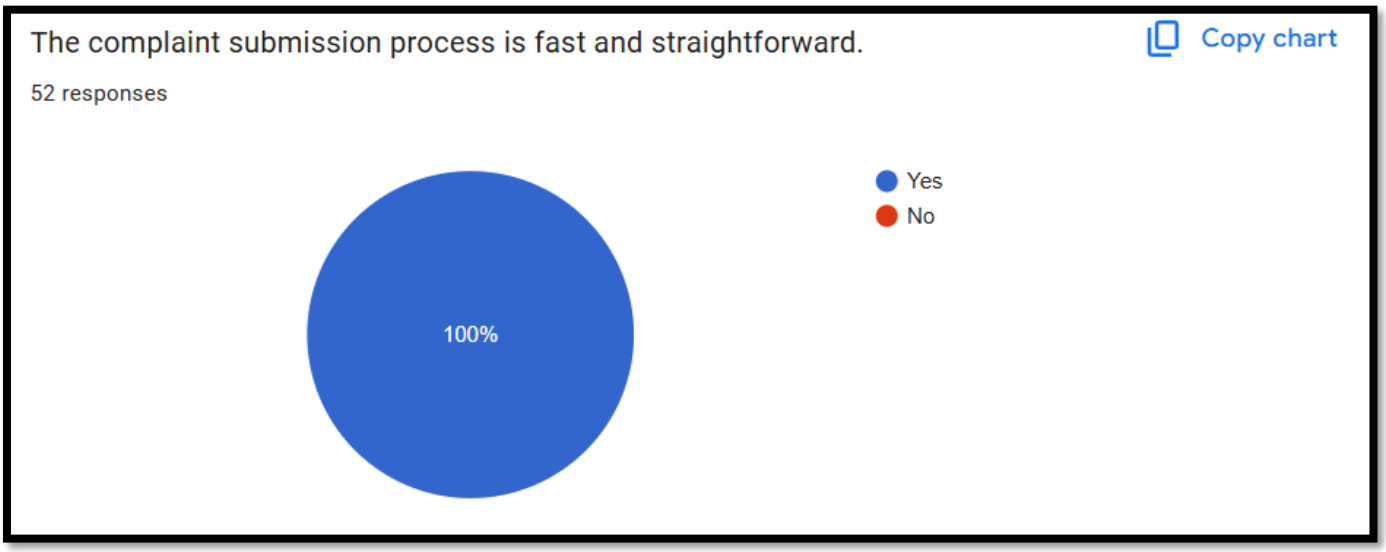


Figure 8.5.5: Question 5

As reflected in figure 8.5.5, majority agreed that submitting a complaint is fast and simple no less than 100% of the respondents 52 people. This is an important finding for the evaluation since improved submission processes are one of the major functions of an e-reporting application. Fixing maintenance problems is

made a whole lot easier if users find the system fast and without barriers they are far more likely to report problems quickly, which helps management keep the property running smoothly and efficiently. So, the graph shows that making a complaint which is the crux of our app has been optimised to make it easier for the user.

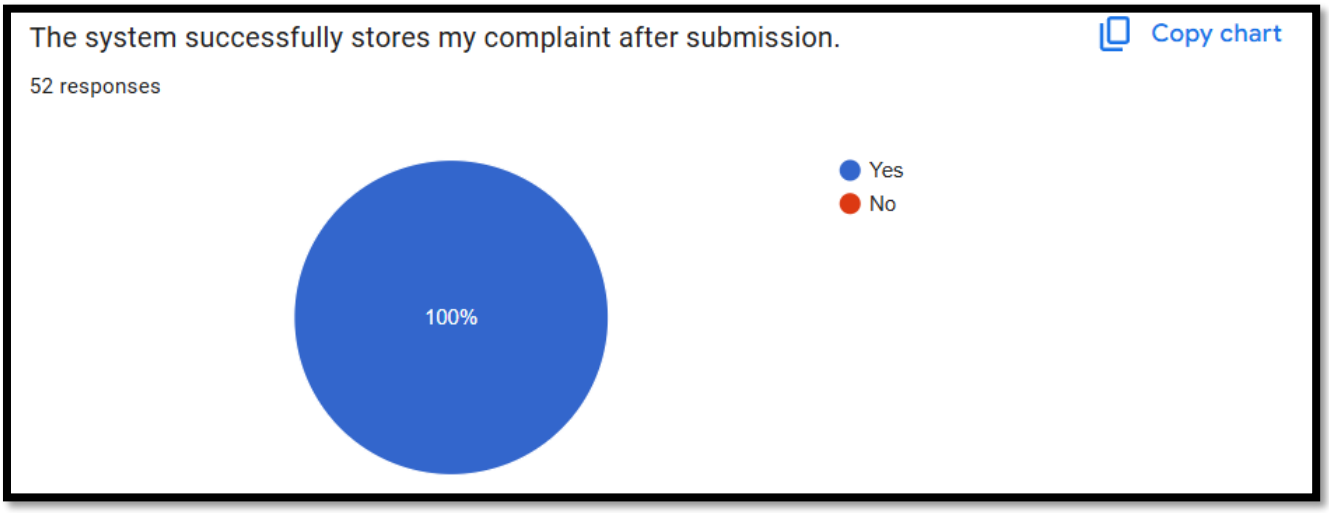


Figure 8.5.6: Question 6

As illustrated in Figure 8.5.6 all respondents (100%) indicate that the system successfully saves their complaint upon submission of it. This index is vital for evaluating system reliability and establishing user trust. Consequently, once a resident feels confident that he can submit his data securely and without technical failures (or losing any of it) so that the application itself is safe to use he will certainly prefer using this new style of problem report instead of resorting back again to traditional methods which themselves might not be free from danger.. The chart shows the backend data handling of the system flawless from the perspective of user level.

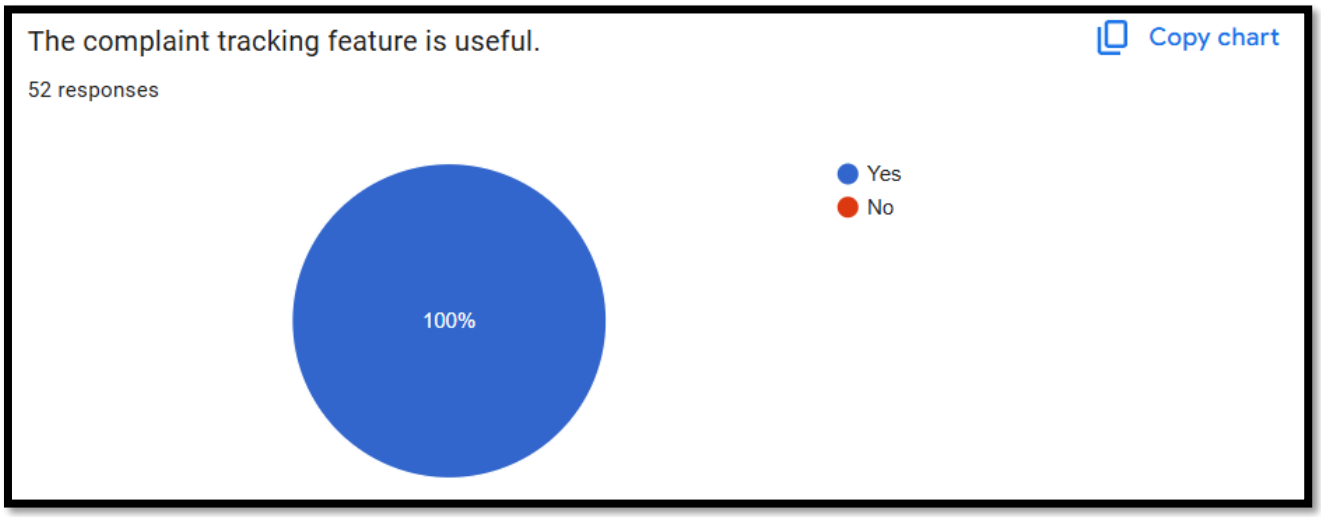


Figure 8.5.7: Question 7

In Figure 8.5.7, we see that all (100%) of the users that were surveyed also found the ability to track their complaints useful only 0% of users disagree. Giving visibility into the status of a reported issue solves a very ubiquitous pain point in property management: lack of communication. It includes a tracking feature that empowers residents by updating them on the progression of their requests, which significantly lessens the need for tenants to chase down management office. In summary, as the entire agreement shows, tracking functionality features are very useful and improve user experience.

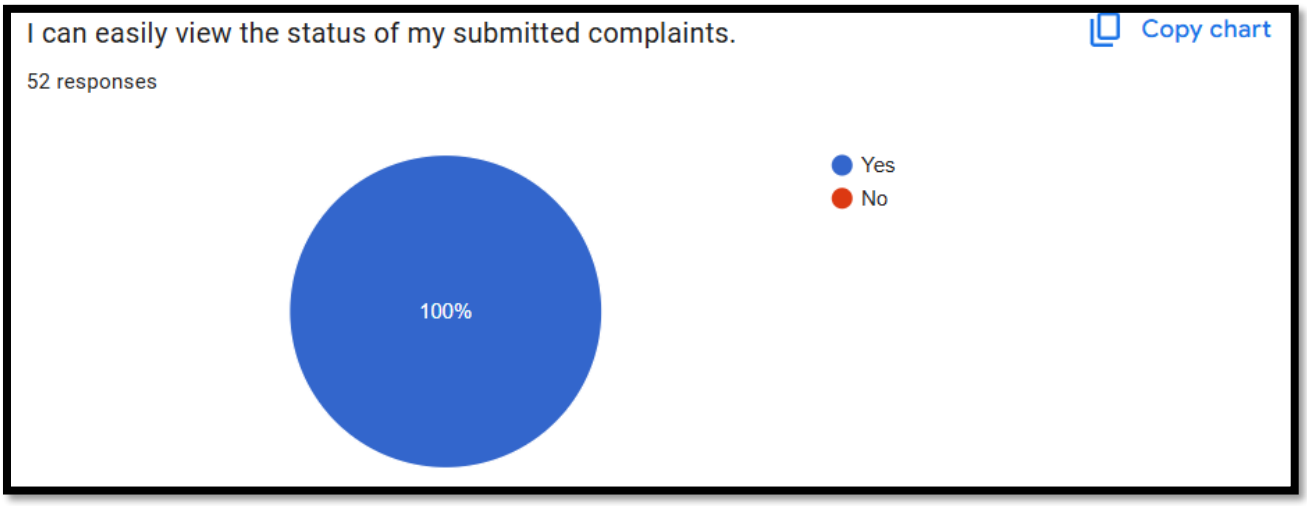


Figure 8.5.8: Question 8

As shown in Figure 8.5.8, 100% of the (52) respondents agreed they could easily view the status of their complaint That is a very good result for evaluation. The easier such information is to acquire, better it can satisfy these 6 human needs: transparency. By removing a sense of anxiety over their report's status and performing it automatically, residents are freed from the burden of frequently contacting management office to obtain up-to-date activity updates. They have a constant ability nowadays (ereporting system made by Silence) to check on and follow their current announcements at any time via desk plotter-based hierarchal form. From the chart in general, we see that this user interface design is effective for giving current complaints visibility.

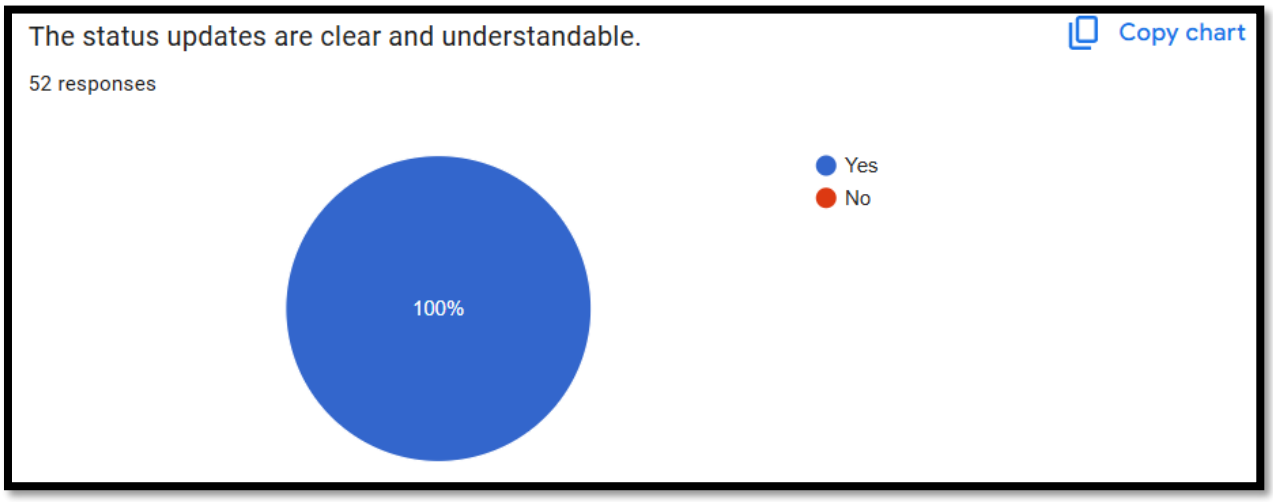


Figure 8.5.9: Question 9

Figure 8.5.9 All (100%) of those questioned answered that the status updates were clear and understandable. Although this may not come directly into play for evaluation, it brings some insight: for example if we display a status without training users in what those words mean ("Pending," "In Progress," or "Resolved"), is there any way at all that can work out except by having every average user primevally understand its meaning equalling the job then assigned to him? It is important to communicate clearly: It means that misunderstandings are avoided and ensures residents not only hear but also understand what actions their city is taking on them. In short, the chart reflects very well on the app it is a document clear of obscurities and thus useful in adequately informing users as to their progress to date.

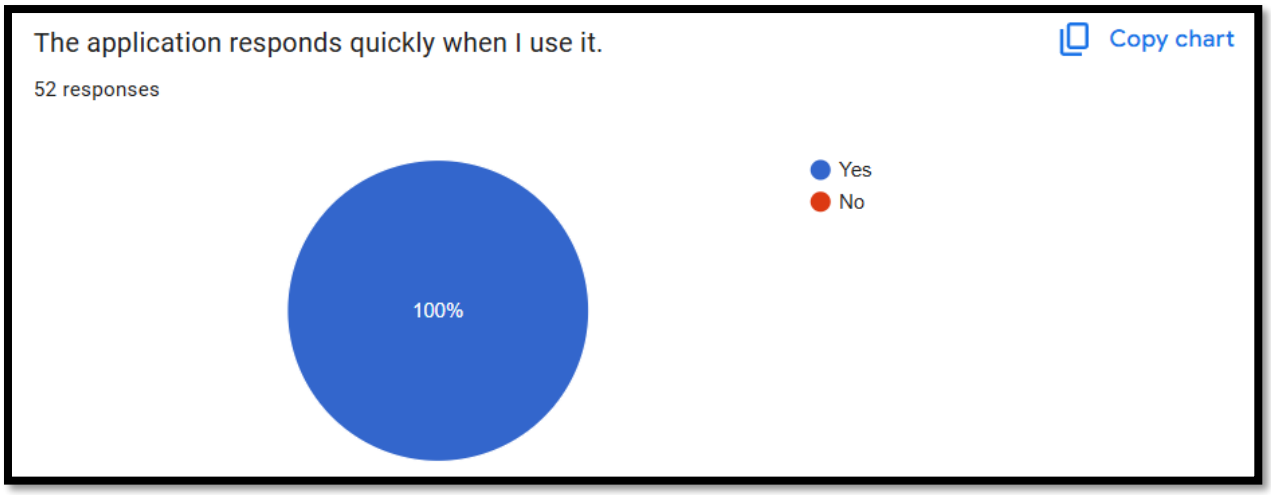


Figure 8.5.10: Question 10

According to figure 8.5.10, in all three cases reviewed (100% agree). This performance measure is crucial because the speed of an application will directly affect user satisfaction and adoption rates; if apps feel slow or if they move furiously about, people often revert back to reporting in traditional ways manually. Fast loading times and smooth transitions are an equally accurate barometer of quality user experience provided.

In general, the entire agreement also suggests that this mobile application is technically reliable and completely hassle-free for users.

