RESIDENT AND VISITOR MANAGEMENT APPLICATION

LOW EE LYNE

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

September 2022

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : LOW EE LYNE

ID No. : 991024-08-6014

Date : 8 September 2022

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"RESIDENT AND VISITOR MANAGEMENT APPLICATION"** was prepared by **LOW EE LYNE** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature	:	
Supervisor	:	MS. BEH HOOI CHING
Date	:	8 September 2022

Signature	:	
Co-Supervisor	:	
Date	:	

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my supervisor, Ms Beh Hooi Ching for her invaluable advice, guidance and her enormous patience throughout the development of the project. Besides, I would like to say thank you to my moderator, Ms Gunavathi a/p Duraisamy, for providing advice to improve my project.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement during the entire development of this project. Moreover, I would like to show gratitude to all respondents who had helped me to fill up the survey questionnaire and the testers of user acceptance test from targeted user groups that helped me to test the final web application and mobile application. Without their support, the requirement would not be completed, and the project may not be know as accepted by the majority of the users.

Lastly, I would like to say thank you to Univeristi Tunku Abdul Rahman (UTAR) for giving me this opportunity to complete the final year project. I have gained a lot of priceless experience and knowledge throughout the entire development process.

ABSTRACT

Many conflicts arise in neighbourhoods due to improper communication. Effectively managing neighbourhoods is important to management teams to foster communication and enhance connections amongst neighbours. However, in Malaysia, there are still a lot of management teams using traditional paper-based methods to collect and organise their administrative work and documents. Besides, the paper-based method to record visitors by security guards causes traffic congestion. Moreover, verbal communications types of complaints or any important updates pasted by management teams on notice boards or messaging applications were frequently overlooked by the management teams or residents. Therefore, a Resident and Visitor Management System is proposed to overcome these issues. The project seeks to create a web application to simplify administrative work and develop a cross platform mobile application to facilitate communication between administrators and residents. The targeted users of this system are management teams of residential area / condominium / service apartments, residents, visitors and security guards. To accomplish the determined objective, a literature review of the related system was carried out to investigate the identical characteristics that are essential and are not addressed in any other existing applications. As this system is being developed, the phased development approach has been chosen in line with the modules that are emphasized in the project's scope. Planning, analysis, and design were the first steps in the approach. Setting up a server database, creating a web application, and creating a mobile application were the three stages of the system implementation phase. While the mobile application was created with Expo and Supabase, the web application was created with Next.js and Supabase. After the system was created, various system tests were carried out to verify if the requirements are met. The overall results of the user acceptance tests were positive and most of the users rated the satisfactory level as more than 4 out of 5. Hence, the resident and visitor management system is willing to accept and adopt by the majority of target users. The project's objectives have, in a nutshell, been successfully met. Both the web application and the mobile application have been created in accordance with the specifications and scope. A few areas

of limitation are noted for future improvement, and the project ends with several suggestions for further development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

KEGA	Key Economic Growth Activities
SDG	Sustainable Development Goals
RAD	Rapid Application Development
SDLC	Software Development Life Cycle
SQL	Structured Query Language
CRUD	Create, Read, Update, Delete
PWA	Progressive Web Application
UAT	User Acceptance Testing
WBS	Work Breakdown Structure
JSX	JavaScript XML
SSR	Server Side Rendering
RLS	Row Level Security
ORM	Object-Relational Mapping
UI	User Interface
API	Application Programming Interface
REST	Representational State Transfer

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Housing is a subject that cannot be avoided by everyone in his or her life. There are different types of houses such as flats, condominiums, terrace houses, townhouses, etc. can be found all around the world, including Malaysia. In some of the condominiums and landed houses, there are management teams or residential management companies or communities manage and maintain the facilities and safety of the residents. However, conflicts are inevitable where people are present. A neighbourhood management team can play a significant role in resolving disputes amongst neighbours in this situation. Besides, a good management and community team can facilitate communication, cooperation, and improve relationships among neighbours. A resident and visitor management system can help with the effective collection and integration of information for the management team.

Following Key Economic Growth Activities (KEGA) 4: Content Industries (Animation, Programming, Entertainment, Culture, and Digitalisation) of Shared Prosperity Vision 2030 proposed by Government of Malaysia (2019), programming industry which contributes to the development of information system or management system or application that is essential for helping organisations deal with changes in global economies and the business enterprise. Besides, KEGA 12: Green Economy (Government of Malaysia, 2019) motivates the development of software applications that benefit the public and help transform traditional paper-based systems to online digital systems. Thus, it also puts the eleventh Sustainable Development Goals (SDG): Sustainable Cities and Communities in practice.

This chapter includes an introduction, problem background, problem statement, project objectives, proposed solution, and proposed approach. The final deliverable in this project are a web-based application that allows management to effectively manage residents, make announcements, and view visitor records, and a mobile application that allows residents to view announcements, register visitors, and security guards to submit requests to management.

1.2 Problem Background

Communication plays an essential role in everyone's life. Effective communication can avoid unnecessary arguments and facilitate many conflicts. However, with the advanced development of technology, many problems still arise in the neighbourhood due to improper communication. For example, the management team makes announcements by publishing a notice on the bulletin board, which most people would not even look at. Due to the pandemic, many of the management teams transform and start to communicate with the residents through messaging applications in order to broadcast the information to the residents in a short time. Although messaging applications provide easier and more convenient ways for management teams, residents, and security guards to communicate with each other, however, there are still some hindrances. Sometimes, management and residents will overlook important messages or announcements as all the messages are gathered, causing the management team to be unable to handle their claims. This can also make it difficult to effectively deliver some vital information to residents.

It is particularly important for management to collect information on residents to help them understand the residents' needs. Resident information, visitors' logs, and residents' needs can help management to manage a community or residential area more effectively. However, there are still many groups that use paper-based collection methods. As noted by Weber (2005) on the study of paper-based versus web-based data collection and management, a paper-based system took total of 173 minutes and a computer-assisted system used total of 44 minutes to collect and process all the data. The paper-based

system is almost 4 times slower than a computer-assisted system. Furthermore, there are many errors found in the paper-based system. The paper-based system also cannot provide real-time updates, and they need to wait longer to obtain an outcome.

In many guarded areas, the security guards still use paper-based methods to record each visitor when checking in. This costs a lot of time because the security guards manually record each person's information, which often results in traffic congestion in front of the entry of a guarded area or condominium. For example, traffic jams are always spotted in front of the Evergreen Cypress Condominium, Flora Green Condominium near the UTAR SG Long campus, and so forth. This is not only causing inconvenience for visitors but also for residents who often encounter traffic jams when entering and leaving the resident areas. This greatly affects the accessibility of life for everyone.

1.3 Problem Statements

1.3.1 Improper Communication Channels

Communication is crucial to maintain harmony in a neighbourhood. Management team must serve as mediator in order to coordinate problems that arise between residents. Effective communication speeds up everything and improves all parties' satisfaction (Eaglesflight, 2022), so frequent communication is a way for management to be more aware of the needs of the residents. However, there are still many management teams adopt that the face-to-face approach to communicate and this kind of verbal communication is often not recorded. As a result, most of the residents' complaints are not recorded, leading to misunderstandings between management and residents. Although with the development of technology and the cause of the COVID19 pandemic, many of them have shifted to communicate online, such as using messaging applications like WhatsApp, Telegram, WeChat, and etc. However, important information is frequently overlooked or neglected by management teams or

residents. In the questionnaires distributed to residents, WhatsApp, face-to-face and not communicating at all are the top 3 selections answered by 30 respondents. The same goes for management teams, WhatsApp and face-to-face approaches are the most popular approach to communication. From the results of questionnaires collected from residents and management teams, barriers in current communication methods and ways to find out the latest news about the community are also collected from both sides. 20 out of 30 residents responded that they learn the latest updates and news of the community through messaging applications like WhatsApp, Telegram, and WeChat. However, 13 out of 30 residents and 3 out of 16 respondents faced overlooked important information on current communication methods are learnt. Therefore, it is essential to put in place a platform that allows smooth communication between management and residents.

1.3.2 Outdated Data Collection System

Resident and visitor information serves as the critical part for management to understand and analyse the needs of resident and visitor. From the survey conducted in this project, 53% of respondents are using the traditional paper-based method, 31.3% of respondents are using management applications, 25% are using google forms and the remaining respondents also use observations to collect residents' information. However, the paper-based data collection technique makes it difficult for management to manage data in a centralised way. It can take a long time to go through records one by one when information is needed. Although there are many spreadsheets have gradually emerged like Google Sheet, Excel, to store and record data which digitalize the managing process. However, it allows everyone in the team to add information concurrently, and it is difficult to distinguish who modified which cell. As a result, with so many revisions and computations going on at once, human error might easily occur and affect the accuracy of the data obtained. Furthermore, a spreadsheet represents data in a two-dimensional grid of data, which is not user-friendly and makes it easy for many users to miss errors unless they go through it row by row. Therefore, a system that allows us to collect data and analyse the

data in an easier, convenient, and effortless way will always be an intelligent alternative for the management.

1.3.3 Slow Visitor Registration Process

When a visitor arrives at the entrance of the guarded area, the security guard will stop the visitor and ask for their identification card or licence for registration. Then, the security guard will record the visitors' IC number, name, phone number, and the unit they plan to visit on the paper-based visitor logbook. There are some locations with stricter security, where they will call the house owners or residents to confirm the visitor's arrival before allowing them access. This is a very typical way in Malaysia for visitors to check in. It takes a lot of unnecessary time to go back and forth and communicate. In many cases, the registering or check-in process takes too much time and causes traffic congestion. For example, traffic jams occur in front of the entry of the Evergreen Cypress Condominium near UTAR Sungai Long Campus. From the result of the survey conducted in this project that collected from management teams, 56.3% of respondents are using paper-based logbook, 31.3% are using visitor management application, and 12.5% does not record any visitor information found. 7 out of 16 respondents from the management side faced traffic congestion. Thus, the module to speed up the registering visitor process is very necessary to be developed in the mobile application to reduce the traffic congestion caused by this process.

In conclusion, a mobile application should be developed to resolve the first and third problem statement, and a web application should be developed to resolve the second and third problem statement.

1.4 Project Objectives

1. To create a web application to simplify administrative work in management team

2. To develop a mobile application to facilitate the communication between the management teams and residents

3. To evaluate the mobile application and web application by conducting user acceptance test

1.5 Proposed Solution

The proposed solution for this project is to develop a web application and cross-platform mobile application. The web application is for the management team to manage and monitor the residents and visitor information, whereas the mobile application is mainly for residents, visitors, and security to submit their requests to the management team. Both web application and mobile application in this project are developed with both front-end and back-end. The web application shares the same server and database with the mobile application to communicate with each other and provide information to both sides' users.

For the technical aspect, Next JS which based on React.Js is used to create the front-end view of the web application meanwhile, React Navigation and Expo that based on React Native are used to create the front-end view of the native mobile application. Supabase is selected to perform as the back-end and database of the application. Figure 1.1 illustrates the overview of the proposed solution.

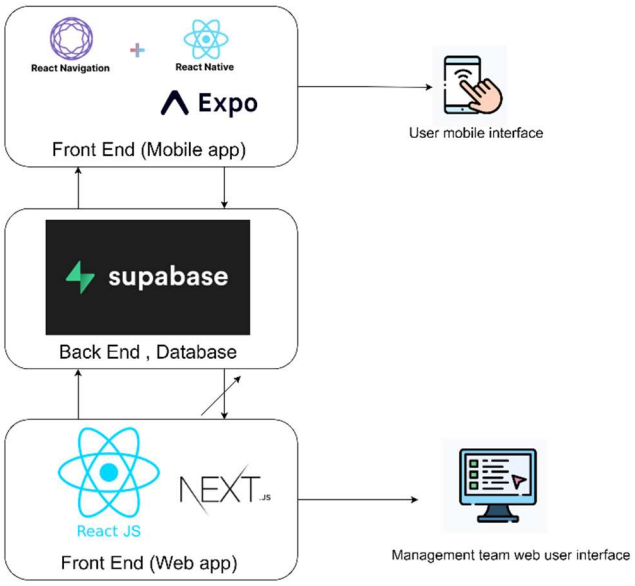


Figure 1.1 Overview of the Proposed solution

The web application allows the management team to use different devices like laptops, desktops, tablets to operate wherever they are as long as they have internet access. This can significantly enhance the productivity of the management team as they no longer need to go back to the office to deal with resident applications or complaints.

The mobile application is developed as cross-platform because it enables maximum coverage of the mobile phone users. Therefore, everyone in the neighbourhood can use it. According to Wurmser (2020), cell phone users will spend around 4 hours per day online in 2020, with applications taking up for 88 percent of that time. Since the coverage of a mobile application is greater than that of a web application, it was chosen to serve as a resident, visitor, and security guards view. This brings convenience to the user and will greatly increase the probability of use.

1.6 Project Approach

1.6.1 Research Approach

Among 3 common research approaches: qualitative, quantitative, and mixed approaches, the quantitative approach is selected as the research approach for this project. Quantitative approach is defined as a systematic examination of issues based on the gathering of quantitative information and the use of statistical, mathematical, or computational approaches (Quantitative Research: Definition, Methods, Types and Examples, 2022). As noted by Sukamolson (2007), Quantitative research comes in a variety of forms. Some examples are survey research, correlational research, experimental research and causal-comparative research. He also advised that survey research uses scientific sampling and questionnaire design to examine demographic variables with statistical precision. Among various types of quantitative research, an online questionnaire is used to collect opinions from respondents. In the research, online questionnaires are randomly distributed to respondents to collect their expected functionality and features of the resident and visitor management application. The results of the questionnaires are analysed to assist in requirement gathering.