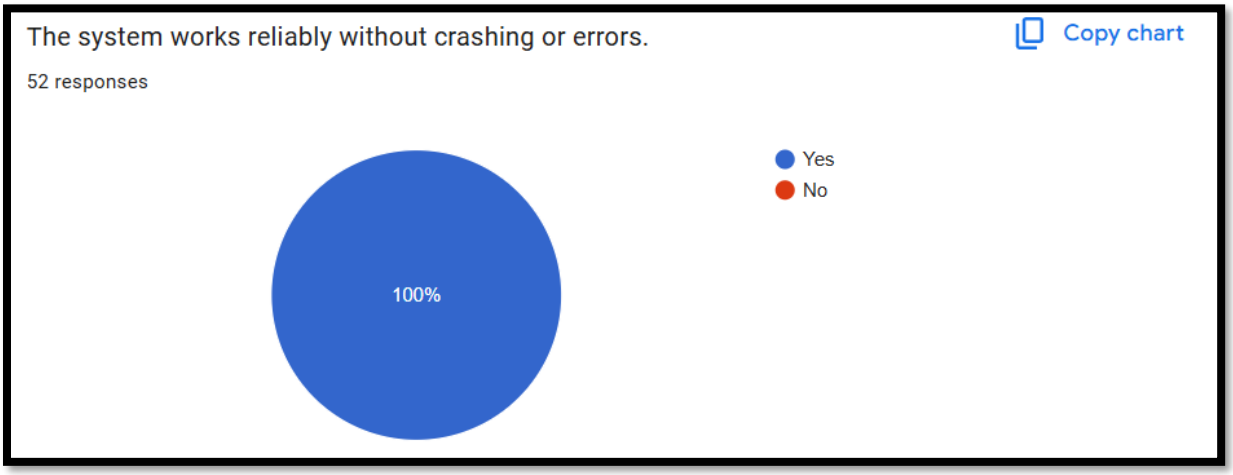


Figure 8.5.11: Question 11

As shown in Figure 8.5.11, The system is so safe that 100% of users that participated in the survey gave it topmarks for not crashing or having error messages. These are important conclusions to be made from any assessment because if technically stable "platform" can win the trust of our users then it must be considered successful. It is very difficult to get the residents involved whether downloading apps or joining outside systems. This will make them cautious of doing anything if the system fails. As the chart shows, yes, the app is firm and sets up a reliable backbone for daily use.

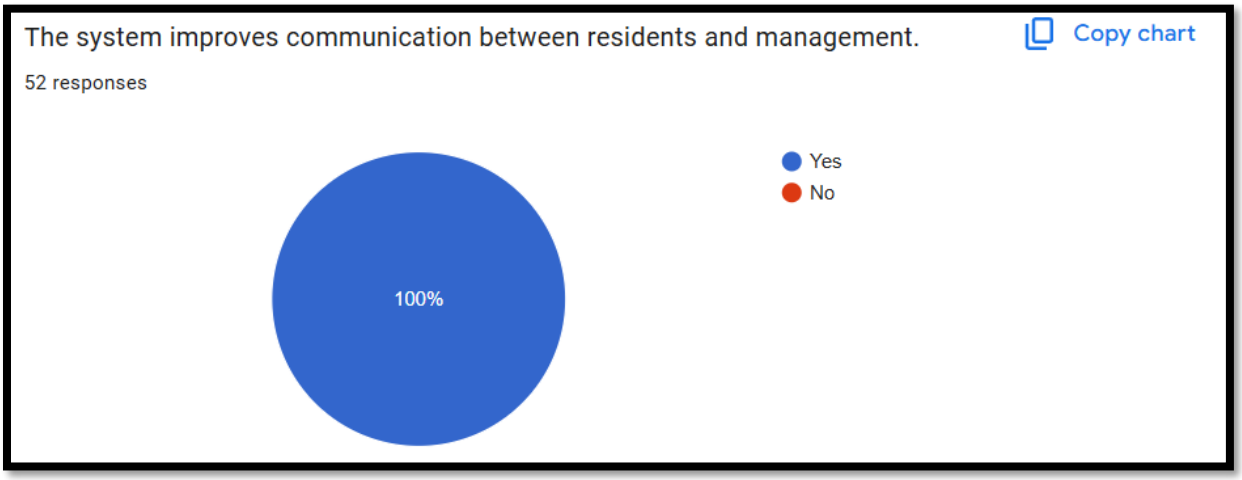


Figure 8.5.12: Question 12

As shown in figure 8.5.12, 100 % of respondents feel that the system has improved communication between residents and management. This is valuable input,because clear information flow is the main purpose of using a digital reporting tool. Word for the kind of guesswork and delays that typifiy traditional property management interactions it sets up a centralised and open channel in which items can be input to the system and then followed up.. In general, the graph says quite a lot it shows that the application indeed improves

on the operational dialogue between what happens in the community and which decisions can be made by the management team.

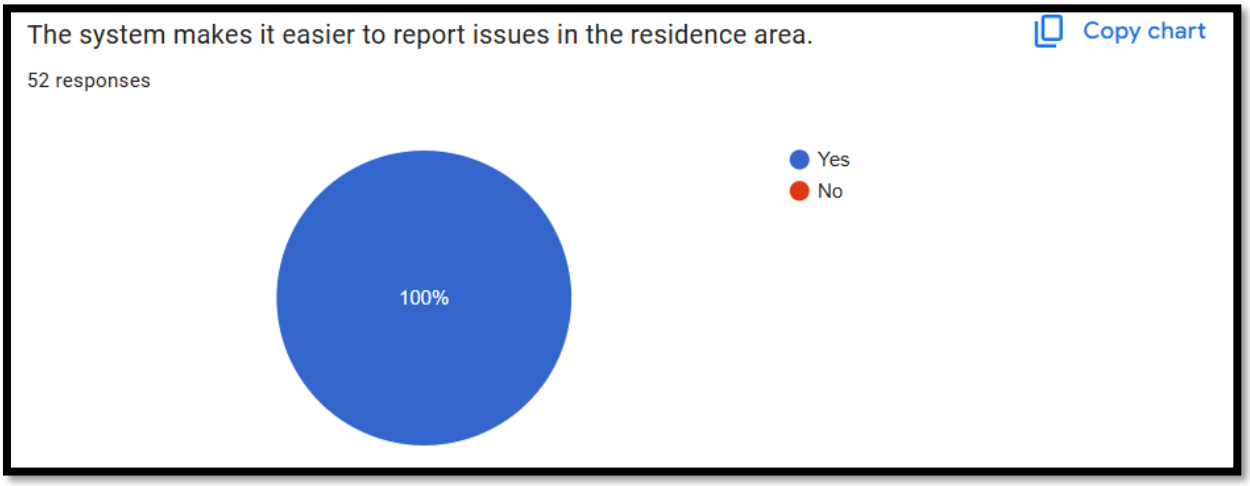


Figure 8.5.13: Question 13

In figure 8.5.13, 100% of the fifty-two respondents agree that contributing to a report in the system makes it easier to report issues in the residence area with unanimous participation. This is also an extremely positive finding for the evaluation, as the main purpose of such a digital solution is to eliminate any inconvenience that recipients may have with traditional reporting methods, such as visiting the management office in person or making phone calls. In fact, by removing the entry cost, using the application allows residents to notify on maintenance or safety issues more readily. Therefore, in general, the chart brings home the fact that the application indeed fulfills its main purpose of making issue logging much more easy and busy-friendly for Shamelin Star community.

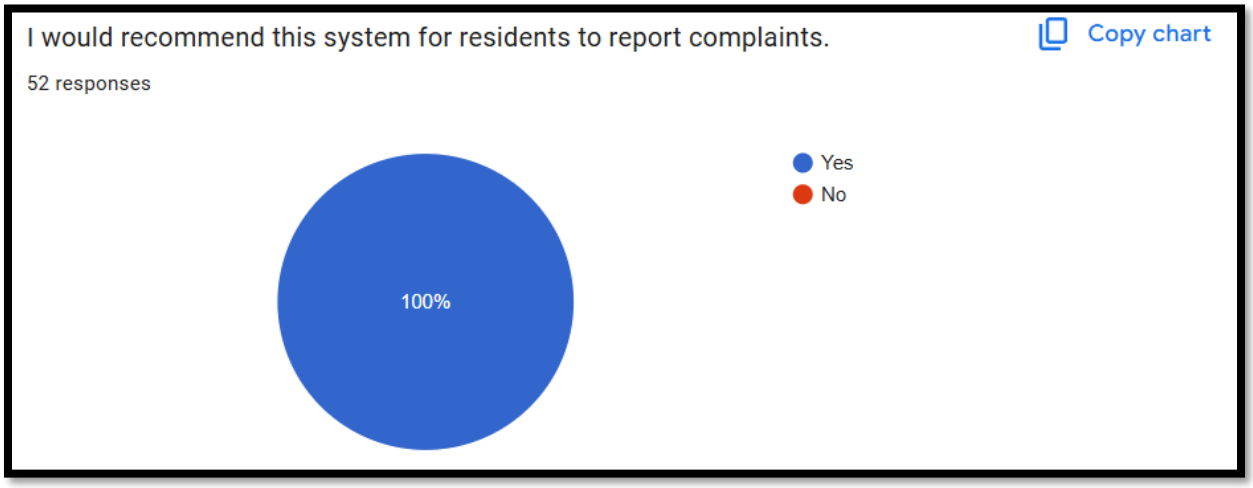


Figure 8.5.14: Question 14

In figure 8.5.14 we can see that the users were 100% unanimous in saying they would recommend this system to residents for reporting complaints. The willingness to recommend is often seen as a key marker

of user loyalty, value perception (both immediate and relative), from this it appears that not only did the residents find the system functional for themselves but also appreciated its utility at the community level. Users endorse destination applications to neighbors when the user is highly confident that this is a useful tool.

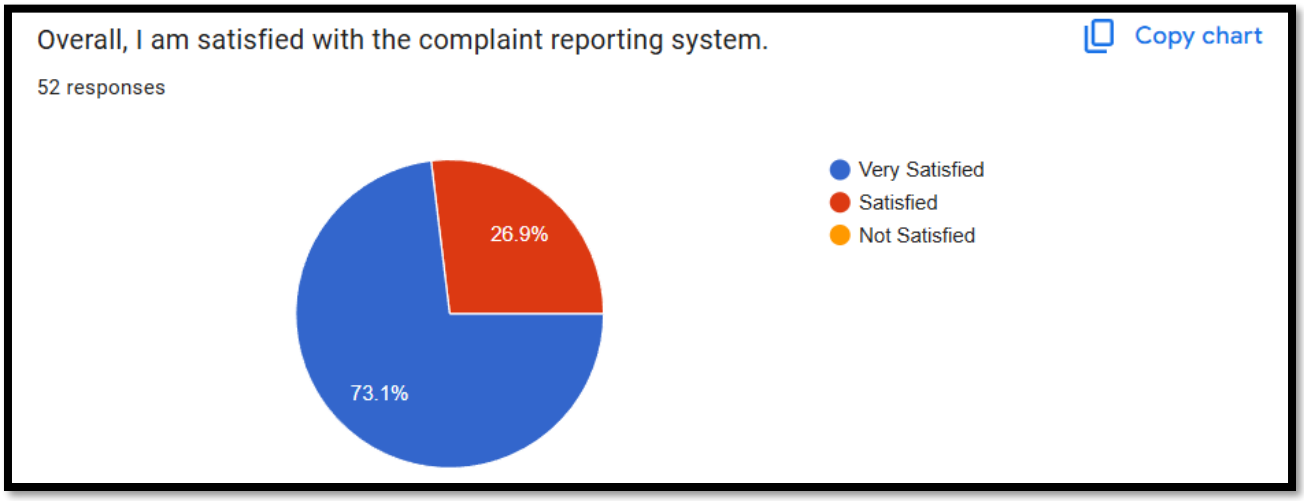


Figure 8.5.15: Question 15

In figure 8.5.15, We can see from a questionnaire that over 70%-73.1% to be exact said they were "very satisfied" with the entire process of reporting a complaint on the basis where 26.9% are described as satisfied and even no-one at all says not pleased. It is an extremely favorable conclusion to the review when all of the pragmatic operating data usability, reliability, speed and communication is pieced together into this final user verdict. Also if the application ticks every user need with 100% positive sentiment achieved even on this basis then its users in the words of the author are getting their cake and eating it too. In all these things together, the system is given a good grade. There's little doubt from results that it is indeed an exceptionally practical, useful system.

8.6 Conclusion

During this chapter, a range of trials were conducted on the Shamelin Star E-Reporting System. Included among these: Unit Testing; the function of one constituent module in isolation (e.g. user login, complaint submission, complaint status change, staff assignment) to itsshaped operation at the component level that is. Integration tests confirmed that the mobile application, web-based management dashboard, and the Firebase back end checkout to match seamlessly. Here, smooth decision exchanges and synchronization allow for real-time transmission of data. Shown by system testing was that this, particular model of an all-integrated but not actually functioning system gets results meeting its the functional and non-functional requirements (including performance, reliability, and usability) as and when it is used under realistic conditions of operation.

Moreover, User Acceptance Testing (UAT) gave residents and management staff direct, real-time verification that the system is one that satisfies their requirements which is suitably customized to cope with residential complaints.After all, the three levels of tests just past have shown that the Shamelin Star E-Reporting System is reliable, stable, and ready to roll. Any issues now to be resolved are minimal indeed, for they havebeen detected.

Resulting from in the future this rather minor Task coming To the system summary- OK, it achieves the following: Improves management of complaints by property owners and could make residence life more convenient. As such, any outstanding issues found in testing are quite trivial and can very easily be addressed as part of future enhancement without breaking the overall functionality of the solution. In conclusion, this system meets overall goals: improves management of complaints and facilitates residents to contact property management.

9 PROJECT MANAGEMENT

9.1 Introduction

The chapter presents which adaptation of the project management approach was used in developing the Shamelin Star E-Reporting System. Project management is the application of knowledge, skills, tools and techniques to project activities to meet the project requirements (Project Management Institute, n.d.). In a software development project, proper planning and scheduling, monitoring schedules and quality, risk management are important so that the system can be completed within scope, time, and quality boundaries to ensure successful completion of each phase (Project Management Institute, 2021).

Shamelin Star E-Reporting System Project Management's focus through FYP 1 and FYP 2, WBS, Gantt Chart tracking, Firebase integration risk, authentication security risk and timeline. FYP 2 focused on implementation of the system, database integration and interface development along with testing and deployment preparation. This chapter describes the project timeline, including the WBS structure of tasks in Gantt chart form with risk management strategies for completing each stage that ensured successful systems delivery.

9.2 Project Schedule

The project schedule is a formal representation of all activities that were done in developing Shamelin Star E-Reporting System. A defined schedule helps in identifying tasks dependencies, estimating their completion time and keeping track of the overall progress. The project schedule was assigned during the FYP 2 period which consists of phase refinements on system design, development, testing and delivery.

The development started with refining the system design, which included designing the system architecture in detail, database designing using Firebase Firestore and frontend for web dashboard and mobile application. After that was system implementation stage in which some of core modules were implemented, developed and tested like user registration, complaint submission, tracking and staff assignment and status update functionalities. In both Web Dashboard and Mobile application, we integrated data for Real-time synchronized.

System testing then was done to make sure implemented functionalities were functioning and stable. Keeping the different components playing nice was achieved through unit testing, integration testing and system testing. Some extra time was spent debugging and any issues which were caused by the Firebase integration or syncing data across platforms

Lastly, the project moved on to documentation, writing reports and readying for final presentation and submission. However, it was important to deliver all functional features with each sprint as well executed in the proposed timeline.

9.2.1 Work Breakdown Structure (WBS)

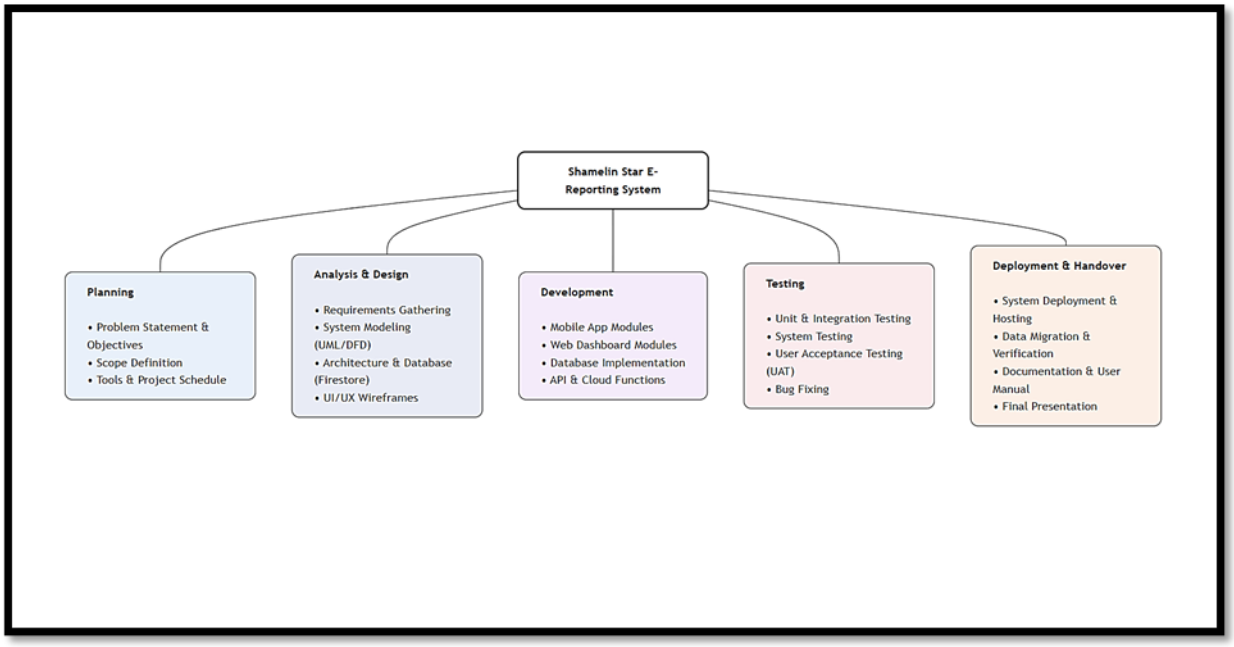


Figure 9.2.1: Work Breakdown Structure for Shamelin Star E-Reporting System

Figure 9.2.1 Work Breakdown Structure (WBS) for the Shamelin Star E-Reporting System The project has five key phases, which are the Planning Phase, Analysis and Design Phase, Development Phase, Testing Phase as well as Deployment and Handover Phases. Each phase is a key deliverable, and each one will be broken down into individual activities needed to complete the project in an orderly manner.

The beginning of Planning involves stating the problem statement, curating objectives for project and its scope along with the schedule & tools used in it. The Analysis and Design phase prepares functional as well as non-functional requirements, system models (UML/DFD diagrams), the architecture of the system with Firestore database structure, wireframes for both mobile application and web dashboard UI/UX. Development phase: For mobile application modules, web dashboard modules, database configuration and APIs & cloud services integration. Unit testing, Integration testing, System testing and User acceptance Testing and fixes to bring the system in line with specifications. Finally comes the Deployment and Handover phase where we focus on deploying the system, configuring its hosting environment, preparing documentation as well as delivering a final presentation.

9.2.2 Gantt Chart

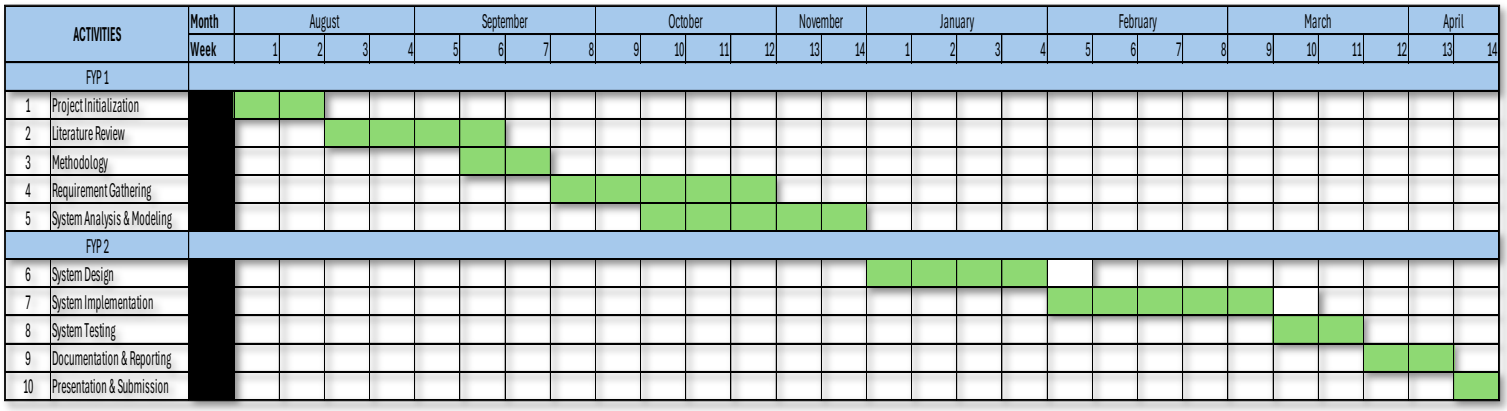


Figure 9.2.2: Gantt Chart for Shamelin Star E-Reporting System Project Timeline

The Gantt chart in Figure 9.2.2 shows the project timeline for the Shamelin Star E-Reporting System development. The phrase below explains the process and time taken to complete tasks across FYP 1 and FYP 2. Activities performed during FYP 1 phase are project initialization, literature review, requirement gathering and system analysis and modeling. These tasks prepared the ground for system implementation in terms of project scope, functionality requirements and system architecture.

FYP 2 was all about implementation and testing phases. System designs are ready, now work begins on mobile application and web dashboard. Then along came Firebase, which made authentication and realtime database sync possible. Also another system testing was performed after the development of a system for the functionality and reliability provision, also preparing documentation finally presenting. The Gantt chart provides a clear outline of the scheduling of tasks and their dependencies, allowing the ability to demonstrate that project progress is being monitored where major milestones are achieved within the proposed period allocated for this academic year.

9.3 Risk Management

Risk management simply means identification, analysis and response to risk that is potential enough to create negative impact on a project. Thus risk management is very crucial across the whole lifecycle of a software development project, getting technical, schedule, security and user-related risks identified early on and strategies in place to mitigate them before they hit criticality. A good risk strategy minimizes uncertainty and maximizes the chances of stability in the project and successful delivery of the system.

Based on that research, a risk register was created to identify potential risks in the course of implementation FYP 2 for the Shamelin Star E-Reporting System. The key risk categories include system integration, hosting and deployment, scheduling, data security and user acceptance. The identified risks with their analysis and mitigation strategies are summarised in Table 9.3.1.

Table 9.3.1: Risk Identification and Mitigation Plan for Shamelin Star E-Reporting System

#	Identification	Description	Risk Analysis	Mitigation Plan
1	Technical / Integration Risk	Integration issues between the web dashboard, mobile application and backend database may cause report submission failures or data inconsistency.	High impact, medium probability	Conduct integration testing early which is use consistent API structure to validate input data, implement error handling and logging. Perform unit and integration testing before deployment.
2	Hosting and Deployment Risk	Deployment issues on the production server may cause system downtime or configuration errors.	High impact, medium probability	Test system on staging server first to ensure proper server configuration. Maintain backup of database and prepare rollback plan in case of failure.
3	Scheduling Risk	Delays due to underestimating time required for debugging, UI refinement and system integration.	High impact, high probability	Use Work Breakdown Structure and Gantt chart to monitor progress, set weekly milestones, prioritise core features and allocate buffer time for testing and bug fixing.
4	Data Security and Privacy Risk	Risk of unauthorised access to management dashboard or exposure of resident complaint data.	High impact, medium probability	Implement role-based access control to enforce secure login authentication and encrypt sensitive data. Use HTTPS during deployment to restrict database access.
5	Usability / User Acceptance Risk	Residents or management staff may find the system difficult to use, leading to low adoption.	Medium impact, medium probability	Conduct user testing sessions to gather feedback from residents and management. Improve interface clarity and simplify navigation and reporting process.

At its most basic level, Risk Management involves identifying, analyzing and responding to risk(s) which can potentially but negatively impact a project. Risk management is thus extremely important also for software development projects in order to identify technical, schedule, security and user-related risks

beforehand and have mitigation strategies in place to avert them before they become critical. The better the risk strategy, the less uncertainty and higher potential for keeping both project and system on track.

A risk register to identify risks was developed for the implementation FYP 2 project of the Shamelin Star E-Reporting System based on that research. Systems integration, hosting and deployment, scheduling, data security user acceptance are all key areas of risk. Table 9.3.1 summarises the identified risks along with their analysis and mitigation strategies.

9.4 Conclusion

To sum up this chapter present the project planning and management activities for Shamelin Star E-Reporting System in FYP 2. It provided the Work Breakdown Structure, Gantt chart and risk management plan to ensure that the development process was properly planned, monitored and controlled. Through specific deadlines and authorisation processes, pre-planning made it easier to devise an architecture plan through system design to prepare for presentation and submission while the risk management aspect also helped this approach by allowing us to predict more technical, timing and security problems long before they could pose a real threat. This project was able to report progress on schedule and with the desired quality, reliability and usability of the developed web dashboard for MRIs and mobile application because systematic planning and continuous controlling was applied.

10 CONCLUSION

10.1Introduction

This chapter highlights the general finding of the Shamelin Star E- Reporting System It outlines project aspects, summaries restriction and limitations encountered heading into development while providing a suggestion to expand future improvement. In Chapter 1, the project is introduced along with a mention of some of the key problems it was designed to address: inefficient manual reporting, issues with tracking complaints and junior management not having structured reports. Due to enhanced data accuracy, accessibility and decision-making efficiency,a new web-based dashboard and mobile app reporting system was proposed. This chapter evaluates how well the developed system achieved its goals and discusses generic lessons learned during development.

10.2 Achievement

In light of all these, the ambitions behind the Shamelin Star E-Reporting System were largely successful. The system even provides a neat, well-maintained and commitment based manner in which all business data gets arranged and analysed. As discussed in Chapter 8, the functional testing and system testing did not reveal any problem with the system functioning according to requirements.

10.2.1 To Digitalize Complaint Reporting

This was achieved by deploying a management web dashboard and mobile app for residents. Complaints can now be made digitally by filling out structured forms, selecting a complaint category, and adding a description with any supporting pictures if available. And as a result that the whole thing will help to avoid complaints in paper format and also it will reduce the chances of loss/ incomplete complaint The answer is yes, each and every complaint that you file gets automatically lodged into a database accross the nation so we can retrieve documentation as needed. It increases effectiveness, reduces admin time and ensures that all the customer complaints are logged accurately and can be traced.

10.2.2 To Provide a Tracking Feature for Residents

Overall, this solution indeed provides a complaint tracking tool which allows the people living in society to view and track their complaints status online. Each complaint has a status: Pending, In progress or Completed. This means that people on these mobile application systems can even go in to login, and see what the app has title set up for, increasing the transparency and also limiting the back follow ups by calls or walking to management office. The resident portal in other words organizes and bring trust to communication between management department and residents.

10.2.3 To Generate Structured Reports for Management

There is a reporting module in the web dashboard too which enables management to build structured and organised reports. By filtering the complaints based on categories, dates or status, management can also understand trends and repeating issues. It shows a systematic of reporting capability to make good decision and planning. Management can identify common issues that are being reported, utilize resources more effectively and watch how staff addresses complaints. It automates report generation where records are collected without labor effort and shifts towards better administration effectiveness.

10.3 Constraint and Limitation

However, during the development process, we encountered some limitations even when our system was successfully implemented. The following constraints were identified:

10.3.1 Time Constraint

The project lasted for two FYP semesters, which was the timeframe we had available. This limited the extent of advanced functionality that could be deployed, reduced the capacity for prolonged testing and long term system evaluation.

10.3.2 Technical and Hosting Limitation

System is limited by hosting environment which may impact scalability and performance. User traffic or data volume may require upgrades and system optimisation in the future.

10.3.3 System Scalability Limitation

This is meant for small to medium use. Additional improvements in the database performance and system architecture might be required for large-scale deployment.

10.4 Future Work and Recommendation

To further enhance the Shamelin Star E-Reporting System, several improvements are recommended:

10.4.1 Mobile Application Enhancement

In the future, they said, push notifications could automatically let residents know when their complaint status changes. This would increase responsiveness and user interaction.

10.4.2 Integration with Notification Systems

Alternatively, users of the application will have to log in frequently in order to receive such information, so integrating with email or SMS notification systems could ease communication lag this way residents would be updated promptly regardless of whether they are currently signed into the application.

10.4.3 System Scalability and Security Improvement

Sections provide specific recommendations on strengthening the security features, implementing role-based access control enhancements and optimising database performance so that these can support higher numbers of users and complaints in future versions.

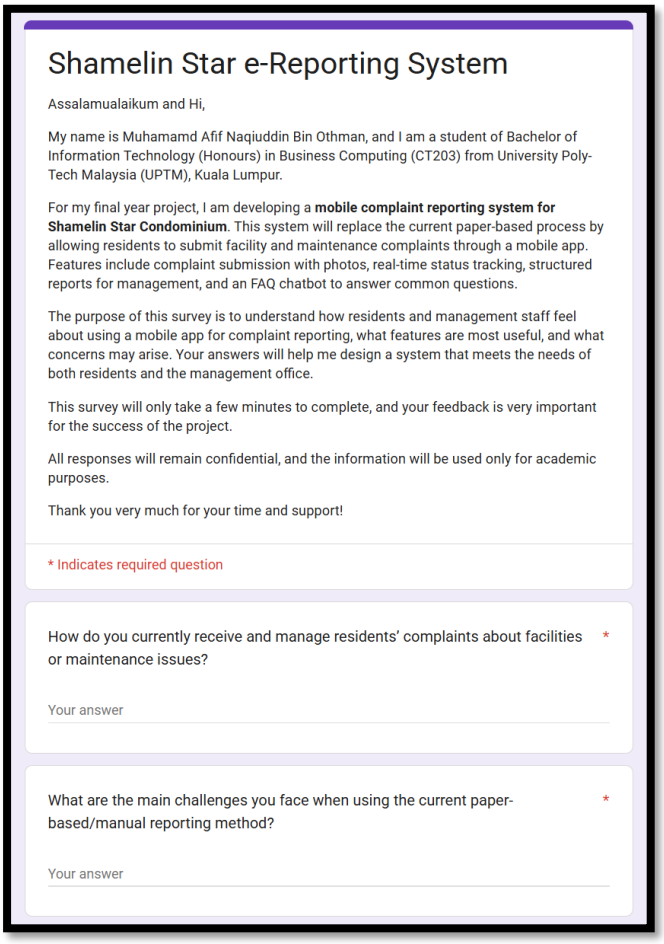
10.5 Conclusion

To summarize, the Shamelin Star E-Reporting System works to its main objectives of reporting complaints digitally, establishing a tracking system Accessible to the residents and generating structured reports for management. This system offers a speedy, clear-cut and organiser digital replacement of the manual forms.

System analysis, database design, mobile and web development, testing and project management during the entire process brought valuable experience thanks to this project. Although some limitations did arise along the way, they were informative growth aspects that underlined points where there was further room for optimisation. Conclusively, the system has significant potential as it matures and is adapted for real use to streamline complaints for residents over time with all stakeholders, providing better quality service for Shamelin Star residents.

Appendix A – Requirements Specification Document

- **Interview (Pre-Development)**



Shamelin Star e-Reporting System

Assalamualaikum and Hi,

My name is Muhamamd Afif Naqiuddin Bin Othman, and I am a student of Bachelor of Information Technology (Honours) in Business Computing (CT203) from University Poly-Tech Malaysia (UPTM), Kuala Lumpur.

For my final year project, I am developing a **mobile complaint reporting system for Shamelin Star Condominium**. This system will replace the current paper-based process by allowing residents to submit facility and maintenance complaints through a mobile app. Features include complaint submission with photos, real-time status tracking, structured reports for management, and an FAQ chatbot to answer common questions.

The purpose of this survey is to understand how residents and management staff feel about using a mobile app for complaint reporting, what features are most useful, and what concerns may arise. Your answers will help me design a system that meets the needs of both residents and the management office.

This survey will only take a few minutes to complete, and your feedback is very important for the success of the project.

All responses will remain confidential, and the information will be used only for academic purposes.

Thank you very much for your time and support!