1.6.2 Development Approach

The development approach for this project is the phased development methodology. It is a type of Rapid Application Development (RAD) method. It divides a large system into several versions that are created in order (Dennis, Haley, Tegarden, 2015). If the requirement changes, it is easier to rebuild and redevelop the application by breaking it down into smaller parts. This method allows for amendments to be made at any point in response to complaints or needs highlighted by users. Figure 1.2 shows the phased development methodology.

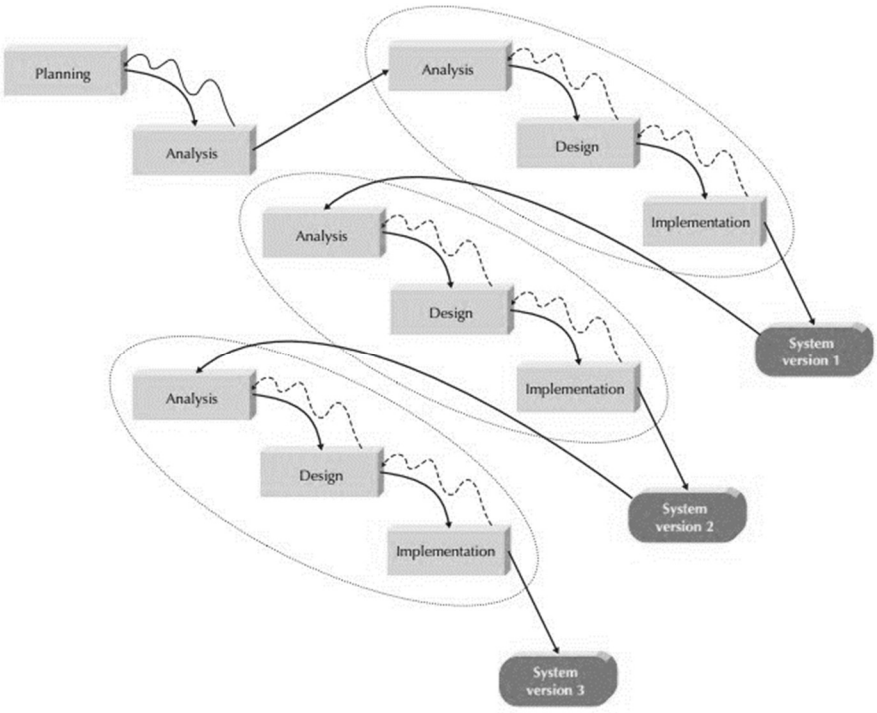


Figure 1.2 Phased Developmen-Baseed Methodology (Dennis, et al., 2015)

According to Dennis, Haley, Tegarden (2015), there are 7 important criteria which include unclear user requirements, complexity, unfamiliar technology, reliability, schedule visibility, and short time schedule to select the appropriate methodology. By analysing each criterion, Phased development process is adopted. Prioritising core features in a phased development method guarantees that the first version of the software is beneficial to users and allows them to suggest additional needs for later versions. The major benefit of this approach over other software development life cycle (SDLC) approaches is that it allows software engineers to test the subsystem and find problems or defects early on. The final deliverables are a web application and mobile application that are interconnected to each other.

1.7 Scope of Project

1.7.1 Target Users

The targeted users for this project can be categorised into 4 groups which include neighbourhood management team of a guarded neighbourhood, residents, visitors, and security guards. Neighbourhood management team is the only user of the web application while residents, visitors, and security guards are the users of the native mobile application in this project. This project focuses 50 % on management teams, 30% on residents, 10% on visitors, and 10% on security guards.

1.7.1.1 Neighbourhood Management Team of Guarded neighbourhood

The neighbourhood management team includes professional resident management companies or community management teams. This project is targeted at those who require a management application to facilitate the data collection and management process in order to manage the neighbourhood effectively. The modules that involve the management team are the registration request and approval module, user profile module, visitors' records module, announcements' broadcasting module, and feedback collection module.

1.7.1.2 Residents

The targeted residents are those users who live in the neighbourhood that the neighbourhood management team is using this resident and visitor management system. Residents require to register themselves and get approval from the neighbourhood team to use the application. The modules involve the user profile module, registration module, visitor's registration module, announcement module, feedback collection module.

1.7.1.3 Visitors

The targeted residents are those users who would like to visit their family or friends or known ones in the neighbourhood. The visitors requires to register by the residents they visit. The module that involves is the visitor's verification module.

1.7.1.4 Security guards

The targeted security guards are those who guard in the neighbourhood. They serve as an authenticator to verify the check-in of every pre-registered visitor and register ad-hoc visitors. Therefore, the modules involved are the visitor's verification module and visitor's registration module.

1.7.2 System Scope

The project covers both web-based applications and cross-platform mobile-based applications. Both applications provides in English only. Axure RP9 is used to create the prototype of the application. Visual Studio and Android Studio are used to develop both applications.

1.7.3 Application Features / Modules

1.7.3.1 Web Application

1. **Registration Request and Approval Module**

   In this module, the management team receives the registration request from the residents and make the decision of either approving or rejecting the application of residents. Only the approved residents are allowed to use the application. The management team is also allowed to view the residents' information.

2. **User Profile Module**

   This module allows the management team with higher authority to add an administrator (member of the management team) to the application. They need to add the new administrator email into the application and the application auto-generates a password for the new administrator to login. Once the profile is created successfully, new administrator can login their account and change new password.

3. **Visitors' Records Module**

   This module allows the management team to view and track the visitors' records.

4. **Announcements Broadcasting Module**

   This module allows the management team to broadcast the announcement to all the residents at once time. They are also allowed to create and delete the announcement.

5. **Feedback Collection Module**

   This module allows the management team to receive the feedback from residents and categorise them into different categories which allow the team to manage easier.

1.7.3.2 Cross-Platform Mobile Application

1. **Registration Form Submission and Login**

   This module allows residents to submit their registration form to the management team. After the management team approves the application, residents are allowed to log in their account to use different function in the mobile application.

2. **User Profile Module**

   This module allows residents to modify their user profile.

3. **Visitor's Registration Module**

   This module allows residents to register visitors who like to visit them and security guards to register ad-hoc visitors. Residents need to provide the visitors information including name, ic, phone number, car plate number.

4. **Visitor's Verification Module**

   In this module, both security guards and visitors are involved. It allows the visitor to check in their visitation registered by the resident and the security guards need to verify the check in.

5. **Announcement Module**

   This module allows residents to receive the announcement broadcasted by the management team.

6. **Residents' Feedback Submission Module**

   This module allows the residents to submit their feedback regarding the neighbourhood to the management team.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

A management application allows to bring a lot of convenience to the management team to improve their work efficiency and solve the problem faced. In order to understand more about how to develop and design the application, a literature review of similar systems and relevant topics was conducted. In this chapter, researches are done on how to design the development framework, database of web applications and mobile applications. Besides, an evaluation of web applications and mobile applications should also be conducted in order to validate both applications that meet the determined user requirements.

2.2 Similar Systems Review

There are lots of different similar applications found in the market but not all of them contain the main features that this project would like to implement. Several existing similar systems are found and the features provided by those systems are studied in order to find out the major features that should be included in the Residents and Visitors Management System. Those systems being researched are Eden Community App, kipleLive, M4U Home System and eCommunity. All features are found through their official websites, and screenshots and reviews are found from application stores including App Store and Google Play. Table 2.1 shows all the features for different users provided by similar systems. Screenshots and user reviews of similar systems are attached in Appendix A for reference.

Table 2.1 Features Provided by Similar Systems

	Features for Management Team	Features for Residents	Feature for Visitors
Eden Community App	- Generate invoices - Manage payments - Make announcements - Post notices - Manage facility bookings - Communicate with residents	- Pay bills - Book facility - Receive notices and announcements - Post information in community wall	- Not applicable
kipleLive	- Manage admin - Manage residents - Manage bookings - Manage communication (e-message & e-news) - Manage bills - Manage forms	- Generate visitors invitation - Announcement blasting - Book facility - Submit forms - Report issues - Pay bills - Emergency button - In-App Community Phone Directory - In-App Home Guide (Community Library)	- Manage visitors - Manage guards - Generate guard attendance report - Generate guard incident reporting

M4U Home System	- Receive feedback on time from the residents - Manage residents' unit profile - Generate invoices and send to all residents - Pass live message to all residents - Record office staff details - View committee members' profile - Manage parking lots - Check status of facility booking - Manage and minitor visitor record - Receive notification alert from residents	- Receive live notifications - View security guard's profile - View committee members - Show emergency & vendors contacts - Show facilities availability - Book facilities and apply for renovations - View invoices and payment status - Send feedback or complaints to management office - Pre-register visitors - Show service providers list	- Monitor visitors - Intercom (communication between security guards and residents) - Collect residents' parcel - View log (visitor's history, parcel collection, SOS alert) - SOS alert button - Show emergency & vendors contacts - Verify visitors pre-registration record through QR code - Record visitor parking availability
eCommunity	- Manage payments - Real-time monitoring of	- Pay bills - Book facilities - Pre-register visitor	- Manage visitors entry

	income and expenses - Send out latest announcements with push notifications - Permit Application and Feedback Portal - Handle facilities bookings - Store important documents - Manage visitors' records - Create survey forms and voting forms	- Receive instants updates on announcements - Manage feedbacks	

From the features listed in Table 2.1, Eden Community App does not provide specific features for security guards while the other three do. M4U home system offers the most features. The main features that are found in all of the similar systems are announcement management, feedback management, visitors management, bill payment management, and facility management.

2.3 Development Framework of Web Application and Mobile Application

In this section, development frameworks have been researched. There are many methods to develop web and mobile applications. However, there is no one solution to solve everything, therefore, it is important to study what is the most suitable development framework for this project. For web applications, ReactJS has been studied and for mobile applications, both Progressive Web Application (PWA) and cross-platform mobile applications have been studied.

The literature on Comparison of Technologies for Multiplatform Mobile Application Development by Crha (2021) has highlighted the performance of different measurable metrics on cross-platform mobile applications (React Native and Flutter) and PWA. In the study, an experiment was conducted to compare React Native, Flutter and PWA performance on Android and iOS environments with Firebase as the cloud backend. The experiment focuses on objectively measurable metrics which include startup times, screen render times, CPU usage, memory usage, frame per second, code sharing, codebase size, and build times. Table 2.2 lists the summary of the measured metrics compared in the study.

Table 2.2 Summary of Measured Metrics Compared in Crha's Study

Metrics	**Cross-Platform Mobile Application (React Native / Flutter)**	**PWA**
Application Startup Time	Faster startup times	Slower startup times
Screen Render Times	Faster screen render times	Slower screen render times
CPU Usage	Similar performance as PWA (perform better than PWA in some screen)	Similar performance as Cross-Platform Mobile Application

		(perform better than Cross-Platform Mobile Application in some screens)
Memory Usage	Higher memory consumption (React Native consume lesser memory than Flutter)	Lower memory consumption
Frames per Second	On Android, React Native showed better result in Route Details Screen, while Flutter showed better result in Map Screen. Other than this two screens, both of them has similar result. On iOS, no significant different between React Native and Flutter	Not applicable
Application Size	On Android: Flutter smaller than React Native. On iOS: React Native smaller than Flutter	On Android: Smallest Size compared to Cross-platform Mobile Application On iOS: Not determined
Code Sharing	Allow to share close to 100% of the code between Android and iOS.	Allow to share close to 100% of the code between Android and iOS.
Codebase Size	React Native's smaller than Flutter's.	Not applicable
Build Time	On Android: Flutter faster than React Native	On Android: Flutter faster than PWA (when it came to clean builds)

	On iOS: Flutter faster than React Native	On iOS: PWA faster than Flutter and React Native
User Experience	Better user experience	Worst user experience

According to the findings listed above, there was no clear winner of the comparison as a whole, and each technology performed well in some aspects over the other. However, a cross-platform mobile application is more suitable to be developed in this project as it performs better in those metrics that are considered more important in the Resident and Visitor Management Mobile Application. For example, the mobile application of this project involves 3 types of users and each type of user will have different user interfaces, so it is important to provide better user experiences to every user.

In addition, a series of analysis on the study to discuss industry practitioners' opinions and thoughts on cross-platform mobile development tools was carried out in An Empirical Study of Cross-Platform Mobile Development in Industry (Biorn-Hansen, 2019). It contributes to a deeper knowledge of the difficulties behind cross-platform mobile development by combining quantitative industry viewpoints. In the study, an online survey questionnaire is conducted and a total of 101 respondents were gathered from different online communities and forums relevant to mobile applications were involved in the questionnaire. According to Biorn-Hansen (2019), concerns about user experience, performance, and the maturity of the technical infrastructure was raised in the study. Based on the response from the survey participants, React Native, PhoneGap and the Ionic Framework are the most interested in terms of cross-platform framework popularity and usage as compared to MoSync and Titanium. In terms of frameworks that respondents use professionally, PhoneGap outperforms ReactNative and Ionic Framework. As from the findings from both studies, React Native is suggested for the development of the proposed mobile application in this project.