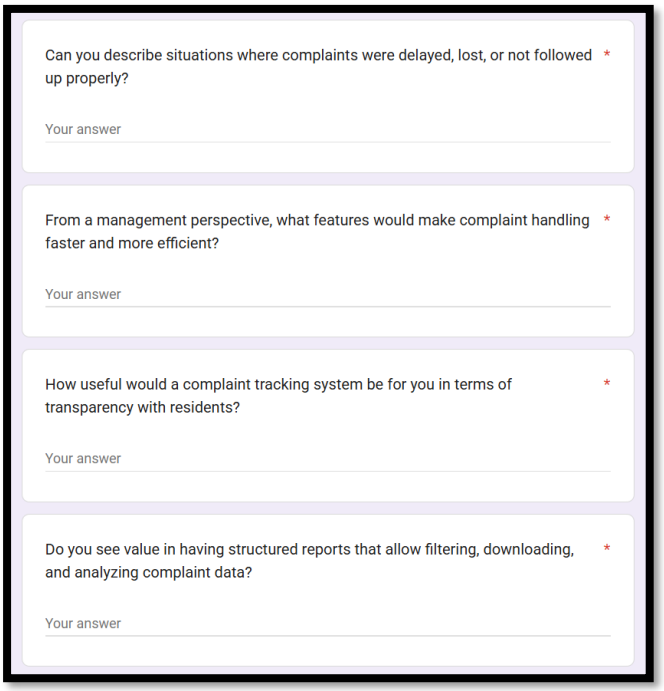
* Indicates required question

How do you currently receive and manage residents' complaints about facilities or maintenance issues? *

Your answer

What are the main challenges you face when using the current paper-based/manual reporting method? *

Your answer



Can you describe situations where complaints were delayed, lost, or not followed up properly? *

Your answer

From a management perspective, what features would make complaint handling faster and more efficient? *

Your answer

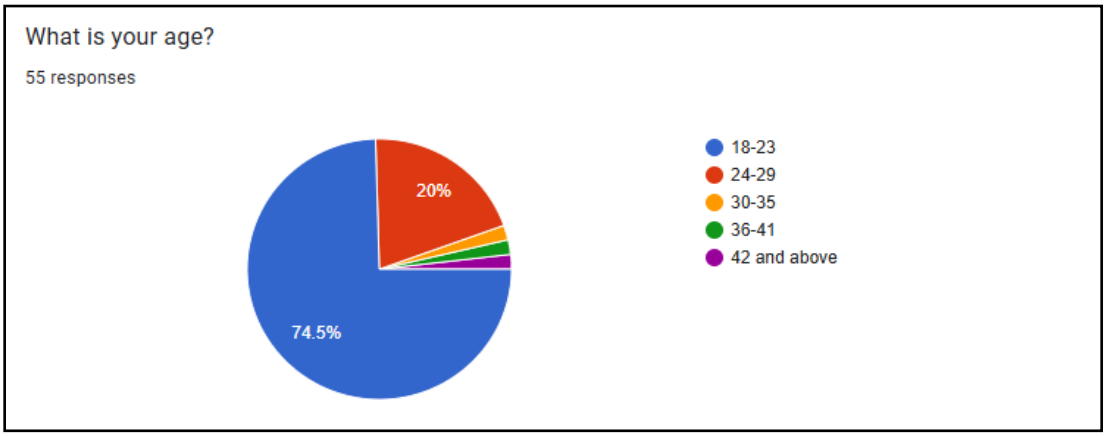
How useful would a complaint tracking system be for you in terms of transparency with residents? *

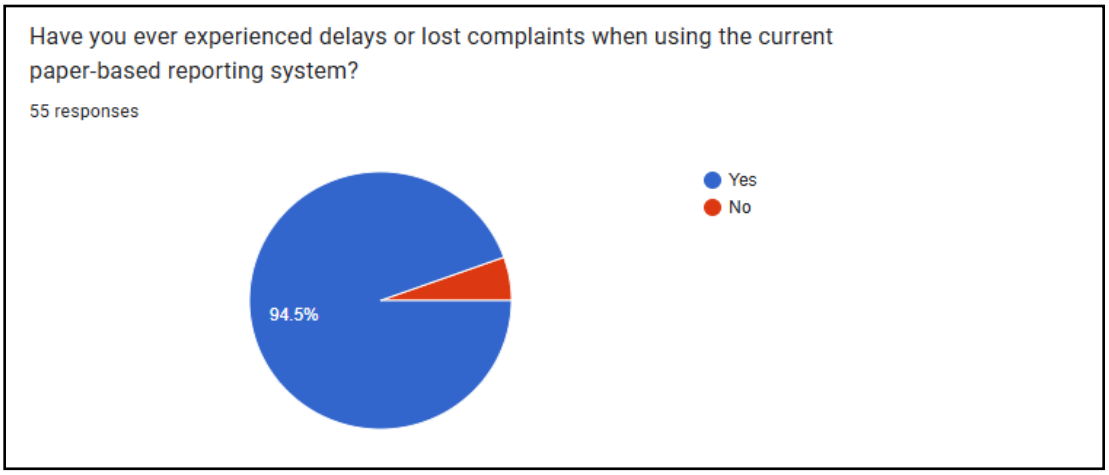
Your answer

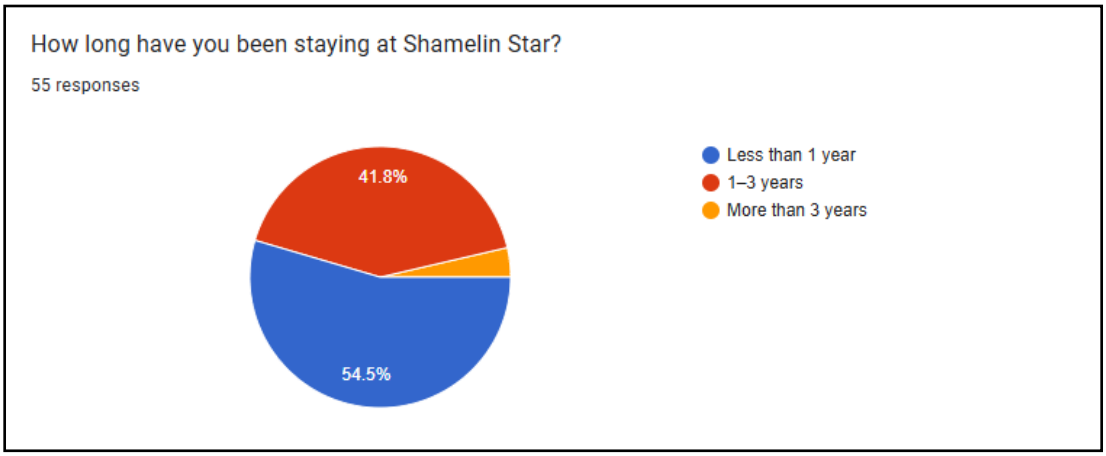
Do you see value in having structured reports that allow filtering, downloading, and analyzing complaint data? *

Your answer

- **Questionnaire (Pre-Development)**

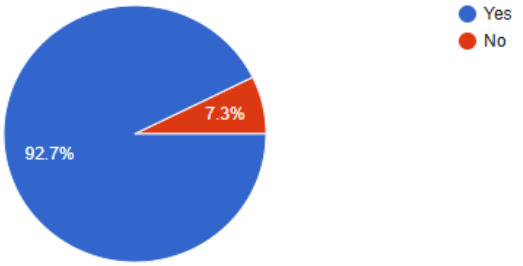






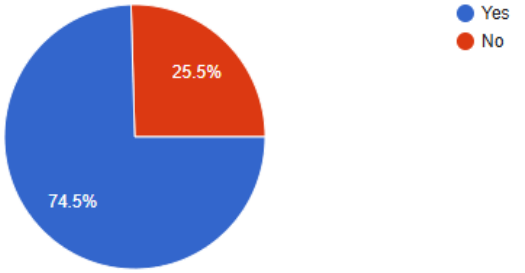
Do you find it difficult to know the status of your complaint once it has been submitted?

55 responses



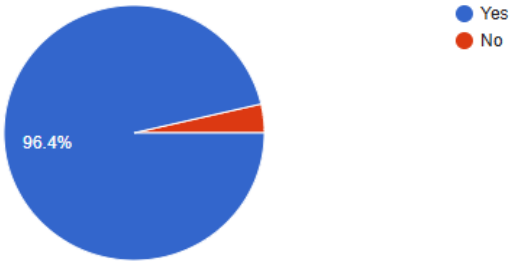
Do you think the management currently lacks clear reports to analyze recurring issues?

55 responses



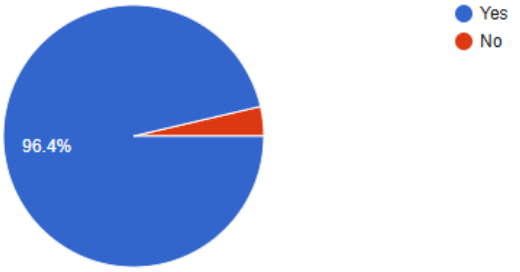
Would you find it more convenient to submit complaints through a mobile application instead of paper forms?

55 responses



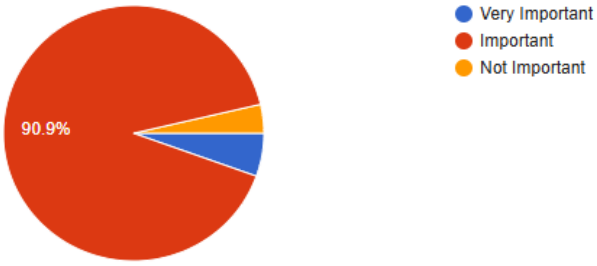
Would a tracking feature showing your complaint status (Submitted → In Progress → Completed) improve your satisfaction with management?

55 responses



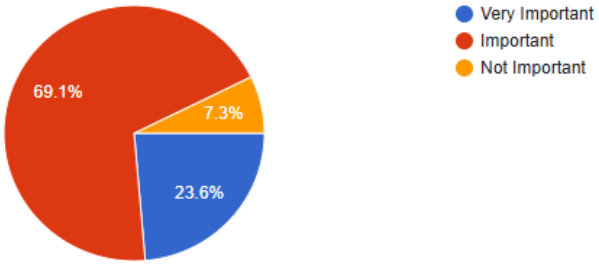
How important would a photo attachment feature be when submitting complaints?

55 responses



How important is it for you to have access to your previous complaint history within the app?

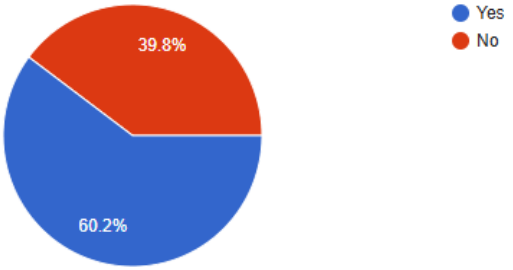
55 responses



Would you use a chatbot in the mobile app to answer frequently asked questions about complaint reporting and facilities?

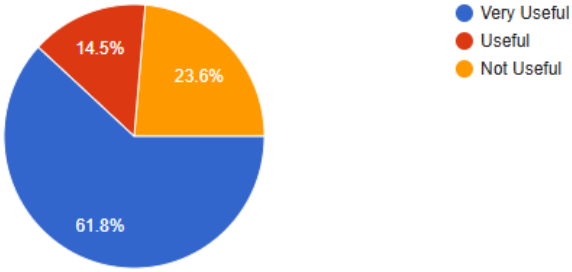
Copy chart

55 responses



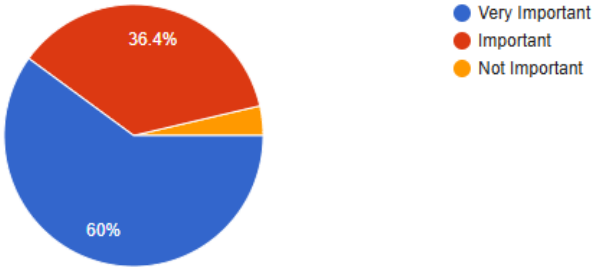
How useful would a real-time complaint status update feature be to you?

55 responses



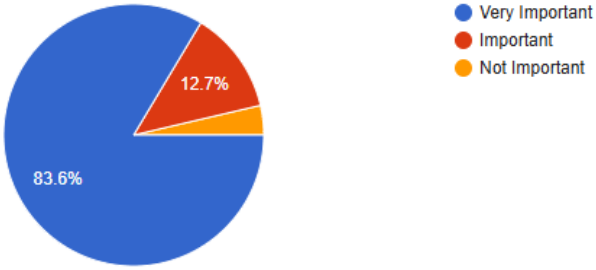
For you, how important is ease of use (simple and user-friendly interface) in such an app?

55 responses



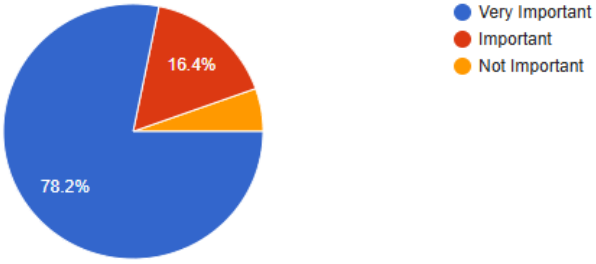
How important is fast performance and quick response time for the app?

55 responses



How important is data security and privacy for you when submitting complaints?

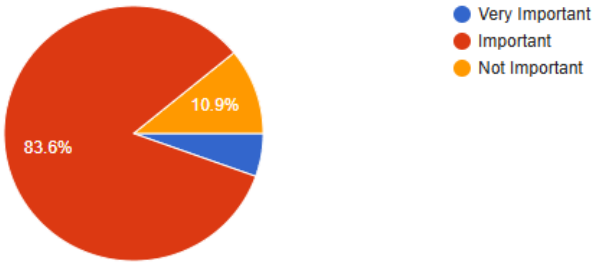
55 responses



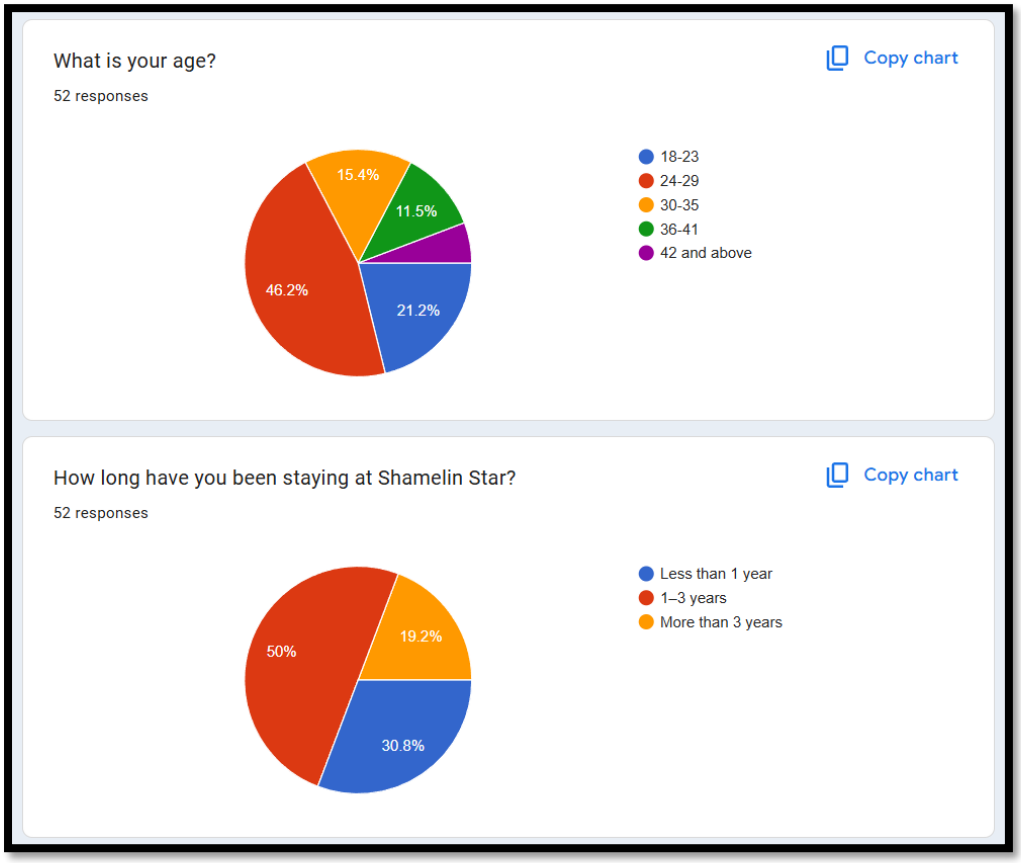
How important is system reliability (app available without frequent downtime) to you?

Copy chart

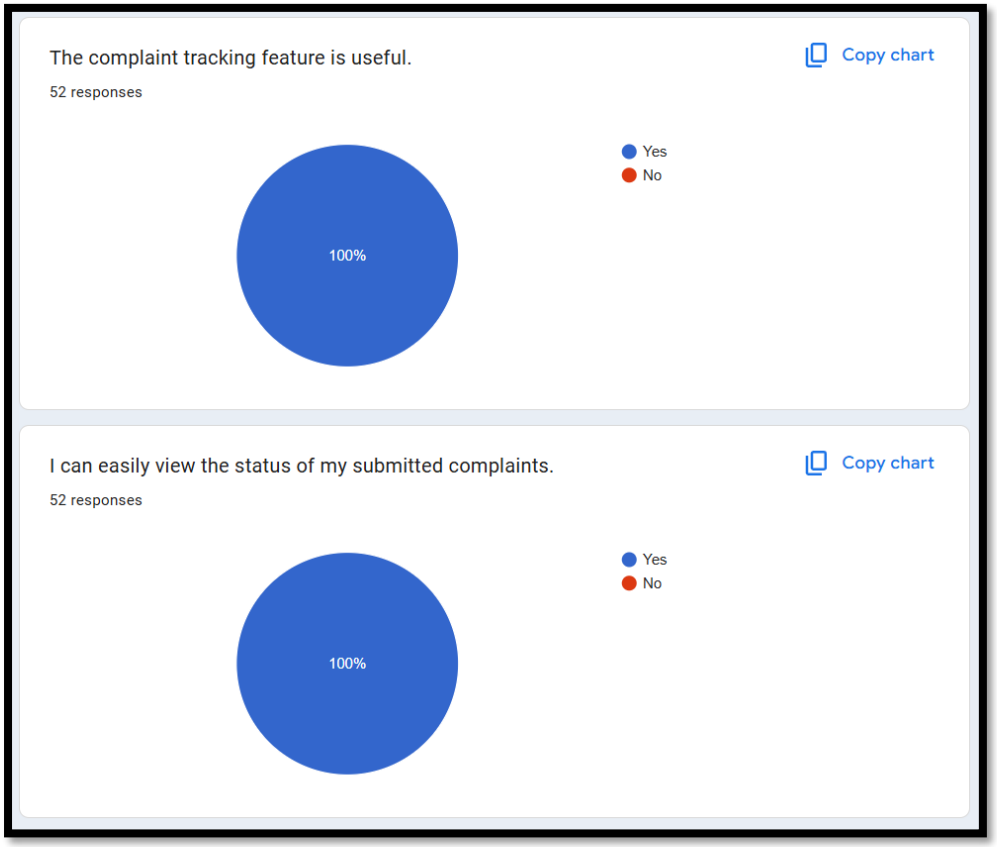
55 responses



- **Questionnaire (Post-Development)**



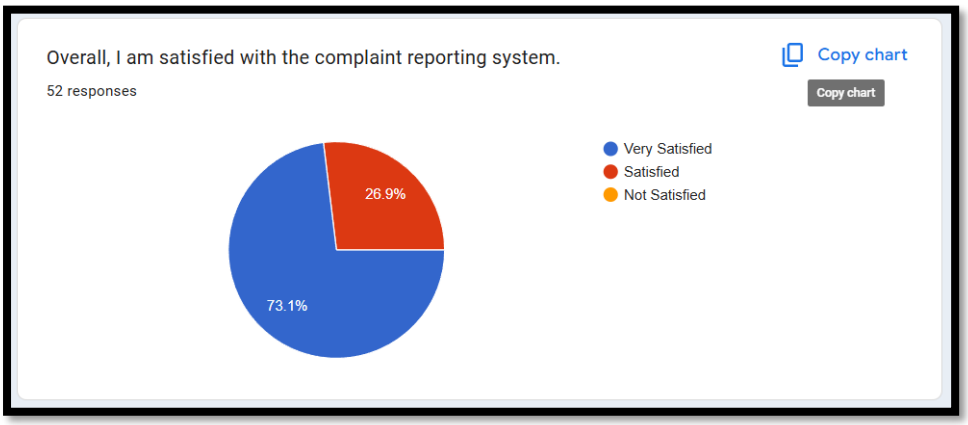

The complaint submission process is fast and straightforward.
Copy chart
52 responses
Yes
No
100%
The system successfully stores my complaint after submission.
Copy chart
52 responses
Yes
No
100%


The complaint tracking feature is useful.
Copy chart
52 responses
Yes
No
100%
I can easily view the status of my submitted complaints.
Copy chart
52 responses
Yes
No
100%


The status updates are clear and understandable.
Copy chart
52 responses
Yes
No
100%
Shamelin Star e-Reporting System
The application responds quickly when I use it.
Copy chart
52 responses
Yes
No
100%


The system works reliably without crashing or errors.
Copy chart
52 responses
Yes
No
100%
The system improves communication between residents and management.
Copy chart
52 responses
Yes
No
100%

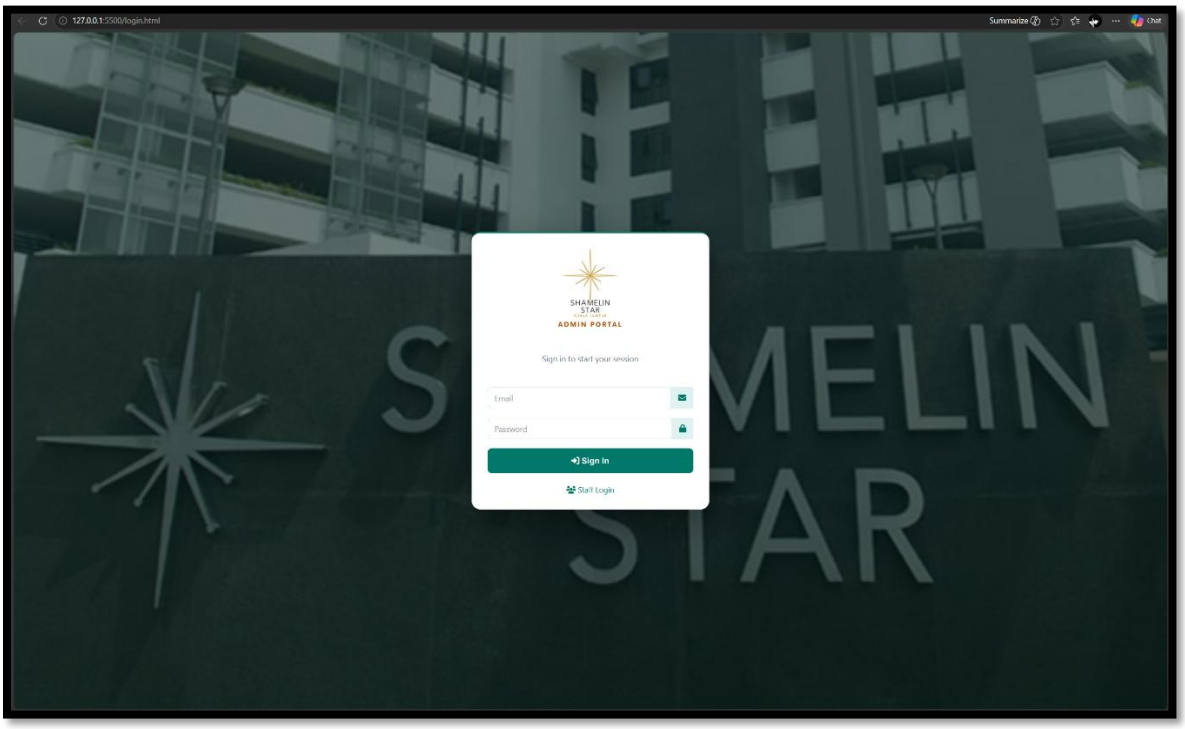

The system makes it easier to report issues in the residence area.
Copy chart
52 responses
Yes
No
100%
I would recommend this system for residents to report complaints.
Copy chart
52 responses
Yes
No
100%


Overall, I am satisfied with the complaint reporting system.
Copy chart
Copy chart
52 responses
Very Satisfied
Satisfied
Not Satisfied
26.9%
73.1%

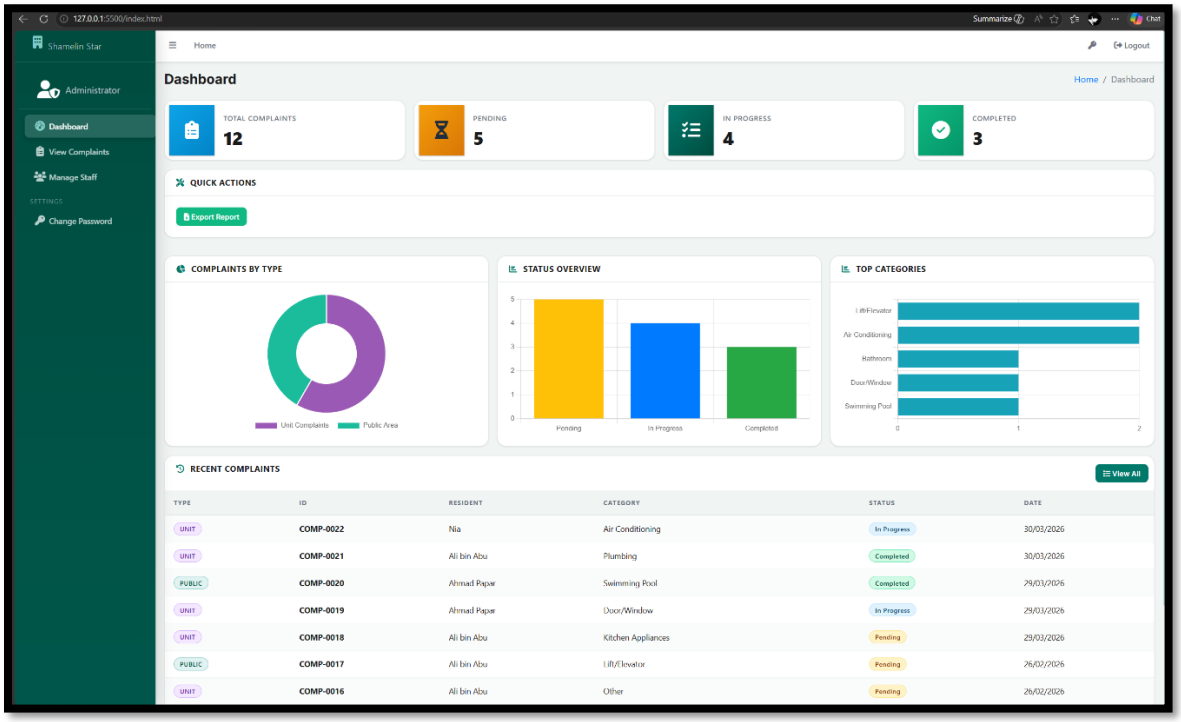
Appendix B – User Manual

1. Resident User Manual

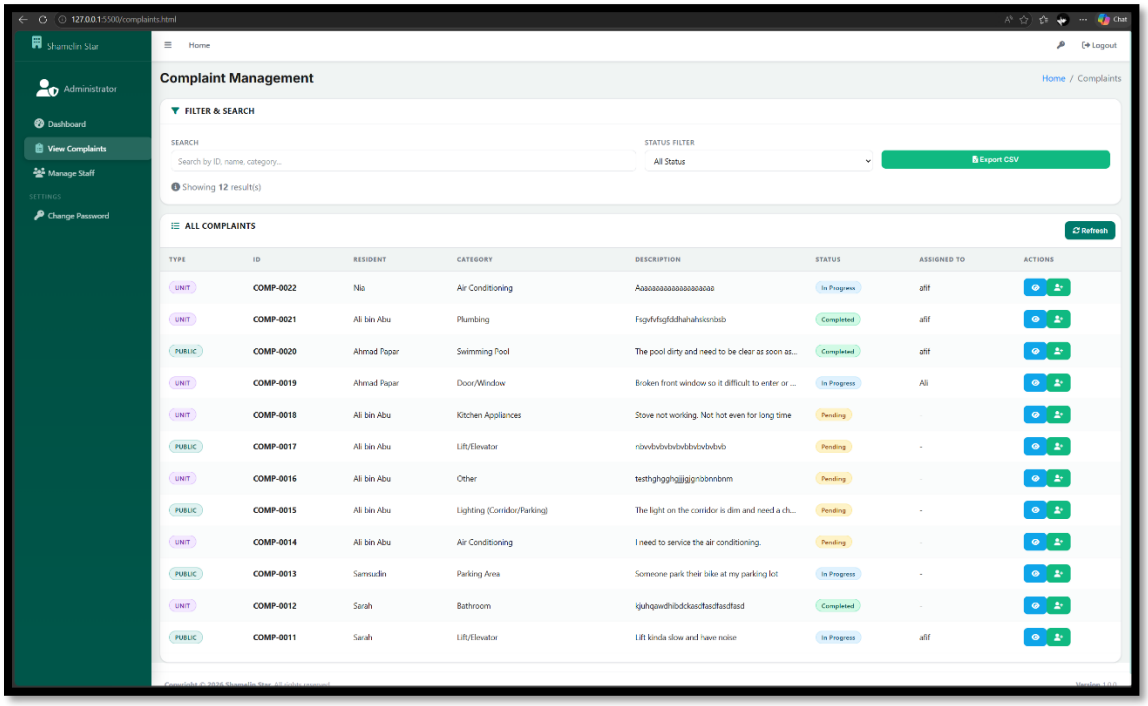
2. Admin User Manual



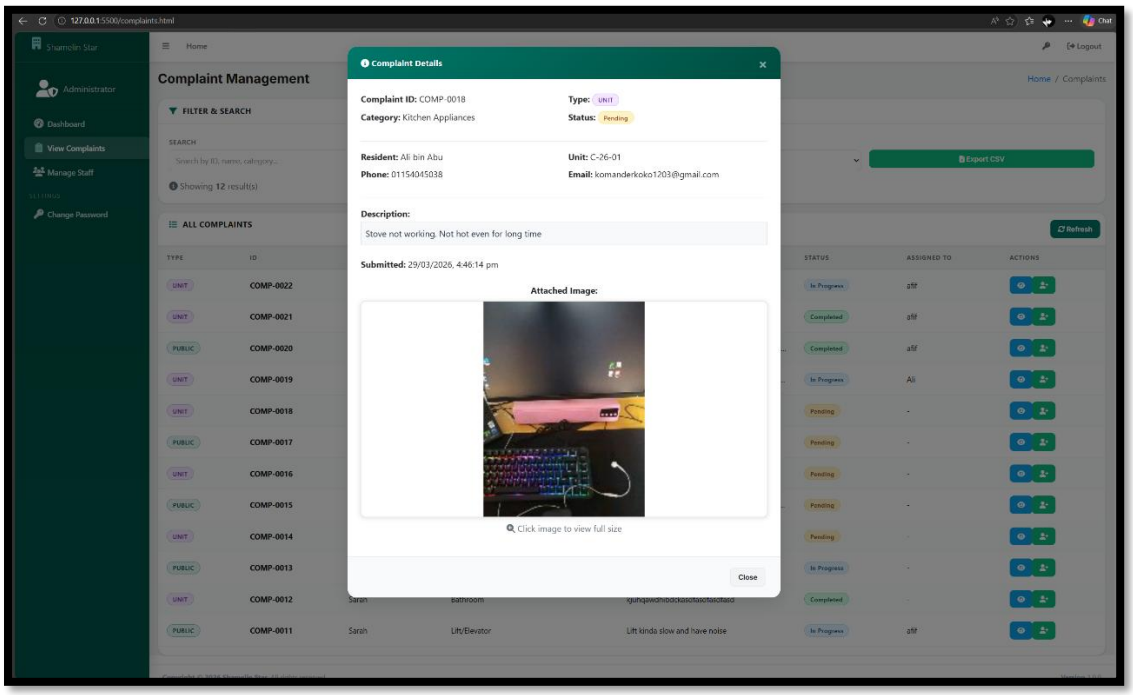
- Admin login to website



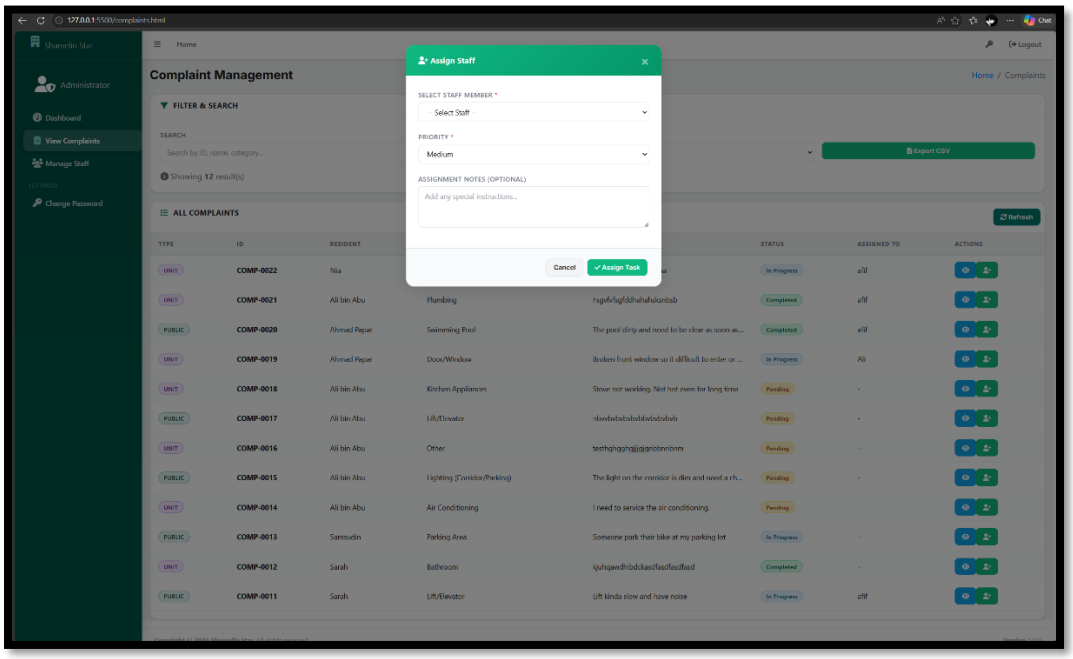
- Admin can check all analysis and reports