React Native is a JavaScript framework for creating natively rendered iOS and Android mobile apps. It is maintained by Meta. It is created with JSX, a combination of JavaScript and XML-like markup. React Native also allow applications to use functionality such as the user's location or the phone camera by using platform APIs. It allows for more efficient code sharing between Android, iOS and the Web and faster development of mobile applications without compromising the quality of the app or the end user's experience.

ReactJS is the same as React Native which uses JavaScript and is maintained by Meta, however, it targets to build user interfaces and web applications rather than mobile applications. ReactJS is one of the most widely used JavaScript libraries today for creating massive web applications that can utilise data and modify without reloading the browser. In the study of ReactJS contributed by Rawat and Mahajan (2020), the Virtual Dom feature of React is one of the most significant features, as it prevents page reloading and improves the overall efficiency of their application. ReactJS which is built over JavaScript, provided with Node Package Manager (NPM) offers an easier way of installing external dependencies. ReactJS allows building user interfaces with a component-based architecture approach in which each component is independent and reusable bits of code. The survey conducted by Stack Overflow (2021) also disclosed that React is the most commonly used web framework and the most wanted web framework, desired by 25.12% of the 66202 respondents.

Next.JS is a web-development framework that allows server-side rendering in React-based web applications. It is used by leading companies such as Netflix, Docker, GitHub, Uber, Starbucks, and many more. This framework also offers functionalities like static exporting options, easy production builds, automatic code-splitting, etc. Therefore, Next.JS is also preferred to use as it expands the capabilities of React and streamlines the development process.

2.4 Database used on Web Application and Mobile Application

A study of different databases has been conducted to gain more knowledge and understanding of different databases. As a database management system offers an effective way for management teams to handle large amounts and multiple types of data. One of the most important considerations when selecting a modern database is whether to use a relational (SQL) or non-relational (NoSQL) data structure. PostgreSQL, Oracle, MySQL, and so forth are examples of SQL databases. NoSQL database examples include MongoDB, BigTable, Redis, etc. In recent years, real-time databases are also popular options that developers choose to implement in their systems. A real-time database is a database system that processes data in real-time to manage workloads that are continually changing (Ramamritham, 1993). Firebase, Supabase are some of the examples of real-time databases. MySQL, PostgreSQL, MongoDB, Firebase, etc. are the databases being studied in this section.

SQL (Structured Query Language) is a coding language that is used to manage relational databases and execute various operations on the data stored. NoSQL is a database management approach that can handle a variety of data structures, such as document, key-value, columnar, and graph formats. A comparison study of the difference of SQL databases and NoSQL databases by Sheldon (2021) is reviewed. Table 2.3 has shown the results of the comparisons.

Table 2.3 Comparison of SQL and NoSQL

	SQL databases	NoSQL databases
Data Structure	- Suitable for highly structured data - Built on a relational model that normalises data across rigid tables and standardised relationships between them	- Does not necessitate the use of a normalised configuration or the use of a relational model. - Versatile enough to support many models

Language	- SQL language	- The language used is determined by the type of NoSQL database, the implementation, and the operation.
Schemas	- Requires a specified schema that dictates data is stored and how tables are set up, causing a strict structure. - Low flexibility	- Makes use of a dynamic schema that eliminates the need for a predefined data structure. - High flexibility
Data integrity	- High degree of data integrity	- Low degree of data integrity - Data in a distributed system may be unreliable at times.
Scalability	- Scale vertically	- Scale horizontally
Querying	- Skilled at processing queries and combining data from several tables.	- Inconsistency among products necessitates extra effort when querying data.
Maturity	- Built on mature technologies - Backed up by large developer communities	- Built on less mature technologies as well as supported as SQL products. - Supported by constantly growing communities

Generally, SQL is not quicker than NoSQL just as NoSQL is not quicker than SQL. SQL databases are normalised databases in which divide data into several logical tables to prevent data duplication and redundancy. For searches, joins, updates, and other operations, SQL databases outperform NoSQL databases in this situation. Furthermore, unstructured data, which might be column-oriented, document-oriented, graph-based, etc., is the main purpose for which NoSQL databases are designed. Instead of partitioning, a single data entity is preserved in its whole in this case. As a result, read and write operations on a single data entity are completed more quickly in NoSQL databases than in SQL databases.

In comparison to NoSQL, a relational database management system (RDBMS) that processes mostly structured data could handle structured data more easily. A recent study (Jung, et al., 2015) has proved that the performance comparison of Create, Read, Update, Delete (CRUD) operations in MongoDB which is NoSQL based was faster than PostgreSQL which is SQL based in general. In the study, unstructured data model and relational data model have been created by using PostgreSQL and MongoDB. Both databases were put to the test by running 300000, 210000, 150000, 90000, and 30000 data cases based on card mileage data. Furthermore, using MongoDB in conjunction with an unstructured data architecture will increase overall efficiency. However, an RDBMS such as PostgreSQL will provide better performance when the environment needs a precise and structured data model.

As Lindberg (2018) points out in her research of MySQL and PostgreSQL response times, PostgreSQL offers significantly faster response time when doing A/B testing compared to MySQL. The comparison was carried out by performing the experiment to measure the response time of MySQL and PostgreSQL. A web application built with mainly HTML, JavaScript and PHP was implemented for the research as well as instances of the two separate databases. Once finished the setup, a pilot study was conducted to measure the response times of image processing of different databases. In three of the testing examples, PostgreSQL was approximately four times faster than MySQL, and in the remaining case, it was ten times faster. Through the research, it can be

concluded that PostgreSQL has outperformed MySQL in terms of image processing. However, the study only covers the image processing response time, therefore, other research including other measurable metrics of the databases should be further researched.

Similarly, Ohyver, et al. (2019) has conducted a study to compare the performance of Firebase Realtime database and MySQL database. It analysed each CRUD operation by using the Wilcoxon Signed-Rank test on their tested mobile application. The findings show that Firebase Realtime Database has better response time than MySQL. In their research, they also concluded that key features provided by Firebase Realtime Database are more suitable for their Daily Nutritional Needs Mobile Application as Firebase provides real-time data and streamlines the backend process, allowing developers to focus on developing the mobile app rather than worrying about server-side programming.

A cloud database is a database system that is produced and accessible via the cloud. It brings the advantages of fast deployment, quick accessibility through the provider's API, lower cost of database maintenance, etc (Marijian, 2021). Following by a survey conducted by Shareef, Shareef and Rashid in 2022, has provided an overview of the performance of different cloud databases. The study compared various key features including portability, accessibility, configurability and so forth in different cloud database frameworks. The results of their investigation have concluded that NoSQL databases work better for most jobs and recommended Amazon RDS for almost all organisations, Amazon SimpleDB for small businesses, and Firebase Realtime Database for group or personal use.

A comparison between Supabase and Firebase is conducted by Salim (2022) and the findings are shown in Table 2.4.

Table 2.4 Comparison between Supabase and Firebase

Comparison	Firebase	Supabase
Database	Uses a non-structured database (NoSQL)	Uses a structured database (PostgreSQL).
Compatibility	Data is difficult to export and use in another platform.	Postgres is compatible with a wide range of framework and tools.
Pricing	Consists of the Spark plan and the Blaze plan	10,000 users and 500MB of storage space are included in the free tier plan.

In general, Supabase outperforms Firebase if a structured database is involved. Nevertheless, Supabase which only launched in 2020 and is still considered new in the market, therefore, there is no further detailed research found to compare the performance of Supabase.

These studies provide important insights into the conclusion that Postgresql and Firebase outperform other database management systems. Through the findings, Supabase can be considered to be used for developing this project. According to Kumparak (2021), Supabase is an open-source Firebase replacement, according to Kumparak (2021), that offers real-time and RESTful APIs services to PostgreSQL databases. Supabase works like Firebase however it uses PostgreSQL for the database and listens to real-time updates through a number of tools that they create. According to Olubisi (2022), Supabase also takes care of the scaling and provides easier data migration.

2.5 Evaluation of Web Application and Mobile Application

Evaluation of systems is significant to verify and validate the quality of the software conformance to the business requirement, policy, standards and so forth. One of the most popular evaluation methods is User Acceptance Testing (UAT). UAT is testing done by the client or end-user to ensure that the software system is acceptable before it is delivered to the production environment. It is crucial because it shows that critical business operations are operating in a manner that is suitable for scenarios in the real world. It will be noted and forwarded to the developers for adjustment if the component doesn't perform as intended during testing. This step serves as a final check to ensure that the final product is well-constructed before it releases to the client. In terms of project management, UAT documentation is also critical to serve as a controlled document that requires clients to sign off to prove they are satisfied with the functions developed in the application and the functions are conforming to the business requirements before entering the deployment phase. Therefore, automated user acceptance testing, a feasible framework for the UAT process and a concurrent think-aloud method are discussed in this section to explore the different approaches and use of UAT in evaluating a system.

There are various automated UAT tools for web applications that could be found in the market nowadays. For instance, Test Complete, ACCELQ, HPE Unified Functional Test, and so forth. Sualim, Yassin and Mohammad (2016) have performed a comparative evaluation of automated UAT tools for web-based applications. TestComplete, Selenium Webdriver and Watir Webdriver are selected to be reviewed in their paper. The tools are evaluated based on the usability criteria such as efficiency, effectiveness, satisfaction and error rate. Based on usability criteria for UAT, Watir Webdriver has outperformed other testing tools. It has shown great test script, the shortest execution time, better user experience, the fastest rate of completion of tasks given and the largest maximum error rate during testing of the web application. It also supports that automated UAT tools increase the software usability and robustness by keeping UAT goals at the requirement and task levels.

Next, a feasible PEF (Planning, Execution, Feedback) framework which involves three components is proposed by Al-Hurmuzi, Al-Khanjari, and Al-Kindi in 2018. The three components included are listed below:

1) Illustrate users took part in the UAT.

2) Lists the requirements (retrieve from a database) that users need to test.

3) Make a list of the users' feedback that should be forwarded to the testing team.

The framework transformed the recommended components into an appropriate and user-friendly computerised tool that can be applied when conducting user acceptance tests. It helps to solve the difficulties faced like the tight schedule for UAT at the last stage of software testing lifecycle, the ambiguous roles and responsibilities of users, and unclear requirements when performing the UAT manually.

Think-aloud is the practise of verbalising thoughts while performing an action (Ericsson and Simon, 1993). Implementing concurrent think-aloud method as usability performance evaluation in UAT Process is recommended in the research paper of Ichsani (2018). As noted by Ichsani (2018), literature reviews and comparisons of the researchers' experiences with relevant pieces of literature are conducted. The advantages of implementing a concurrent think-aloud method in UAT process includes allowing developers to immediately understand the end users' experience, and maximise end users' contribution. For disadvantage, the paper also reports that the respondents may not provide any feedback or comments on the evaluated application. In order to capture more user expressions and thoughts, using eye-tracking software and speech-to-text software are suggested to replace the observations with basic web cameras, headsets and questionnaires. Overall, the study highlights concurrent think-aloud method as UAT evaluation assists in gathering additional information from respondents in specific tasks such as creating an account and finding information.

2.6 Summary

In this chapter, the research of development framework, database and evaluation of web applications and mobile applications has been successfully carried out. In view of all that has been mentioned so far, the development and evaluation of web applications and mobile applications are clearer and more identified. The cross-platform framework, React Native is recommended to develop a native mobile application. ReactJS is also suggested to implement when developing a web application for Resident and Visitor Management Application as it works similarly to React Native. The same goes for Next.JS should be used because it extends React's functionality and simplifies the development process. In terms of databases, MySQL, PostgreSQL, MongoDB, Firebase, and other cloud databases are studied. With the out-performance of PostgreSQL and Firebase, Supabase which is called Firebase alternative and uses PostgreSQL as a database is recommended as it has better scalability and offers easier data migration. Lastly, User Acceptance Test is recommended to be the tool to evaluate both applications.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

In this chapter, software development methodology, development tools, and work plan for this project were discussed. The Resident and Visitor Management System was developed using the phased development methodology, which is covered in this chapter. Details of each phase were also explained in this chapter. Moreover, it consist of a list and description of the development tools used in this project. Last but not least, a work breakdown structure and Gantt Chart were included at the end of this chapter.

3.2 Software Development Methodology

The phased development methodology was selected for this project. There were 4 main phases in this methodology which include planning phase, analysis and design phase, development, testing phase, as well as closing phase. Each phase started only when the previous phase is finished. Nevertheless, the development and testing phase performed iteratively until the system is fully developed. In general, the highest priority features were created before the other features. as they require respectively longer time for implementation. The next phase was kicking start once the previous phase was ended with different system versions. After iterative development phases, the final phase which is to close the project as the system has been fully developed. The overview of the phased development methodology is shown in Figure 3.1.

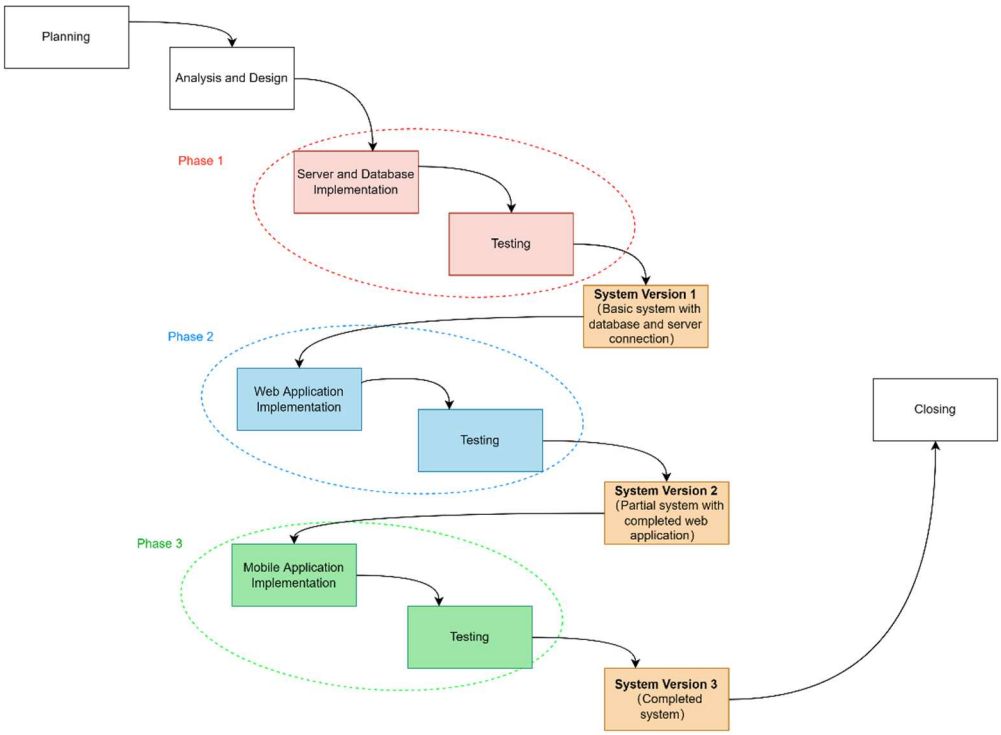


Figure 3.1 Phased Development Methodology of Resident and Visitor Maanagement System

3.2.1 Planning

Planning stage is the initial stage of the phased development methodology. Project proposal, requirement gathering and elicitation, and project schedule are covered in this phase.

3.2.1.1 Project proposal

The project proposal's first task of this project was to identify issues that residential management teams confront, as well as find the objectives for resolving them. The purpose of objectives was to guide the direction of the project throughout its development to assure that it stays on track. The following task was to provide a project solution after the objectives were specified. This activity provided an overview of the system in order to give a broad understanding of the final deliverable. The planning phase then moved on to determining the project methodology. To make sure that the development

process was efficient and effective, the right development methodology should be chosen. Finally, the project scope was outlined to define the project's constraints. For instance, system scope, user scope and module scope were included in the project scope. A completed and submitted project proposal was the final deliverable of this part of the planning phase.

3.2.1.2 Requirement Gathering and Elicitaion

Once the proposal has been approved by the supervisor, the project moved on to the requirement gathering and elicitation phase. Two sets of questionnaires were prepared and distributed to the management teams and residents throughout this stage of the planning phase. The information gathered from the questionnaire responses were used to prepare the user's requirements. Information on functionalities that were essential for the application was gathered by comparing some similar existing applications. After the requirements engineering process, all of the collected data was analysed, and the requirements were finalised.

3.2.1.2.1 Questionnaires

The online survey questionnaires approach was selected as it was able to collect more data in a shorter period of time. Two separated sets of questions were designed for management teams and residents by using Google Form. Both sets of questionnaires consisted of three sections with different focuses. The first section was the demographic section to study some basic information from target users. The questions listed in the second section weree to understand the issues faced by the respondents. The third section focused on the features of the application. The questionnaires were distributed to the target users through social media, email, and so forth. The questionnaires for residents successfully received 30 responses while the questionnaires for the management teams only reached out to 16 respondents.

3.2.1.2.2 Review of existing systems

A total of four similar systems were reviewed and compared. All the features in the four similar systems were listed in Chapter 2 and related screenshots, as well as user reviews, are tabulated in Appendix A for reference. Four of the systems were available in application stores including Google Play and App Store. After the comparison, the main elements and necessary functionalities given by the majority of the systems were integrated in this project. The main features included in this project were:

1. Visitor management
    a. Manage visitor's records
    b. Check-in visitors
    c. Verify visitors check-in
    d. Pre-register visitors
2. Announcement management
    a. Publish announcement
    b. Read announcement
    c. Manage announcement
3. Feedback or complaint management
    a. Submit feedback or complaint forms
    b. Read and reply to feedback forms
4. Administrators and residents management
    a. Manage resident's registration status
    b. Manage administrators profile
    c. Manage residents profile

3.2.1.3 Project scheduling

The final step in the planning phase was scheduling the project. A detailed work breakdown structure (WBS) was established to organise the tasks and provide an overview of essential tasks to be done in each phase. It helped to ensure the features will not be missed out during the development process.

Next, a Gantt chart of every phase in the phased development methodology was created. Each task was listed with its expected start date and expected end date in order to determine the duration of the project. Gantt charts were used to track project progress and ensured that the entire development process runs in lockstep with the typical timeline. Any delay in completing the task may result in lost time or increased project costs. As a result, the project's WBS and Gantt chart were the stringent guidelines that must be followed.

3.2.2 Analysis and Design

The development process entered this phase once the planning phase had ended. In this second phase of phased development methodology which was analysis and design, the project scope analysis was performed to design the system. In order to better comprehend the system structure prior to deployment, use case diagrams were created using the analysis's findings. To demonstrate the comprehensive information of all use cases, use case descriptions were also produced. Use case diagrams and use case descriptions were further discussed in Chapter 4.

Furthermore, interface flow diagrams and interactive prototypes designed to illustrate the interface design of the system and interaction between each page of the system. During the process of creating the user interfaces, issues could be identified and resolved before entering the development of the application. In this project, a few prototypes prepared as the system consists of a web application and a mobile application for different users. The prototypes just concentrated on the design and simple navigation of the various pages. The prototype generated served as a guide for the actual application; however, tweaks and improvements was made throughout development. All the interface flow diagrams and each page of the interactive prototypes were shown in Chapter 4.

3.2.3 Development and Testing

In development and testing, all the discussed and analysed tasks in the previous two phases were implemented. According to the phased development methodology, this phase was split into three iterations. The connection between the applications and the database as well as system setup was the primary emphasis of phase one. In phase two, the web-based application for management teams was developed while in phase three, the cross-platform mobile application for residents, visitors, and security guards was developed. The most important features was developed and tested first, followed by the others. The features produced were merged and tested together at the end of each sub-phase to guarantee that the system will not fail.

3.2.3.1 Phase One

All development and testing tools were set up first in phase one. The next step was to document the version of each tool used in this project. To set up the server's connection later, the application folders for both applications were created. The connections were then configured and tested to confirm that the server, database, and apps can all communicate. Since most system functionalities need a database to perform CRUD activities, therefore, it is important to set up the server and database connection first before developing both applications. This ensures that the developer can conduct testing seamlessly. In the middle of the development process, setting up the server and database connection is tedious and time-consuming. A version control system, Git was set up in order to manage source code changes on a regular basis.

3.2.3.2 Phase Two

The main focus in phase two was the development of the web application. According to the priority of the feature of the web application, the main features were divided into numerous sub development and testing phases. Unit testing was done following the implementation of each feature to look for errors in that feature. After all of the features had been thoroughly designed and tested,

integration and system testing were carried out to ensure that the web application functions correctly.

3.2.3.3 Phase Three

In the third phase, mobile application development took place. The development of a mobile application, like phase two, was broken down into multiple sub-phases. The sub-phases were completed in order of the necessity of the mobile application's features. Once each feature had been developed, unit testing was carried out. Lastly, integration testing and system testing were conducted to make sure the mobile application works without errors.

3.2.4 Closing

After the system has been developed and tested, a user acceptance test (UAT) was performed to evaluate and validate both applications. Targeted testers and tested features were defined. Then, suitable testers were invited to conduct user acceptance testing. A list of users' feedback was collected after UAT. Project documentation that provided a detailed description of the system was created. In the documentation, system explanations, screenshots of the finished product, test scenarios, and so on were included. Presentation slides are created to highlight the project's progress and outcomes after the documentation is completed.

3.3 Development Tools

The development tools used in this project includes Axure RP9, Visual Studio Code, React Native, ReactJS, NextJS, Supabase, Expo, and Git.

3.3.1 Axure RP9

Axure RP9 is an all-in-one software design tool for creating prototypes, specifications, and diagrams. Axure RP enables the creation of prototypes of web applications and mobile applications with interactive user interfaces

without the need for coding. It is achieved by setting up triggers for an event to make the prototype interactable and lively. It is used to build an interactive prototype for both applications.

3.3.2 Visual Studio Code

Visual Studio is an open-source text editor which supports a variety of programming languages such as JavaScript, Java, PHP, Python and so forth. It is made by Microsoft and supports Windows, Linux and macOS. Studio Code has a wide range of extensions that can assist developers to speed up the programming process. For example, IntelliSense supports debugging, code completion, rich semantic source code understanding, refactoring and navigation. This will greatly ease the development process when coding, refactoring or debugging the code.

3.3.3 React Native

React Native, an open-source JavaScript UI framework that enables developers to create apps for iOS, Android and other platforms, was created by Meta Platforms, Inc. It has a clean, fluid, and responsive user interface that reduces load time significantly. It's also considerably quicker and less expensive to build React Native apps than it is to build native apps, without sacrificing quality or functionality. It is used to develop the cross-platform mobile application.

3.3.4 ReactJS

ReactJS is a frontend framework developed by Meta that is based on JavaScript. It is best known for its virtual DOM feature. As it is a component-based approach, it enables developers to build UI components that are reusable. It uses a special JSX syntax that allows the mixing of HTML and JavaScript in a file. ReactJS is supported by the growing community and solid corporate support, therefore, React code is easy to update and maintain due to its modular structure.

It is used to develop the web-based application of the Residents and Visitors Management System.

3.3.5 Next.js

Next.js, created by Vercel, is an open-source web development framework that creates static websites and allows server-side rendering for React-based online applications. Traditional React applications can only generate their content on client-side browsers, however, Next.js expands this functionality to encompass server-side applications.

3.3.6 Supabase

Supabase is an open-source backend-as-a-service that offers an instant RESTful API which is often known as Firebase alternative. It offers real-time connectivity with a SQL database over WebSockets, allowing us to execute joins and create stored procedures. Supabase is compatible with a wide range of tools and frameworks because it is PostgreSQL based. It employs a PostgreSQL database with real-time capabilities and allows for the creation of tables and relationships, all of which are crucial to this project. As a result, it will act as the backend and database for this project.

3.3.7 Expo

Expo is a framework used to create React Native applications. It is essentially a collection of services and tools created specifically for React Native that will make it simple for developers to start creating React Native applications. Expo pre-loades with many native APIs for both iOS and Android. This makes it relatively simple for developers to add native functionality to the app. The file system, camera, location, push notifications and social authentication are just a few of the typical native features offered by Expo.

3.3.8 Git

Git is a version control system that allows to track of changes on any file. To manage the source codes, it must be installed locally. As there will be numerous versions or amendments of source code all across the development of the system, implementing Git into this project can help to track all modifications. Through Git, it is easier to perform better corrections on the mistakes if any error occurs by converting back to the older version and compared to the latest version.

3.4 Work Breakdown Structure (WBS)

1.0 Planning
- 1.1 Analyse Project Title
- 1.2 Study Problem Background
- 1.3 Identify Problems Statements
- 1.4 Define Project Objectives
- 1.5 Propose Project Solution
  - 1.5.1 Research Similar Solutions
  - 1.5.2 Compare Similar Solutions
  - 1.5.3 Finalise Project Solutions
- 1.6 Propose Project Approach
  - 1.6.1 Propose Research Approach
  - 1.6.2 Propose Development Approach
- 1.7 Define Project Scope
  - 1.7.1 Identify Target Users
  - 1.7.2 Determine System Scope
  - 1.7.3 Define Modules Covered
- 1.8 Requirement Gathering
  - 1.8.1 Conduct Online Questionnaires
    - 1.8.1.1 Design Questions
    - 1.8.1.2 Review Questions
    - 1.8.1.3 Refine Questions
    - 1.8.1.4 Distribute Questionnaires

1.8.1.5 Analyse Questionnaire Findings

1.8.2 Review Similar Systems

1.8.2.1 Research Similar Systems

1.8.2.2 Identify Common Features

1.8.2.3 Compare Common Features

1.8.3 Literature Review

1.8.3.1 Identify Research Areas

1.8.3.2 Research Identified Research Areas

1.8.3.3 Review Literature Study

1.8.3.3.1 Development Frameworks on Web Application and Mobile Application

1.8.3.3.2 Database Used on Web Application and Mobile Application

1.8.3.3.3 Evaluation Method of Web Application and Mobile Application

1.9 Requirement Elicitation

1.9.1 Choose Recommended Features

1.9.2 Draft Functional Requirements and Non-Functional Requirements

1.9.3 Review Requirements and Non-Functional Requirements

1.9.4 Refine Requirements and Non-Functional Requirements

1.10 Project Scheduling

1.10.1 Create Work Breakdown Structure

1.10.1.1 Define Main Activities

1.10.1.2 Breakdown Activities into Smaller Task

1.10.2 Create Gantt Chart

1.10.2.1 Identify Task Dependency

1.10.2.2 Estimate Duration of Project

1.10.2.3 Draft Gantt Chart

1.10.2.4 Review Gantt Chart

1.10.2.5 Finalise Gantt Chart

2.0 Analysis and Design

- 2.1 Design Use Case Diagrams
- 2.2 Generate Use Case Descriptions
- 2.3 Design Entity Relationship Diagram
- 2.4 Design Interface Flow Diagrams
- 2.5 Design Prototypes