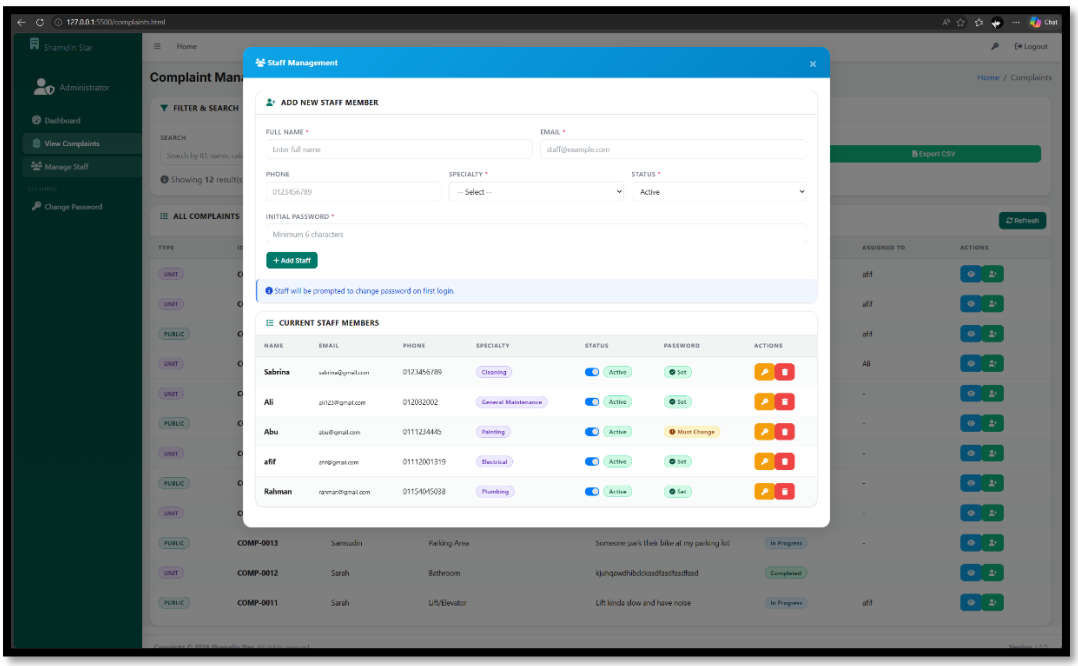
- All the reports here to be looked and assign



- Admin look in details of the complaints

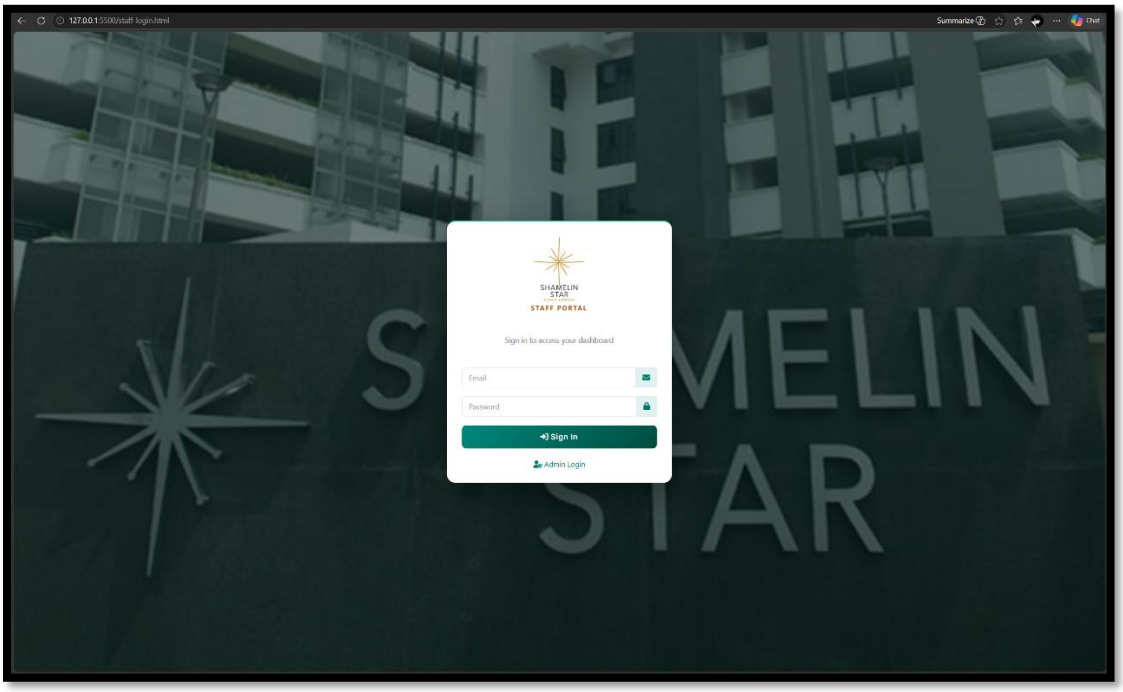


- Admin assign the complaint to staff

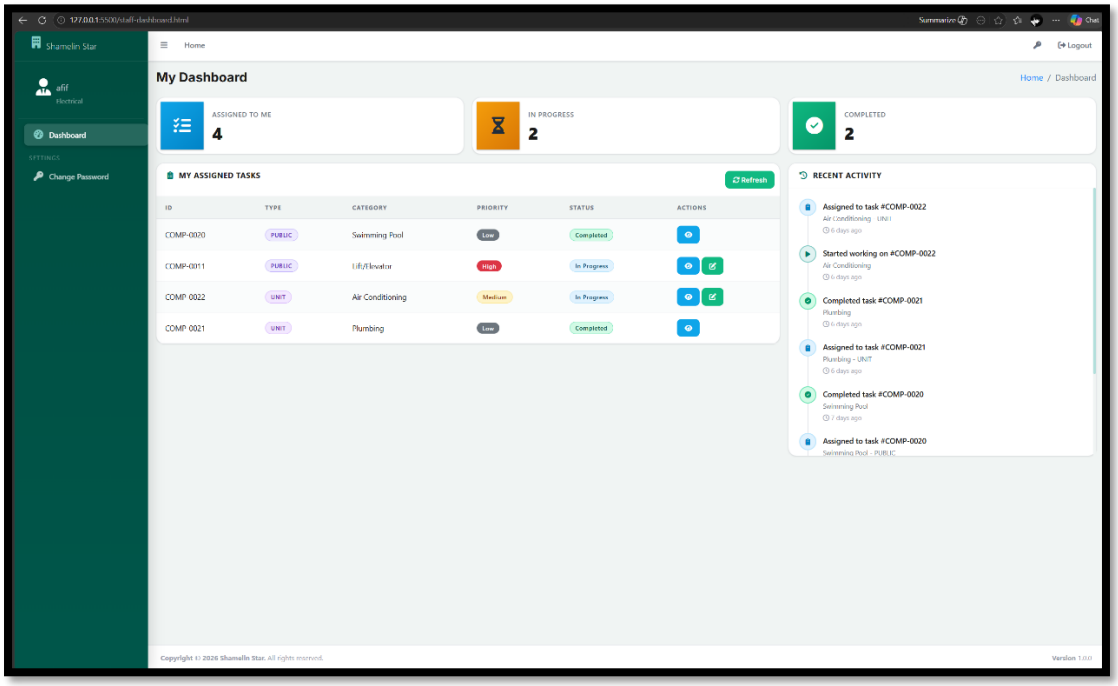


- Admin make account for staff

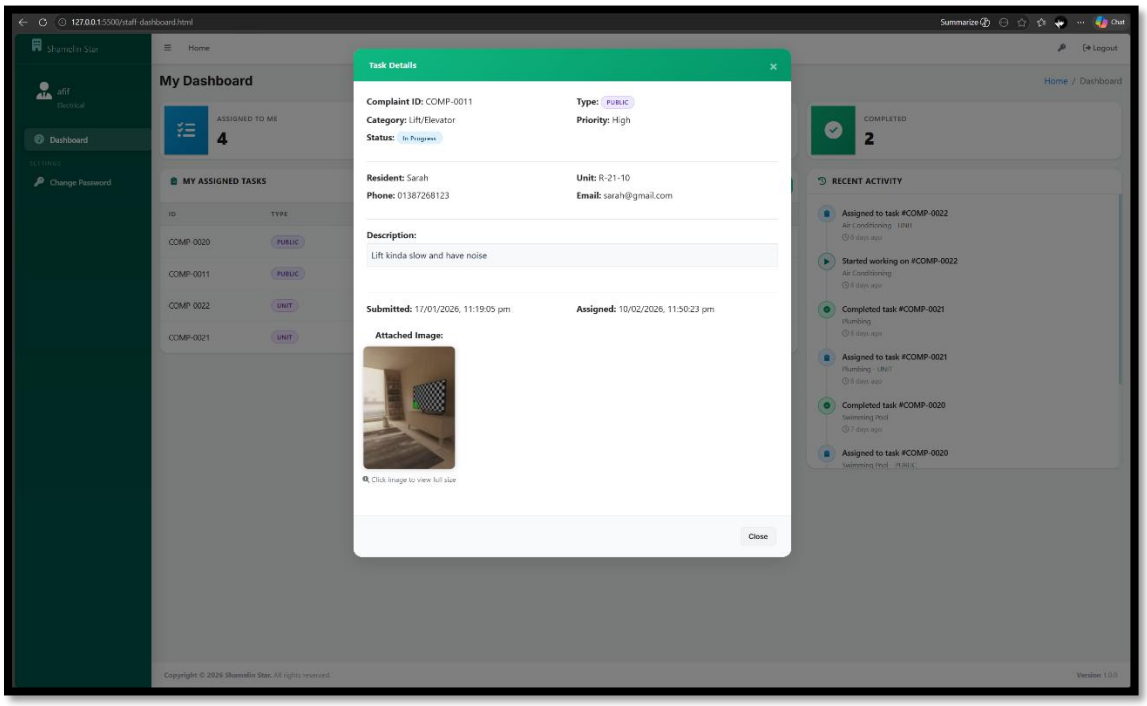
3. Staff User Manual



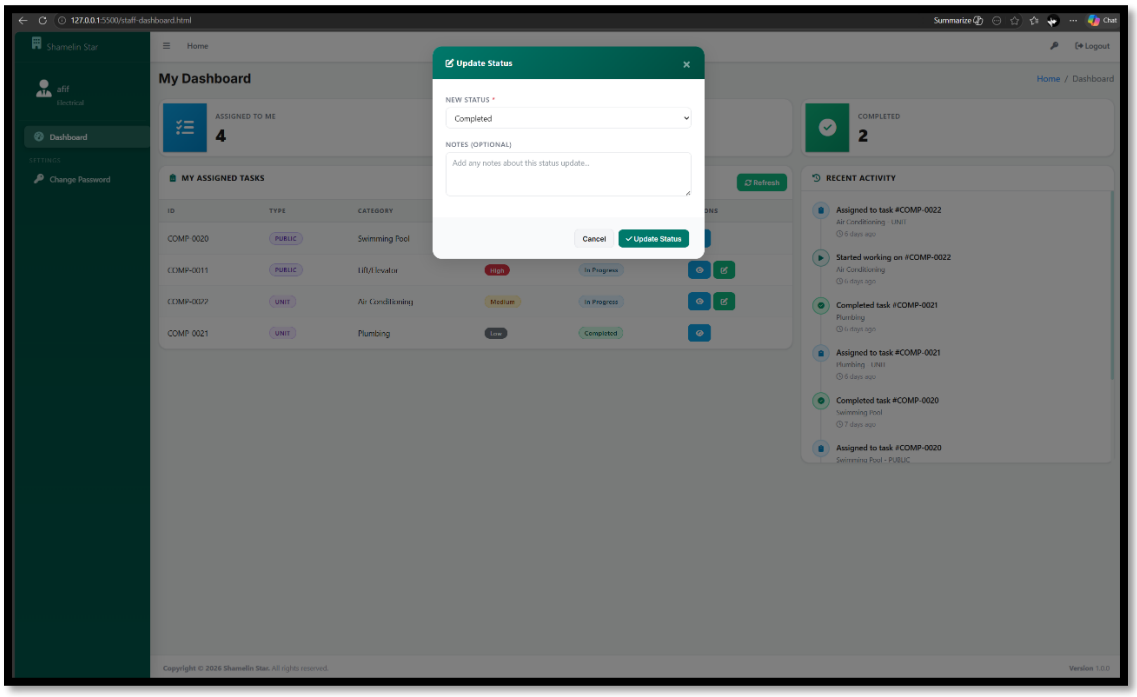
- Staff login into their respective site



- Staff should look for assign complaints to be solve



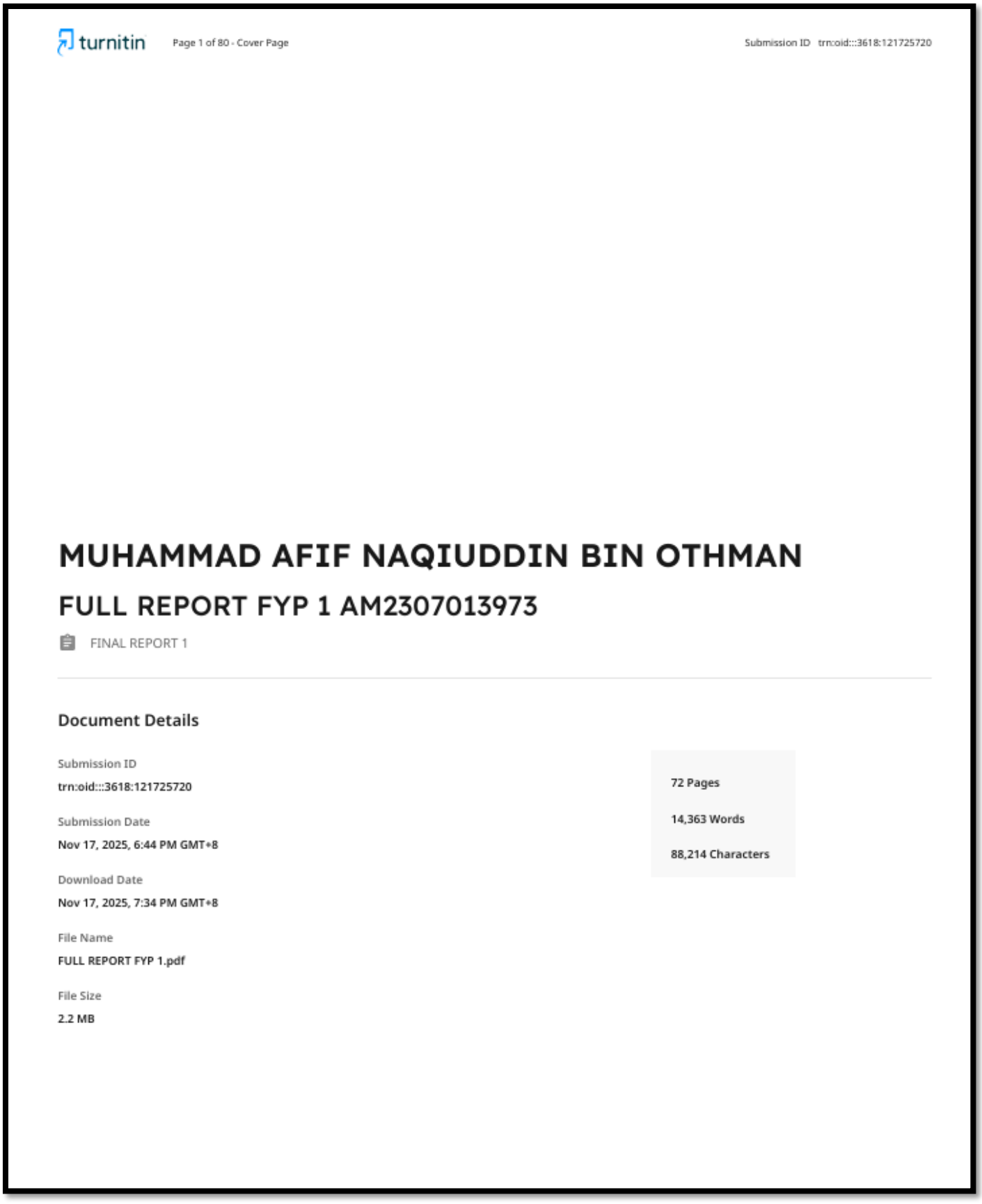
- Staff look in details for the assign complaints



- Staff update the complaints status

Appendix C – Turnitin Result

1. FYP 1 Result



turnitin Page 1 of 80 - Cover Page Submission ID trn:oid:::3618:121725720

MUHAMMAD AFIF NAQIUDDIN BIN OTHMAN

FULL REPORT FYP 1 AM2307013973

FINAL REPORT 1

Document Details

Submission ID
trn:oid:::3618:121725720

Submission Date
Nov 17, 2025, 6:44 PM GMT+8

Download Date
Nov 17, 2025, 7:34 PM GMT+8

File Name
FULL REPORT FYP 1.pdf

File Size
2.2 MB

72 Pages

14,363 Words

88,214 Characters

turnitin Page 2 of 73 - AI Writing Overview Submission ID trn:oid:::1:3414375227

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they 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.

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

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

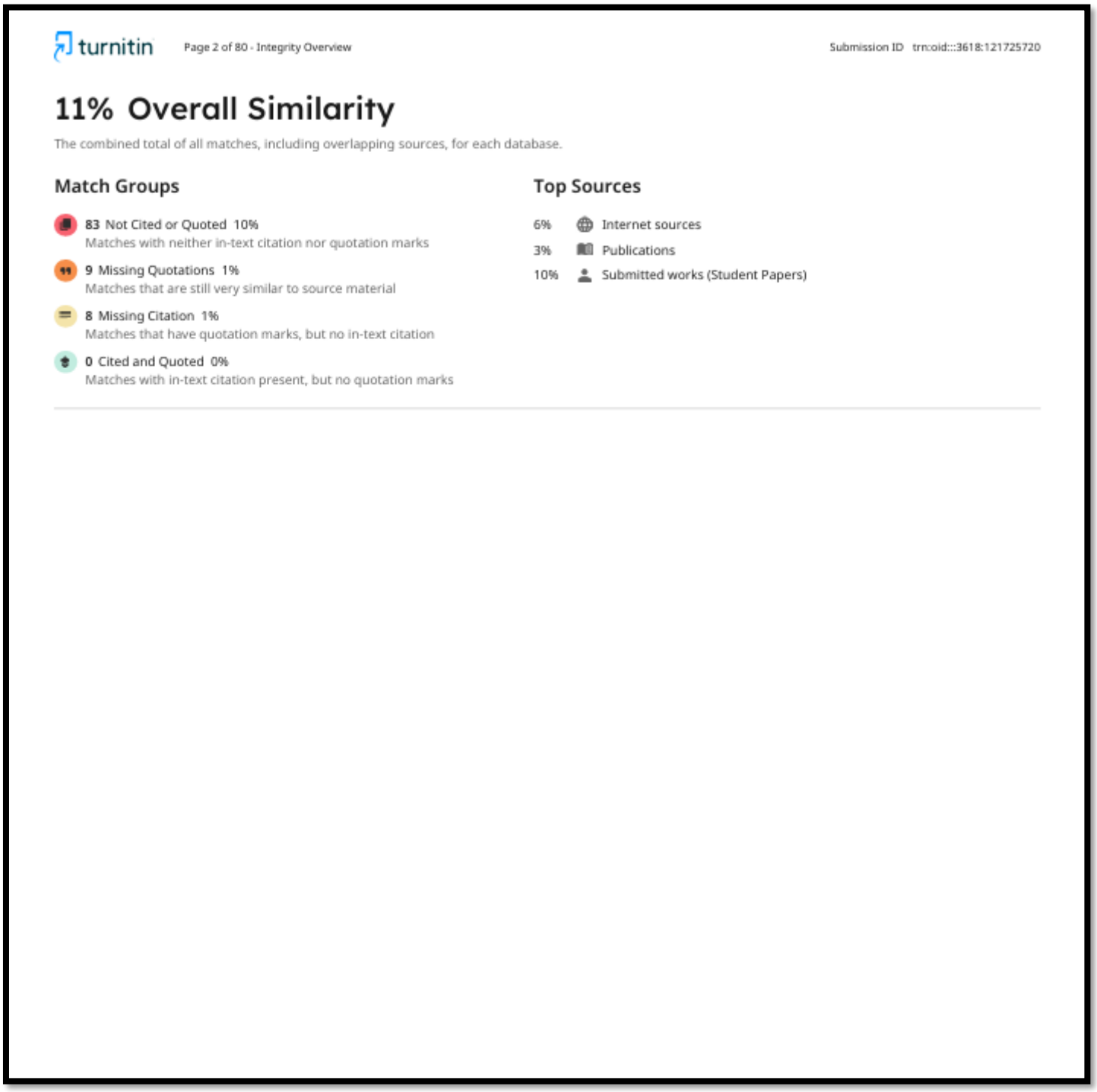
False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



turnitin Page 2 of 80 - Integrity Overview Submission ID trn:oid:::3618:121725720

11% Overall Similarity

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

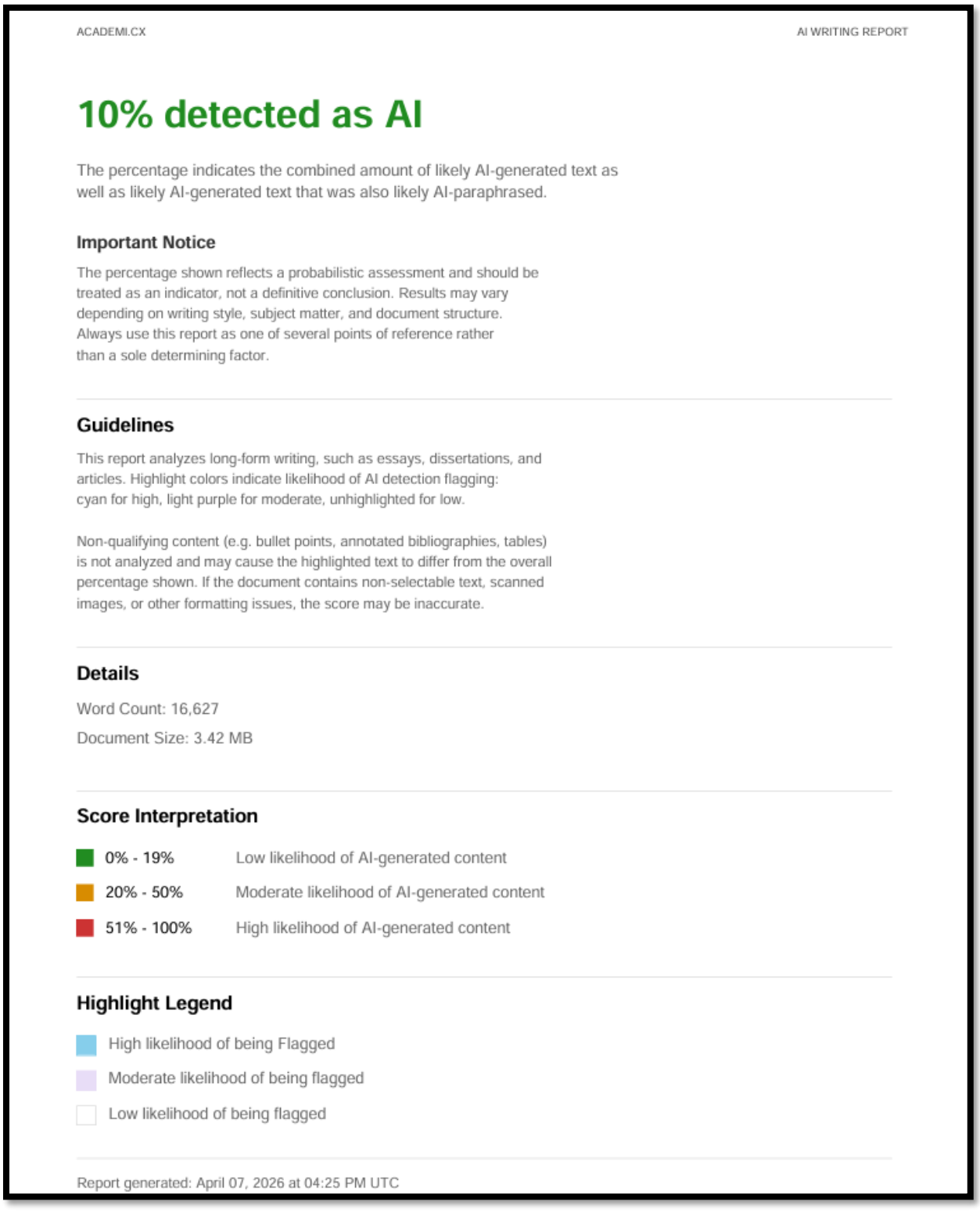
Match Groups

- **83** Not Cited or Quoted 10%
  Matches with neither in-text citation nor quotation marks
- **9** Missing Quotations 1%
  Matches that are still very similar to source material
- **8** Missing Citation 1%
  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

6% Internet sources
3% Publications
10% Submitted works (Student Papers)

2. FYP 2 Result



ACADEMI.CX AI WRITING REPORT

10% detected as AI

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

Important Notice

The percentage shown reflects a probabilistic assessment and should be treated as an indicator, not a definitive conclusion. Results may vary depending on writing style, subject matter, and document structure. Always use this report as one of several points of reference rather than a sole determining factor.

Guidelines

This report analyzes long-form writing, such as essays, dissertations, and articles. Highlight colors indicate likelihood of AI detection flagging: cyan for high, light purple for moderate, unhighlighted for low.

Non-qualifying content (e.g. bullet points, annotated bibliographies, tables) is not analyzed and may cause the highlighted text to differ from the overall percentage shown. If the document contains non-selectable text, scanned images, or other formatting issues, the score may be inaccurate.

Details

Word Count: 16,627

Document Size: 3.42 MB

Score Interpretation

0% - 19%	Low likelihood of AI-generated content
20% - 50%	Moderate likelihood of AI-generated content
51% - 100%	High likelihood of AI-generated content