3.0 Development Phase One

- 3.1 Set Up Connection
  - 3.1.1 Set Up Repository for Web Application
  - 3.1.2 Set Up Repository of Mobile Application
  - 3.1.3 Configure Server and Database
  - 3.1.4 Connect Web Application to Server and Database
  - 3.1.5 Connect Mobile Application to Server and Database
- 3.2 Test Connection
  - 3.2.1 Test the Connection of Web Application, Server and Database
  - 3.2.2 Test the Connection of Mobile Application, Server and Database

4.0 Development Phase Two

- 4.1 Develop Web Application
  - 4.1.1 Create Web Application Framework
  - 4.1.2 Create Login Feature
    - 4.1.2.1 Create Login Page User Interface
    - 4.1.2.2 Implement Algorithm to Allow User Login
  - 4.1.3 Test Login Feature
    - 4.1.3.1 Test Login Algorithm
  - 4.1.4 Create Manage Resident's Registration Feature
    - 4.1.4.1 Create Residents's Registration List User Interface
    - 4.1.4.2 Implement Algorithm to Read All Registration Request

6.0 Closing

6.1 Conduct User Acceptance Testing

6.2 Create System Documentation

6.3 Finalise Documentation of Project

3.5 Gantt Chart

3.5.1 Overview

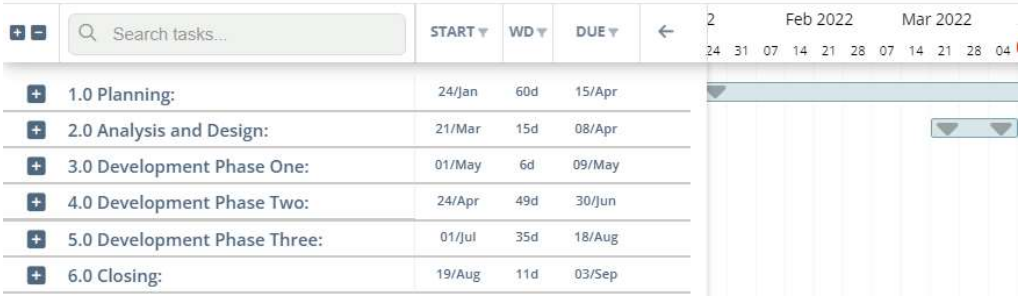


Figure 3.2 Overview Of Project Schedule

3.5.2 Planning Phase

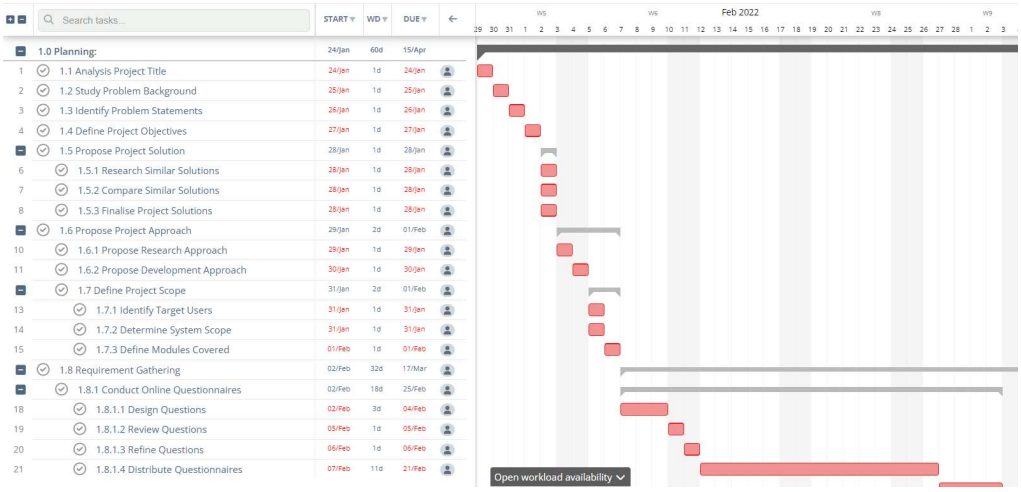


Figure 3.3 Planning Phase Schedule

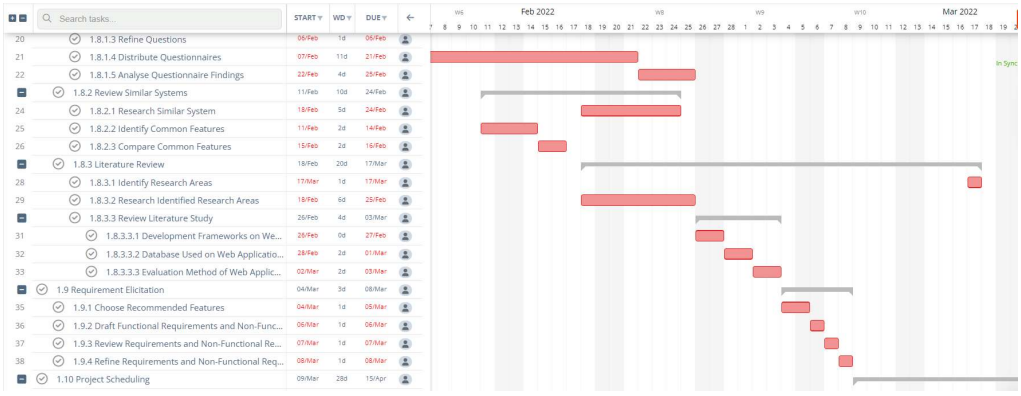


Figure 3.4 Planning Phase Schedule (cont.)

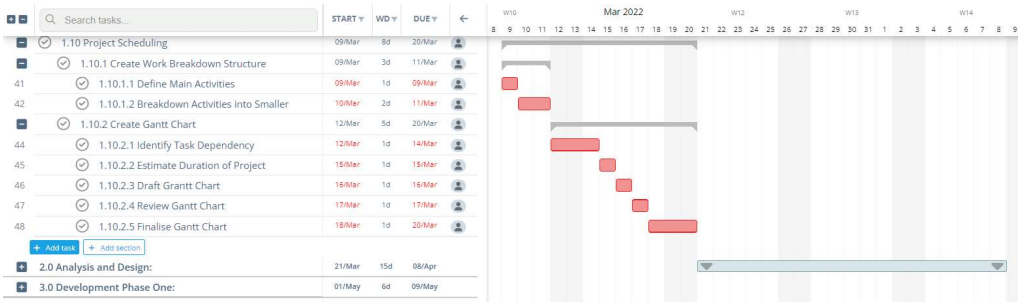


Figure 3.5 Planning Phase Schedule (cont.)

3.5.3 Analysis and Design Phase

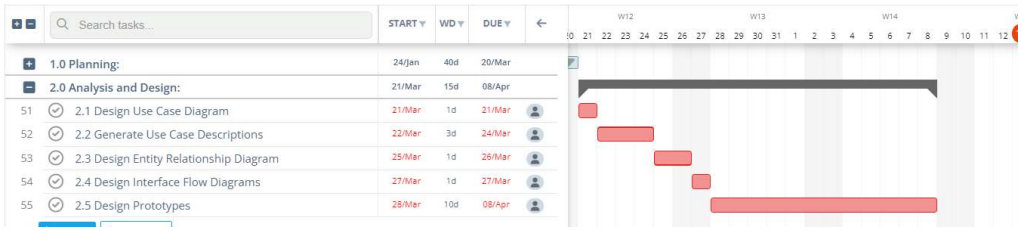


Figure 3.6 Analysis and Design Phase Schedule

3.5.4 Development Phase 1

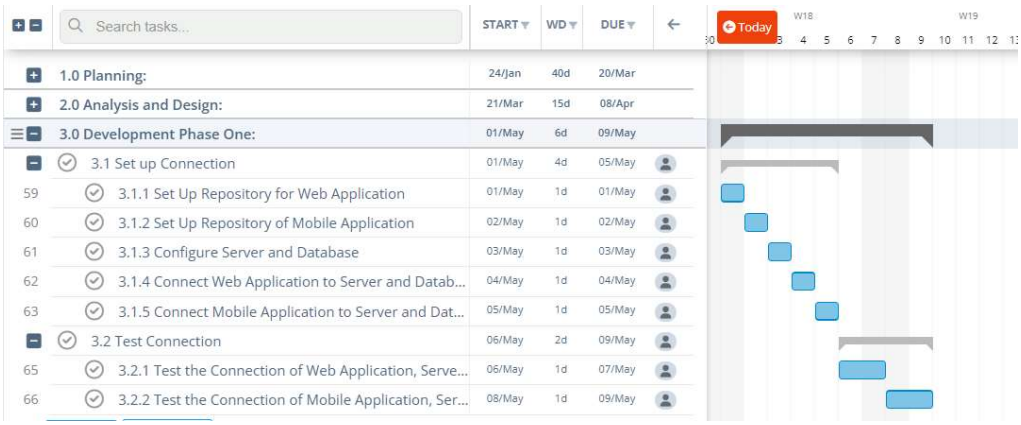


Figure 3.7 Development Phase 1 Schedule

3.5.5 Development Phase 2

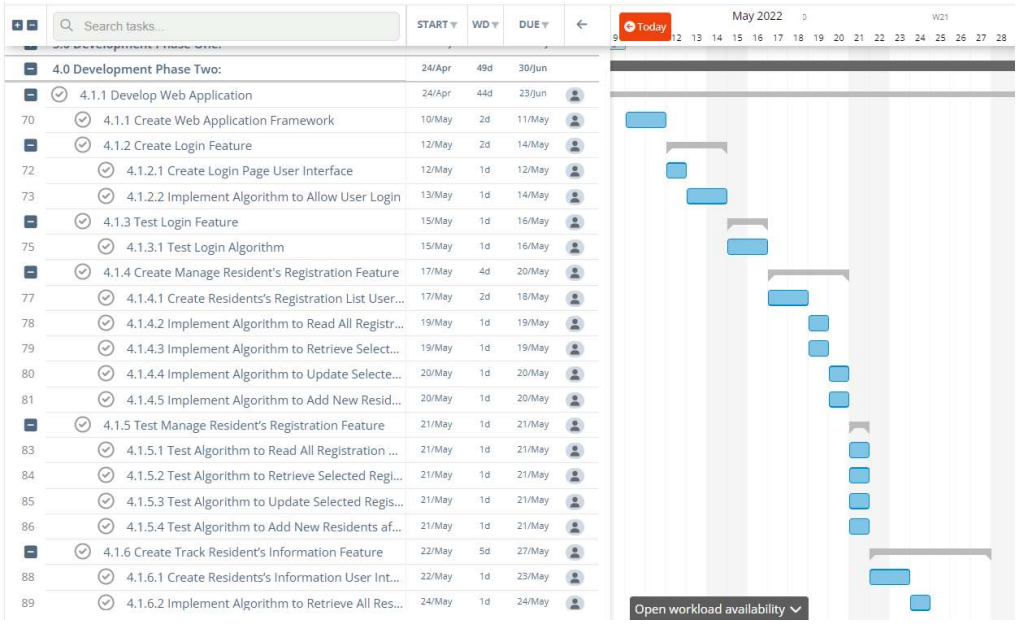


Figure 3.8 Development Phase 2 Schedule

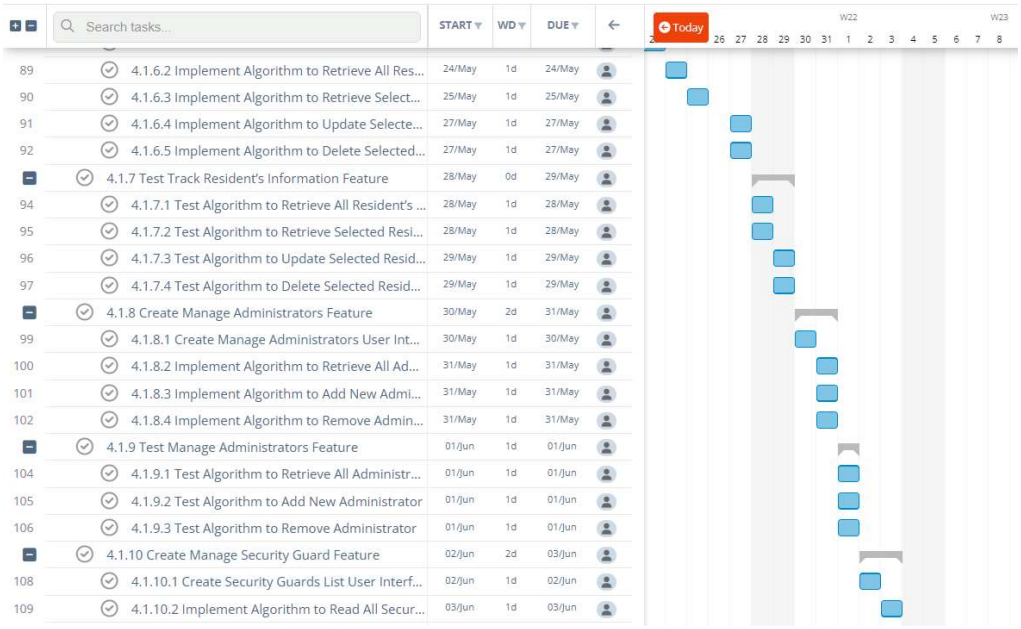


Figure 3.9 Development Phase 2 Schedule (cont.)

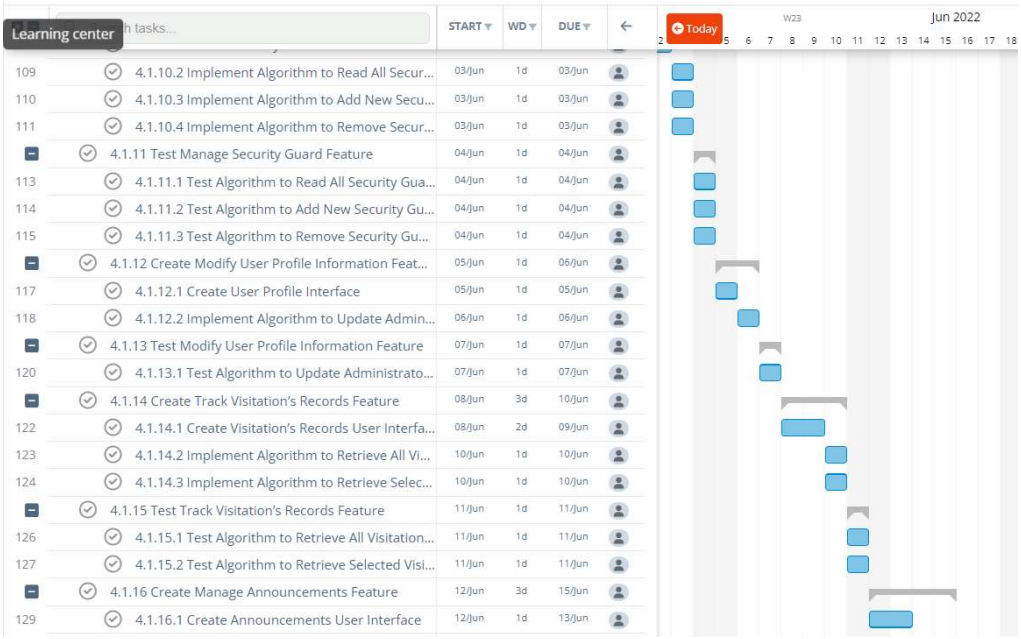


Figure 3.10 Development Phase 2 Schedule (cont.)

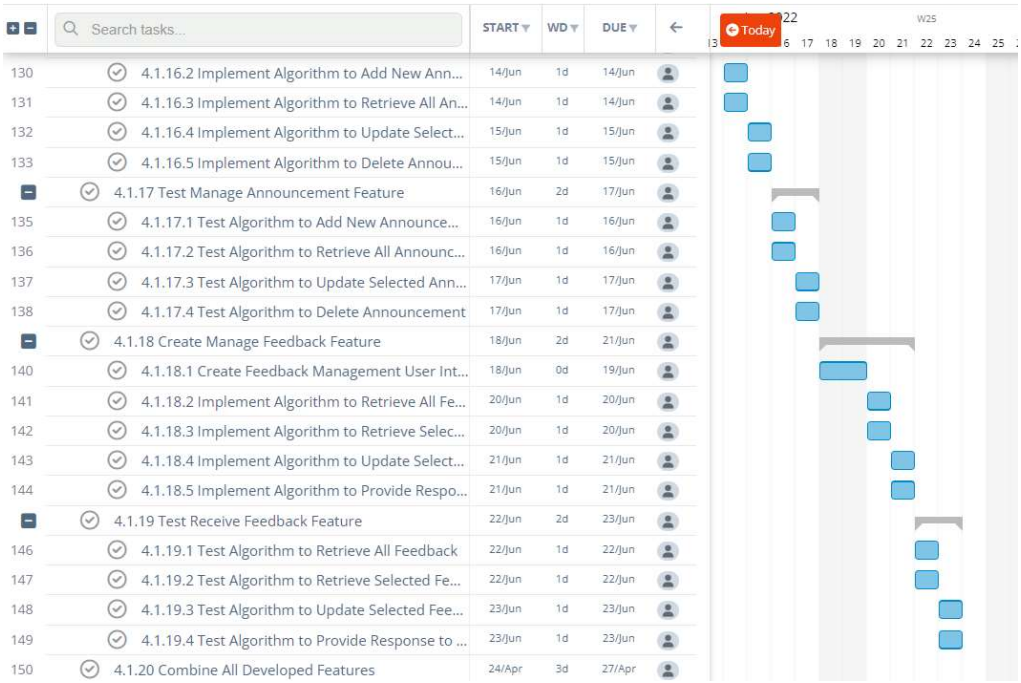


Figure 3.11 Development Phase 2 Schedule (cont.)

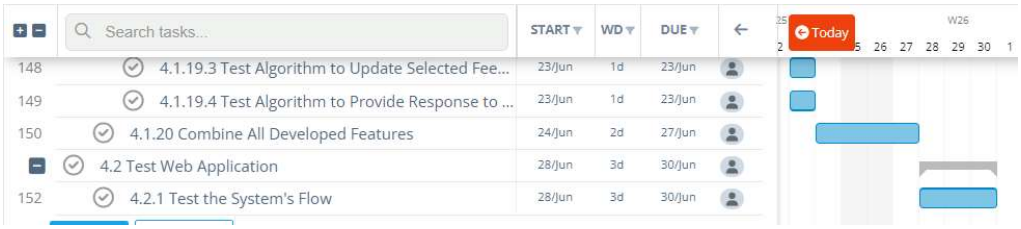


Figure 3.12 Development Phase 2 Schedule (cont.)

3.5.6 Development Phase 3

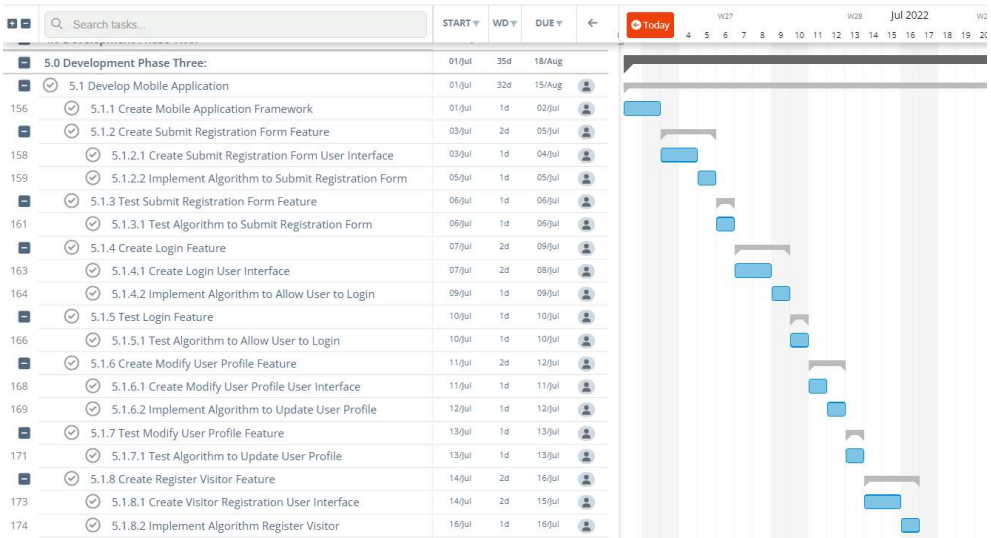


Figure 3.13 Development Phase 3 Schedule

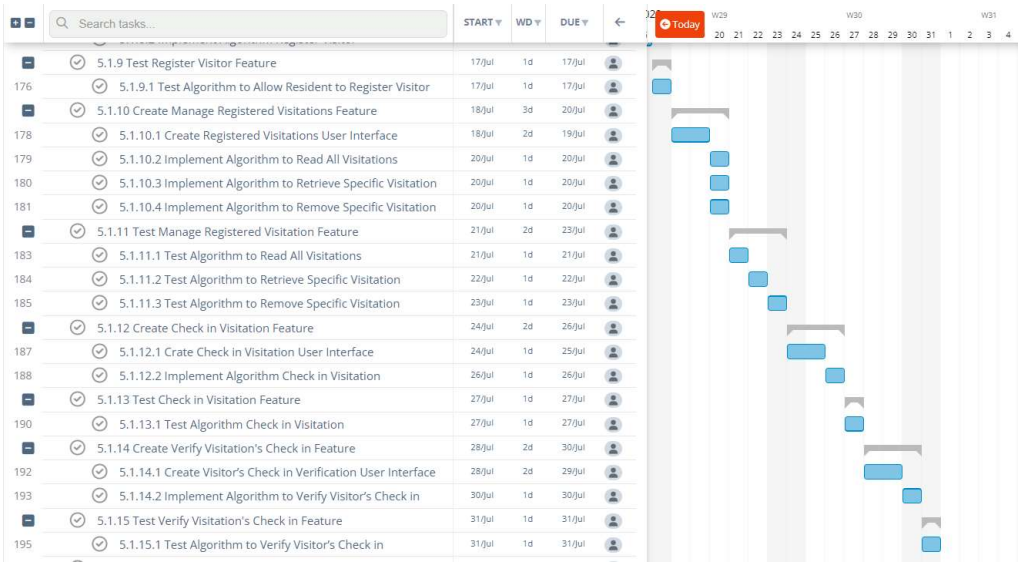


Figure 3.14 Development Phase 3 Schedule (cont.)

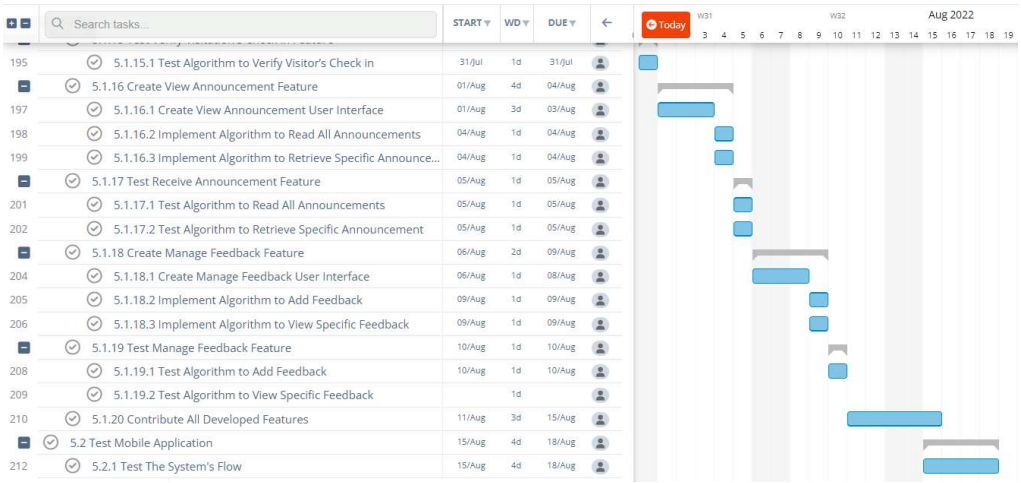


Figure 3.15 Development Phase 3 Schedule (cont.)

3.5.7 Closing Phase

Figure 3.16 Closing Phase Schedule

3.6 Summary

In conclusion, the phased development methodology was used for this project's system development life cycle. This methodology's four primary phases were defined, as well as the procedures and activities involved with each level. In addition, six development tools had been identified and chosen as the project's primary development tools. Finally, a WBS and a Gantt chart of the project were also provided.

CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction

This chapter included the analysis of gathered requirements through questionnaires conducted. The requirement specifications were listed. Besides, use case diagrams, interface flow diagrams, and user interfaces for both web and mobile applications were built and covered in this chapter to give a better view of the systems' functionalities and designs.

4.2 Fact finding

Online questionnaires were created and distributed to collect information from the intended users in order to better understand the system's requirements. 2 sets of questions were prepared. 1 set was for management teams of guarded residential areas and 1 set was for residents. All the target users must be over 18 years old. Both sets of questions had been distributed to the respondents via Google Form. It was a cost-effective and efficient method of obtaining public responses. A total of 30 responses was collected from the residents, however, a total of 16 responses was only collected from the management teams. Both sets of questions were split into 3 sections. Section A was used to collect demographic information, while Section B and C were used to collect users' opinions and experiences on the Residents and Visitors Management System.

4.2.1 Responses of Questionnaire from Management Teams

4.2.1.1 Section A

In this section, demographic information like age and gender are collected.

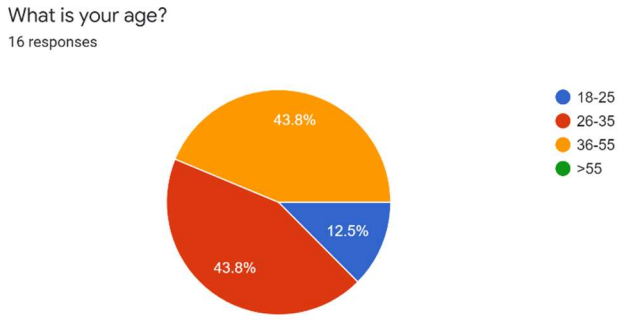


Figure 4.1 Age of Respondents

The first question of the questionnaire is intended to investigate the age range of the respondents. Figure 4.1 above shows most of the respondents are aged from 26 to 35 and from 36 to 55 with a total of 7 respondents for each age range. The remaining 2 respondents out of 16 respondents fall within the range of 18 to 25 years old. This question has shown the questionnaire is answered by all the aged group above 18 years old and different views from various age group can be collected in this survey.