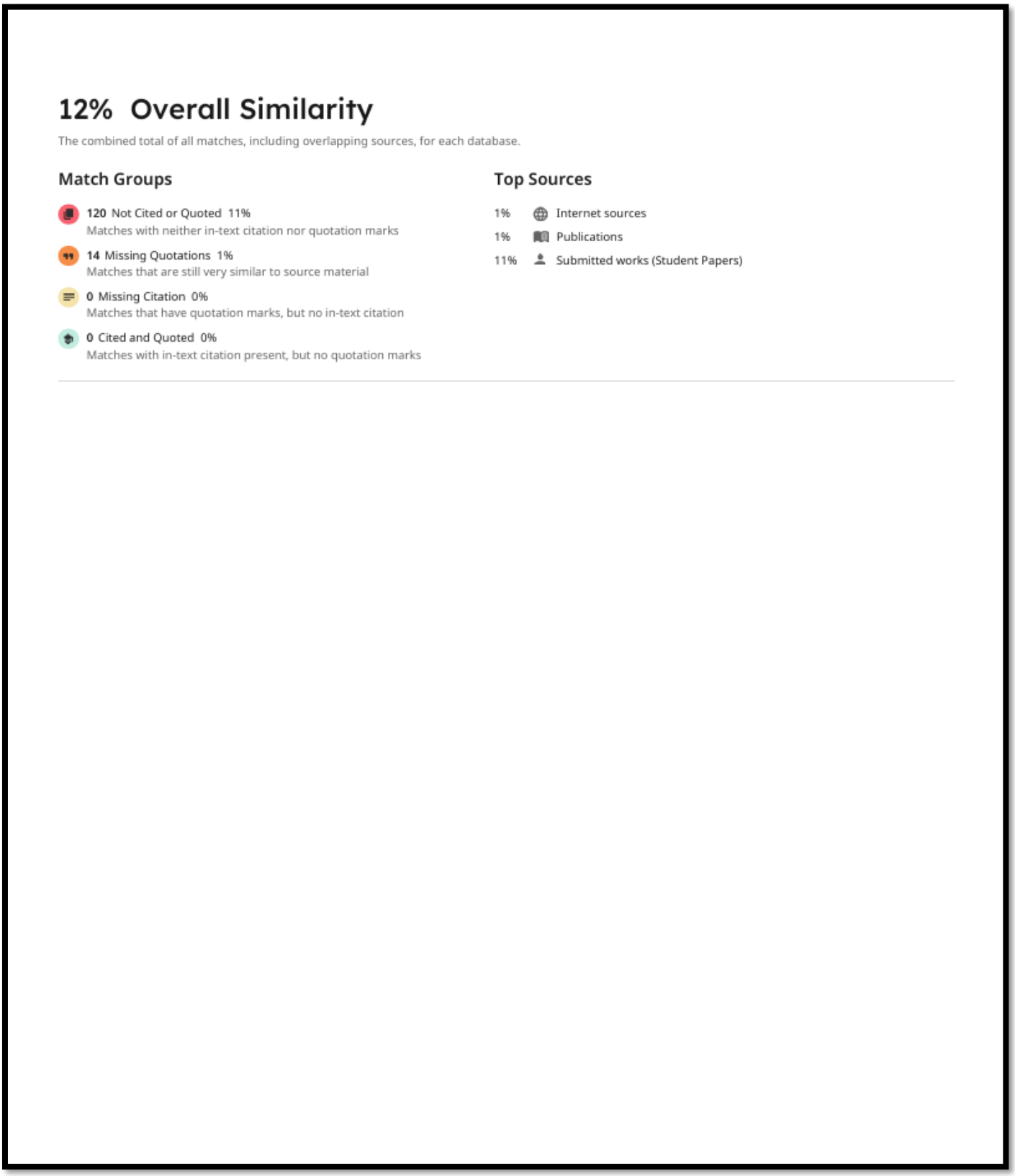
Highlight Legend

High likelihood of being Flagged

Moderate likelihood of being flagged

Low likelihood of being flagged

Report generated: April 07, 2026 at 04:25 PM UTC



12% Overall Similarity

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

Match Groups

120 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks

14 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

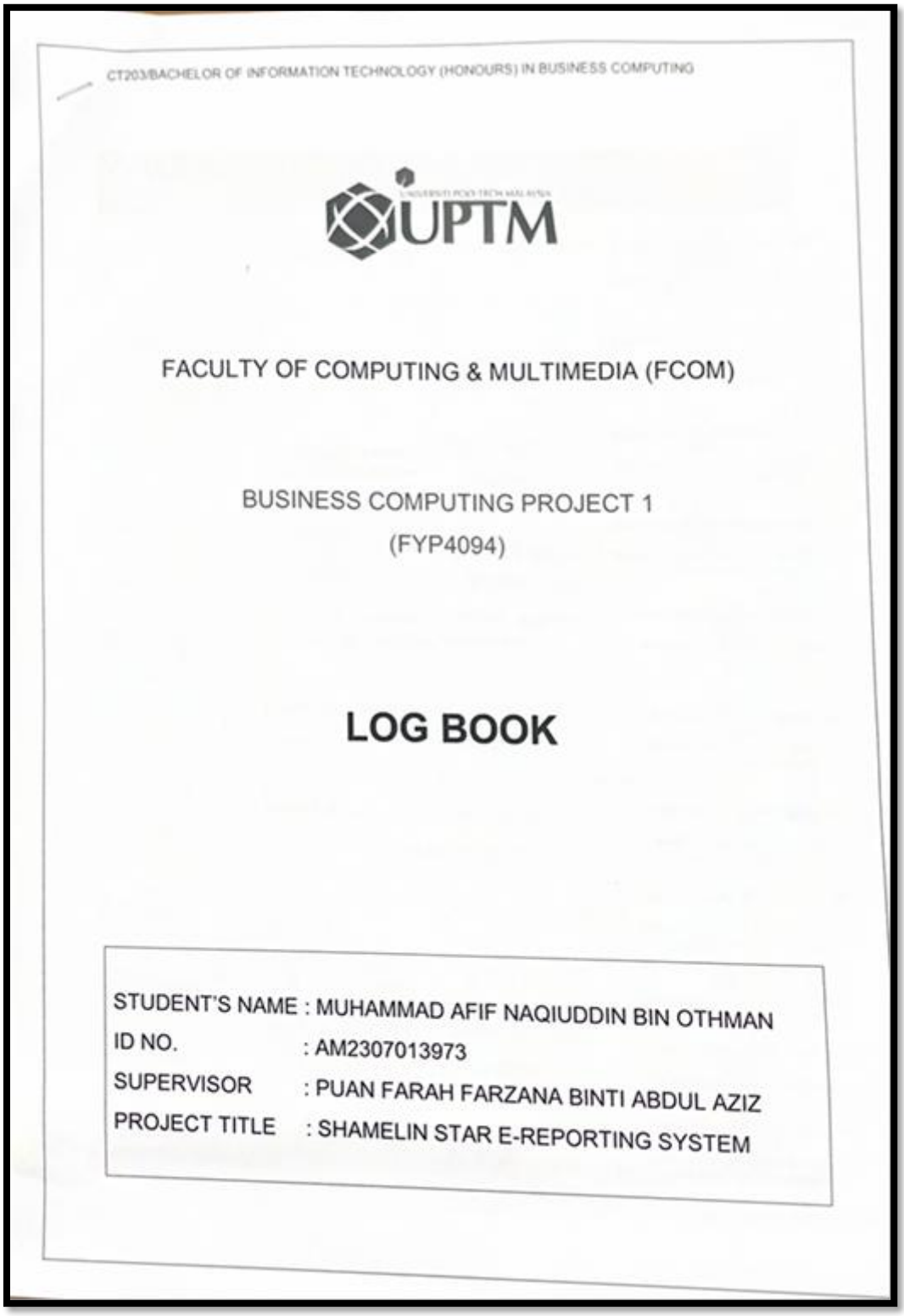
0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
1% Publications
11% Submitted works (Student Papers)

Appendix D – Logbook

- **Log Book FYP 1**



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

UPTM

FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

BUSINESS COMPUTING PROJECT 1

(FYP4094)

LOG BOOK

STUDENT'S NAME : MUHAMMAD AFIF NAQIUDDIN BIN OTHMAN
ID NO. : AM2307013973
SUPERVISOR : PUAN FARAH FARZANA BINTI ABDUL AZIZ
PROJECT TITLE : SHAMELIN STAR E-REPORTING SYSTEM

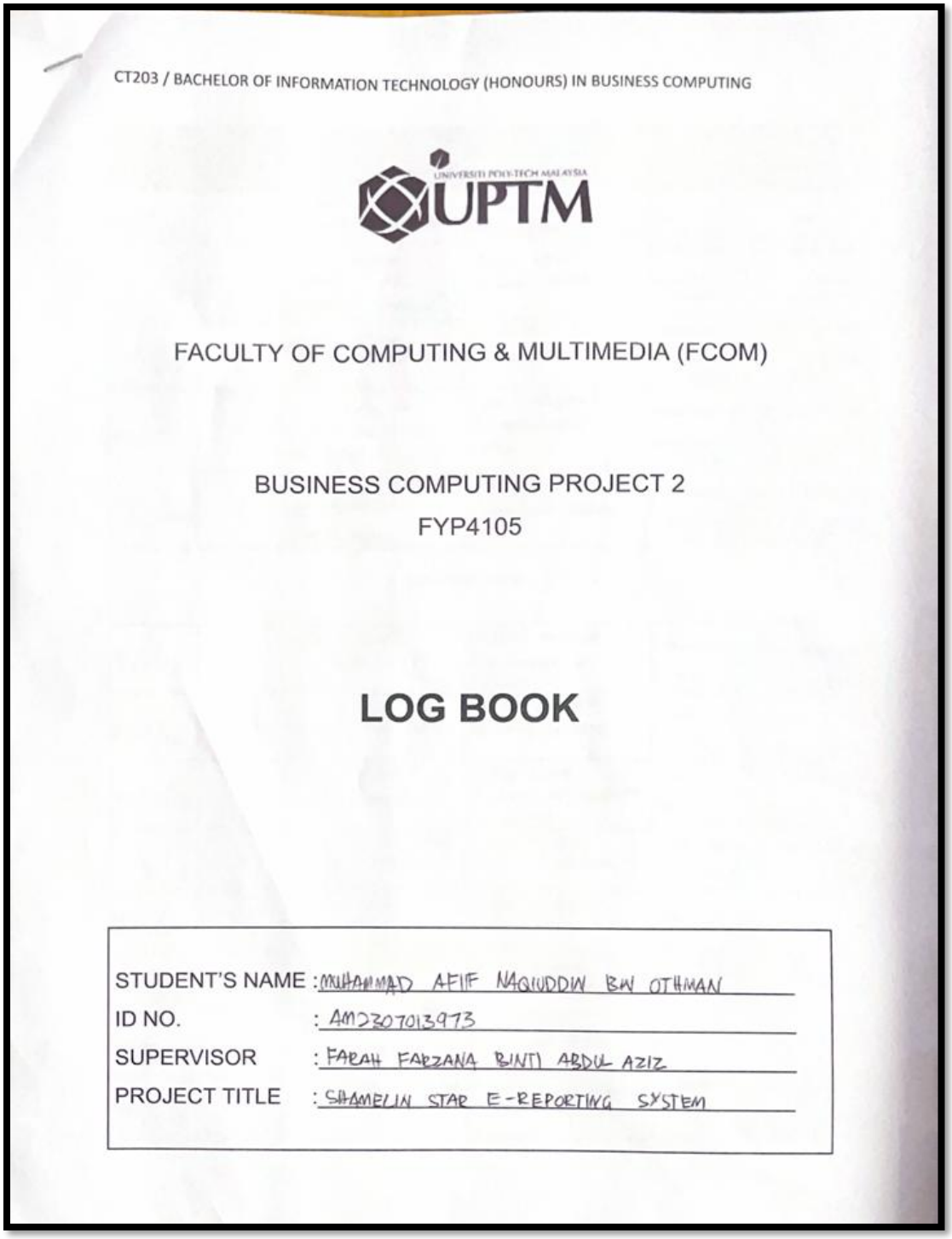
CT203/BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) IN BUSINESS COMPUTING

Date/ Week		Agenda	Next Agenda	Signature (Supervisor)
7/8/2025	1	Find potential Supervisors	Finalize topic and title with SV	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
4/8/2025	2	Discuss the topic and title with SV	Finalize problem statements and objectives	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
21/8/2025	3	Discuss the problem statements and objectives	Begin writing chapter 1	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
25/8/2025	4	Completed draft Chapter 1	Start Chapter 2 literature review	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
1/9/2025	5	Conducted literature review and summaries	continue system comparison	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
8/9/2025	6	completed comparison table	Begin chapter 3 methodology	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
15/9/2025	7	Waterfall method, diagrams and explain	Start Chapter 4 Requirements	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
22/9/2025	8	Conduct questionnaire to resident	write overall chapter 4	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
29/9/2025	9	complete Functional and non-functional	Begin chapter 5 analysis	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
6/10/2025	10	write analysis for questionnaire and interview	create flowchart and BPMN	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
13/10/2025	11	Finalized chapter 5	Supervisor review 1-5	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

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

20/10/2023	12	Repair any mistake in each chapter	Star preparing slide presentation	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
27/10/2023	13	Finish presentation slide	Final consultation with supervisor	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
3/11/2023	14	Presentation	Finish all FYP I and submit full Report	FARAH FARZANA BINTI ABDUL AZIZ Penyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

- **Log Book FYP 2**



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

UNIVERSITI POLYTECHNIC MALAYSIA
UPTM

FACULTY OF COMPUTING & MULTIMEDIA (FCOM)

BUSINESS COMPUTING PROJECT 2

FYP4105

LOG BOOK

STUDENT'S NAME : MUHAMMAD AFIF NAQIUDDIN BIN OTHMAN
ID NO. : AM2307013973
SUPERVISOR : FARAH FARZANA BINTI ABDUL AZIZ
PROJECT TITLE : SHAMELIN STAR E-REPORTING SYSTEM

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

Week		Agenda	Next Agenda	Signature (Supervisor / Coordinator)
4/1/26	1	Brainstorming, Introduction to the course and Title selection	Finalize system architecture and tools	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
11/1/26	2	Finalize system architecture and technologies (Firebase)	Design system database structure	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
18/1/26	3	Design database structure using Firebase & Firestore	Design user interface for mobile and web	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
25/1/26	4	Design UI/UX for resident mobile app and admin dashboard	Start development of authentication module	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
1/2/26	5	Develop user authentication login registration using Firebase	Develop complaint submission module	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
MID-TERM BREAK				
8/2/26	6	Develop complaint submission for resident	Implement complaint data storage in Firestore	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
15/2/26	7	Integrate Firestore database for storing data	Develop complait tracking feature	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
22/2/26	8	Develop complaint tracking feature	Develop admin dashboard for complaint management	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
1/3/26	9	Develop web dashboard to view and manage complaints	Implement staff assignment feature	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
8/3/26	10	Implement staff assignments and complaint management	Develop complaint status update feature	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
15/3/26	11	Develop complaint status update and sync with web and mobile	Conduct system integration testing	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
22/3/26	12	Perform integration testing and fix bug	Conduct system testing and improve performance	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia
29/3/26	13	Perform system testing and UAT	Prepare documentation and final report	FARAH FARZANA BINTI ABDUL AZIZ Pensyarah Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

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

5/4/2026	14	Finalize report	Final submission	FARAH FARZANA BINTI ABDUL AZIZ Penyelaras Fakulti Pengkomputeran & Multimedia Universiti Poly-Tech Malaysia

References

Altexsoft. 2023. Functional vs. non-functional requirements: Examples and types. Retrieved from https://www.altexsoft.com

Android Developers. 2023. Android Studio User Guide. Retrieved from https://developer.android.com/studio

ASUS ROG. n.d. ROG Strix G15 Series Specifications. Retrieved from https://rog.asus.com/laptops/rog-strix/rog-strix-g15-series/spec/

Balaji, S. & Murugaiyan, M. S. 2012. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management. 2(1): 26–30.

Beaton, C. 2024. What are non-functional requirements? Definition and examples. TechTarget. Retrieved from https://www.techtarget.com

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. & Thomas, D. 2001. Manifesto for Agile Software Development. Agile Alliance.

Booch, G., Rumbaugh, J. & Jacobson, I. 2005. The Unified Modeling Language User Guide. 2nd edition. Addison-Wesley.

Brace, I. 2018. Questionnaire Design: How to Plan, Structure and Write Survey Material for Effective Market Research. 4th edition. Kogan Page Publishers.

Dennis, A., Wixom, B. H. & Roth, R. M. 2015. Systems Analysis and Design. 6th edition. Hoboken: John Wiley & Sons.

Duckett, J. 2014. Web Design with HTML, CSS, and JavaScript. Indianapolis: Wiley.

Flanagan, D. 2020. JavaScript: The Definitive Guide. 7th edition. O'Reilly Media.

Google Cloud. 2023. Firebase Documentation. Google LLC.

GSMArena. n.d. Realme 6 Specifications. Retrieved from https://www.gsmarena.com/realme_6-10113.php

Harrington, J. L. 2016. Relational Database Design and Implementation. 4th edition. Burlington: Morgan Kaufmann.

Hassan, R., Abdullah, S. & Yusof, N. 2021. Digital transformation in property management: Improving communication and efficiency through mobile technology. Journal of Facilities Management. 19(4): 453–470.

Ismail, M. N. & Ahmad, R. 2022. Facility management systems and user satisfaction in Malaysian condominiums. International Journal of Building Pathology and Adaptation. 40(6): 812–830.

Laudon, K. C. & Laudon, J. P. 2020. Management Information Systems: Managing the Digital Firm. 16th edition. Harlow: Pearson.

Lim, Y. C. & Tan, P. W. 2020. Smart condominium systems: The role of mobile apps in improving communication and operations. International Journal of Smart Home. 14(2): 87–101.

MDN Web Docs. 2023. CSS3 Reference. Mozilla Foundation.

Moniruzzaman, A. B. M. & Hossain, S. A. 2013. NoSQL database: New era of databases for big data analytics – Classification, characteristics and comparison. International Journal of Database Theory and Application. 6(4): 1–14.

Moroney, L. 2020. The Definitive Guide to Firebase. Berkeley: Apress.

Myers, G. J., Sandler, C. & Badgett, T. 2011. The Art of Software Testing. 3rd edition. Hoboken: John Wiley & Sons.

Myers, M. D. & Newman, M. 2007. The qualitative interview in IS research: Examining the craft. Information and Organization. 17(1): 2–26.

Nguyen, T. H. & Lee, J. H. 2020. Enhancing residential facility services using digital reporting systems. Facilities. 38(3/4): 145–160.

Nielsen, J. 1994. Usability Engineering. San Francisco: Morgan Kaufmann.

Oracle. 2023. The Java Platform Overview. Oracle Corporation.

Phillips, B., Hardy, C. & Stewart, K. 2019. Android Programming: The Big Nerd Ranch Guide. 4th edition. Atlanta: Big Nerd Ranch.

Pressman, R. S. & Maxim, B. R. 2020. Software Engineering: A Practitioner's Approach. 9th edition. New York: McGraw-Hill Education.

Rahman, A. F., Othman, N. & Lim, S. C. 2021. Data-driven approaches for maintenance management in residential properties. Property Management. 39(2): 221–237.

Rouse, M. 2019. System requirements definition. TechTarget. Retrieved from https://www.techtarget.com

Sadalage, P. J. & Fowler, M. 2012. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Boston: Addison-Wesley.

Sommerville, I. 2016. Software Engineering. 10th edition. Boston: Pearson Education.

Udesh, S. S. 2021. Waterfall Methodology, Prototyping, and Agile Development. Retrieved from https://www.researchgate.net

Visual Paradigm. 2019. Visual Paradigm UML tools. Retrieved from https://www.visual-paradigm.com

W3C. 2022. HTML5 Specification. World Wide Web Consortium.

Zhang, Q. & Chen, L. 2023. AI chatbots for customer service automation: Design and adoption in smart living environments. Journal of Intelligent Information Systems. 61(1): 55–78.

Links

1. Project Source Code (GitHub Link)

https://github.com/naqiu1203/FYP-Shamelin-Star-E-Reporting-System

2. Demonstration Video (YouTube Link)

- Web App Demo - https://youtu.be/DnZSRZc8yfA
- Mobile App Demo - https://youtu.be/qllipalX-rg