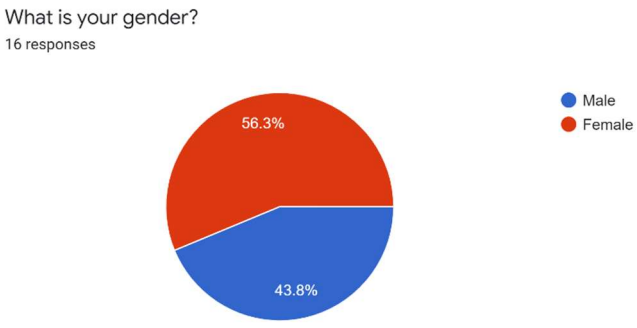


Figure 4.2 Gender of Respondents

The following question is to study the gender of the respondents. Based on the data collected, 9 respondents are reported as female and the remaining 7 respondents are reported as male.

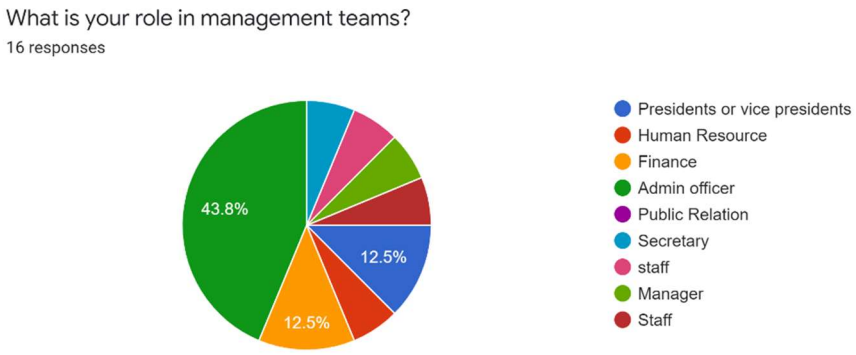


Figure 4.3 Role of Respondent in Management Teams

The third question is to study the role of respondents in the management team to collect different views from various roles. There is a lot of roles found in this question. These include presidents or vice presidents, human resources, finance, admin officer, public relation, secretary, staff and managers. Admin officers are reported with the highest responses with the support of 7 respondents. The president or vice president or management team member from the finance department is the next followed by the admin officer with 2 respondents supporting each. The remaining answers are only reacted with 1 respondent each.

4.2.1.2 Section B

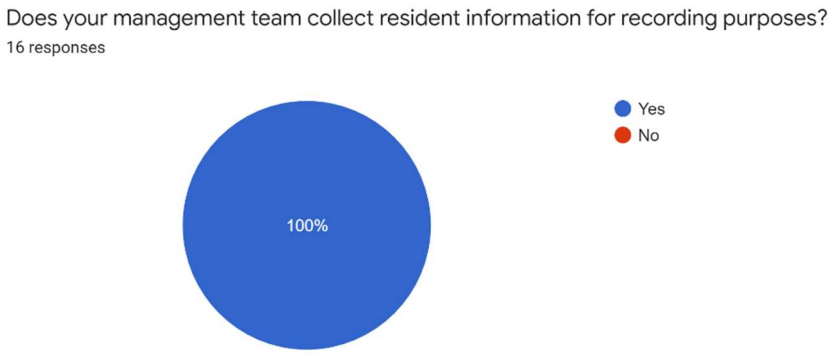


Figure 4.4 Collect Resident Information

In Section B, the initial question is asking about whether the management team collect their resident information for recording purpose. All respondents have responded that they do collect their resident information.

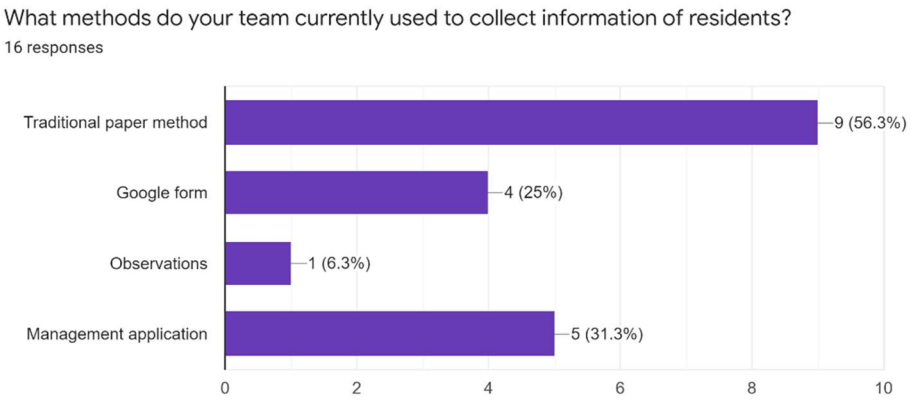


Figure 4.5 Methods Used to Collect Information Of Resident

In this question, the methods the respondents currently used to collect residents' information are asked. 4 selections including traditional paper method, paper form, observations, and management application are provided. Respondents can choose more than one answer for this question. Most of the respondents said that the traditional paper method is used. There are 5 respondents who are using management applications. While the other 2 selections which are Google form and observations are supported by 4 respondents and 1 respondent respectively.

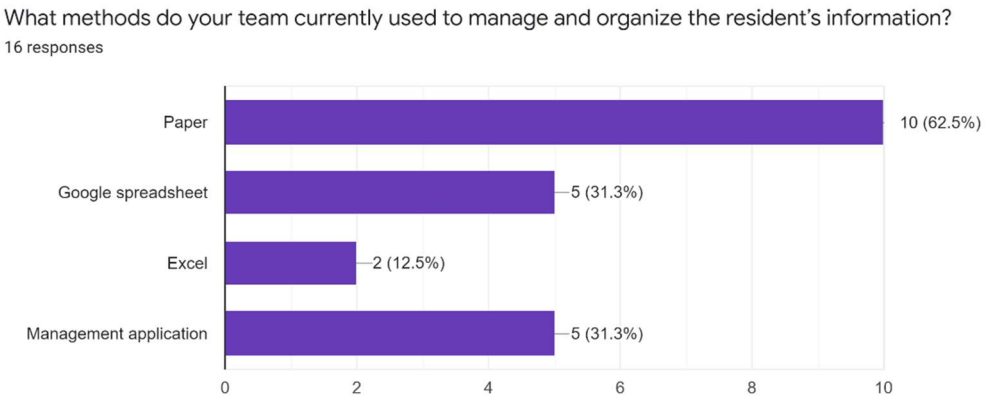


Figure 4.6 Methods of Manage and Organize Resident Information

According to the data collected, the third question is to study how the management teams manage and organize the residents' information. There are

4 selections provided. For instance, paper, Google Spreadsheet, Excel and management applications. Respondents are allowed to select more than one answer for this question. Out of 16 respondents, 10 respondents reacted that papers are used by them to manage and organize the resident's information. Google Spreadsheet and management application are supported by 5 respondents each. Lastly, there are 2 respondents who reacted that they are using Excel.

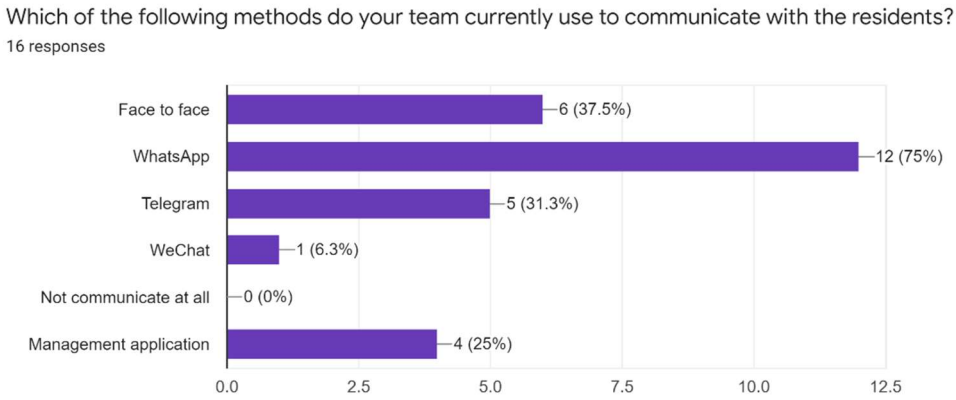


Figure 4.7 Methods of Comminucation with Residents

Figure 4.7 visuallises the question about methods of the management teams communicating with the residents. Face to face, WhatsApp, Telegram, WeChat, Not communicate at all, Management application are provided as the answers of this question. This is a There is 0 respondent reacts to the no communication at all option. WhatsApp is the most welcomed messaging platform which supported by 12 respondents out of 16 respondents. Meanwhile, communication through face to face method, Telegram method, and Management application are supported by 6, 5 and 4 respondents each. For the option of WeChat, there is only 1 respondent are recorded in this survey.

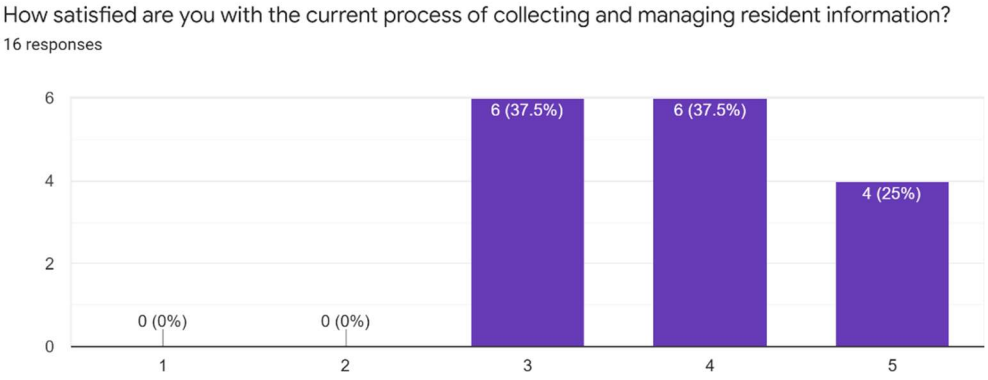


Figure 4.8 Satisfaction of Respondents on Current Collecting and Managing Process

In this question, the respondents are asked whether they are satisfied with the current collecting and managing residents method. Scales 1 (very not satisfied) to 5(very satisfied) are provided to respondents to choose from. Each respondent is only allowed to choose 1 answer for this question. Among all respondents, there are no respondents not satisfied (2) or very not satisfied (1) with the current process used. There are 6 responses collected for each neutral (3) and satisfied (4) option which are the highest among all the respondents. 4 respondents reported being very satisfied (5) with the current process of collecting and managing resident information.

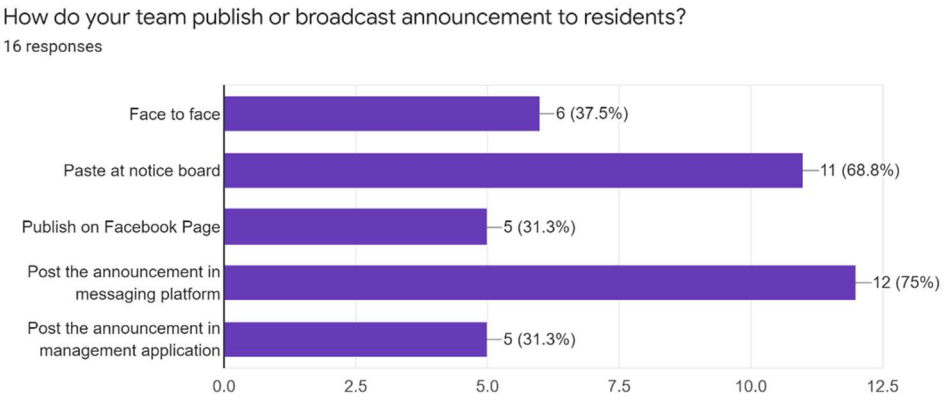


Figure 4.9 Methods of Respondents Used to Publish or Broadcast Announcement

The following question is asking about how the management teams publish or broadcast announcements to residents. There are 5 selections offered to the respondents and each respondent is allowed to choose more than one answer for this question. The selections offered included face to face, paste at the notice board, publishing on Facebook Page, posting the announcement on messaging platform, and posting the announcement on management applications. The selections with the highest supporters are posted the announcement on messaging platforms and pasted on notice boards which is supported by 12 respondents and 11 respondents. Face to face method is reacted by 6 respondents. There are 5 respondents who choose to publish on their Facebook Page and also 5 respondents reported posting the announcement in the management application.

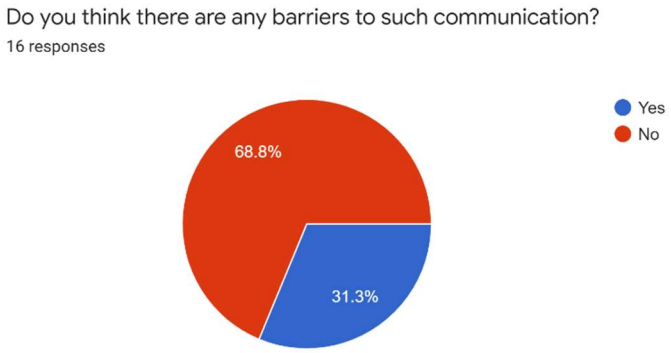


Figure 4.10 Thought on Barriers of Communication Method Used

Figure 4.10 shows Question 7 in Section B which is about whether the respondents find there are any barriers to the communication method they are using. 11 respondents responded “no” and the remaining 5 respondents responded “yes”. The 5 respondents may further answer the barriers that they are facing in the next question while the rest can skip the next question.

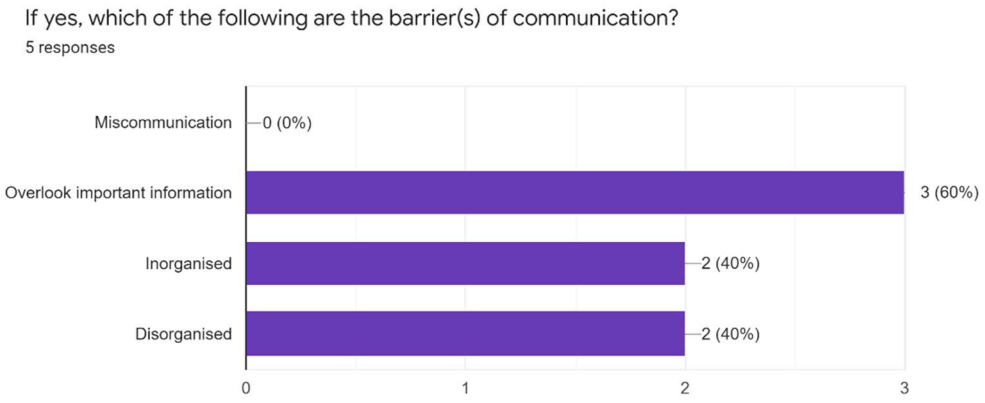


Figure 4.11 Barriers of Communication Method faced

Figure 4.11 shows a multi-selection question. In this question, 3 barriers are included and each respondent can choose more than 1 barrier for this question. From the 5 respondents, the barrier to the current communication method facing is overlooked important information with the support of 3 responses out of 5. Followed by disorganised or unorganised supported by a total of 4 respondents. No respondent selects miscommunication as the barrier of the communication that they are facing.

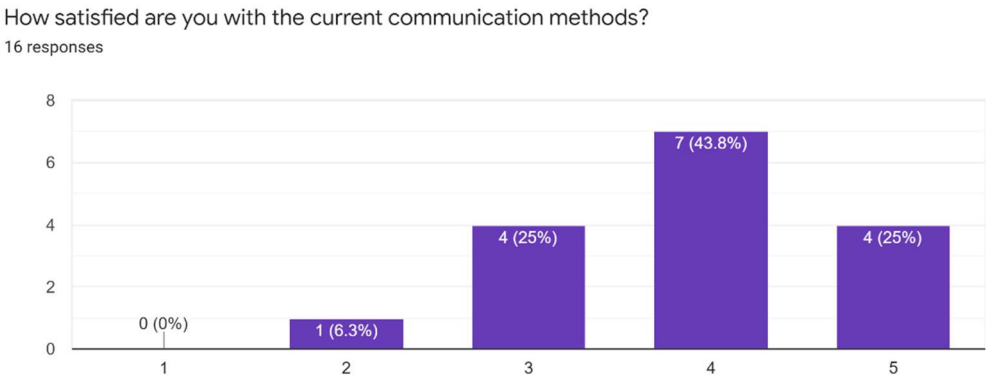


Figure 4.12 Satisfaction of Respondents on the Current Communication Methods Used

According to the data collected, the question illustrates the satisfaction of the management teams with the current communication methods. Scales 1 (very not satisfied) to 5 (very satisfied) are provided to respondents to choose

from. Most management team members (7 respondents) responded they are satisfied with the current communication method. 4 respondents are very satisfied with the current communication methods and 4 respondents are neutral. There is only 1 respondent reported they are unsatisfied with the current communication method and no respondent is very not satisfied with the current communication method.

Figure 4.13 Methods Used te Register Visitor Records

This question is aimed to determine the method the responded management teams used to register visitor records. 9 respondents reported that they are using paper-based logbooks to record registered visitors. Visitor Management Applications are used by 5 respondents and 2 respondents do not use any method to record the visitors.

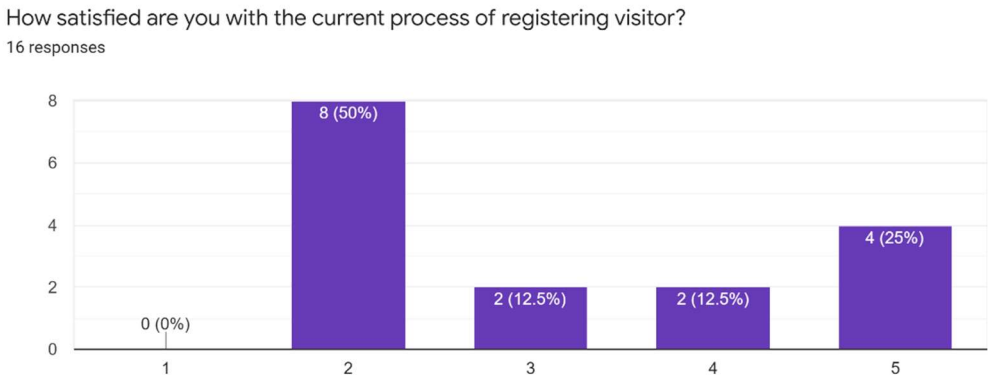


Figure 4.14 Satisfaction of Respondents with The Current Register Visitor Process

Figure 4.14 illustrates the satisfaction of the management teams with the current registering visitors process. Scales 1 (very not satisfied) to 5 (very satisfied) are provided to respondents to choose. There are 8 respondents responded that they are not satisfied (2) with the current visitors registering method. Followed by 4 respondents are very satisfied (5) with the registration method used. The other 4 respondents reacted that they are satisfied (4) and neutral (3) with the current process used with each scale supported by 2 respondents. Lastly, no response of very not satisfied (1) is found.

Figure 4.15 Inconveniences Faced when Using the Current Register Visitor Method

The last question in Section B is to study if the way of recording the registration of visitors brings any inconveniences. This is a multiselection answer and 4 options are provided. The options included waste of time, traffic congestion, low security, and no inconveniences. There are 7 respondents reported that they faced traffic congestion issues when registering visitors. Waste of time is supported by 6 respondents and low security is supported by 5 respondents. There are also 5 respondents found that there are no inconveniences during the process of registering visitors.

4.2.1.3 Section C

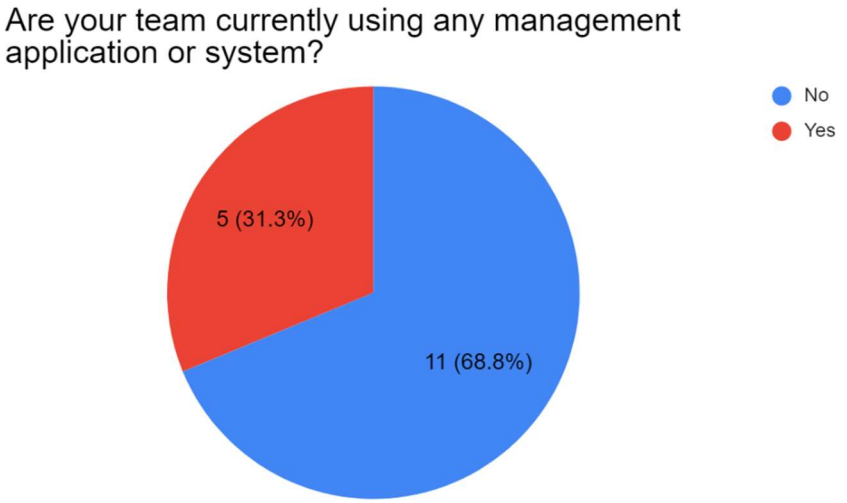


Figure 4.16 Any Management System Used by Respondents

The first question in Section C is about to collect the information if the respondents are using management applications to assist and facilitate their tasks. From Figure 4.16, 11 respondents responded with “no” and 5 respondents responded with “yes”. The 5 respondents who reported using management applications will be answered the next question and the rest will skip the next questions.

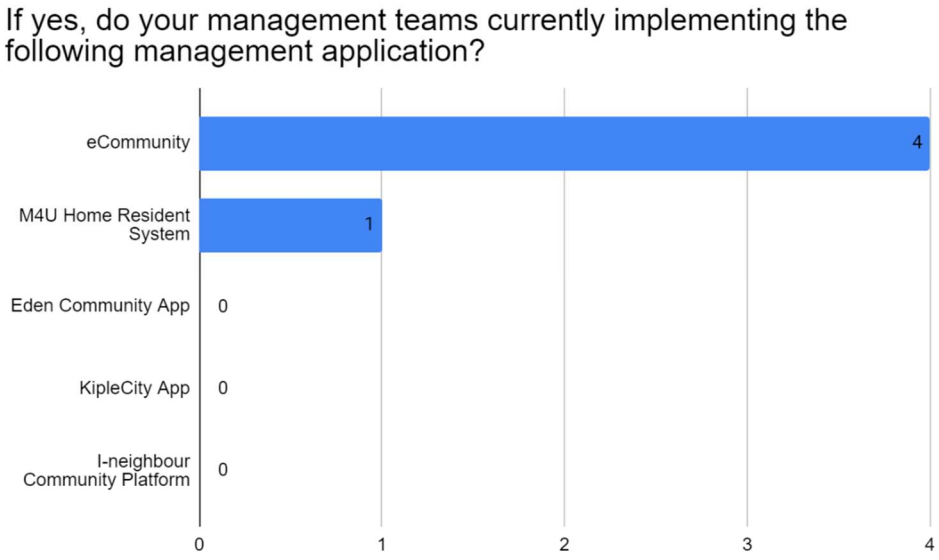


Figure 4.17 Example of Managment Applications

5 management applications are listed in this question. There are eCommunity, M4U Home Resident System, Eden Community App, KipleCity App, and I-neighbour Community Platform. 4 respondents are reported using eCommunity while 1 respondent reacted that he or she are using M4U Home Resident System.

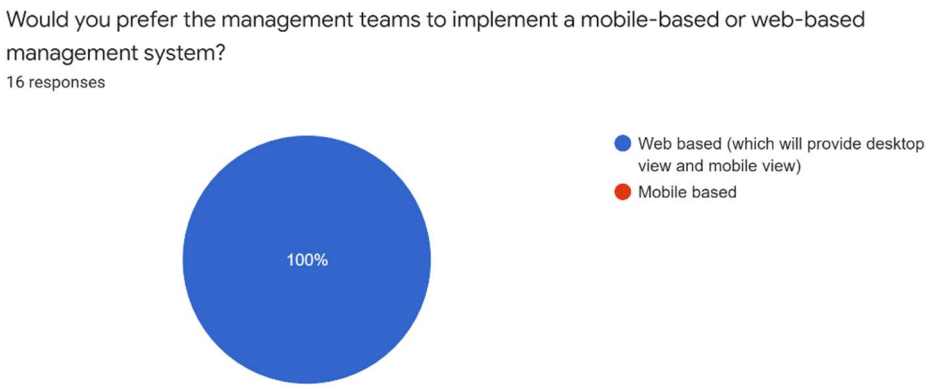


Figure 4.18 Opinion of Respondents on Web Application or Mobile Application

The third question is to study whether the management teams prefer to implement answered they prefer to have web-based management application.

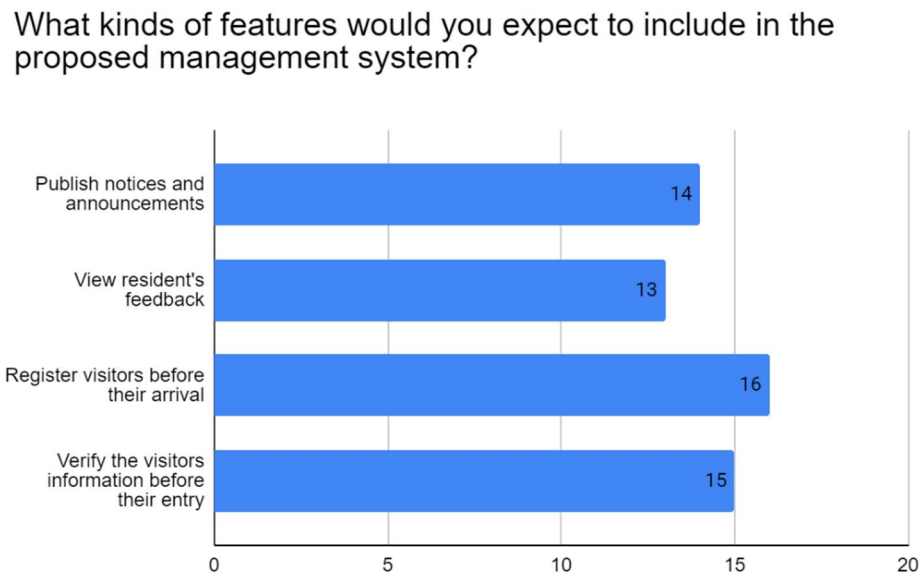


Figure 4.19 Features Expected by The Respondents

According to the fourth question, four features of the management application are provided to discuss which features are preferred by management teams. These features included are publish notices and announcements, view residents' feedback, register visitors before their arrival, and verify visitors' information before their entry. All the respondents supported the features of registering visitors before their arrival. The feature of verifying visitors' information before their entry is supported by 15 respondents and the feature of publishing notices and announcements is supported by 14 respondents. Lastly, there are 13 respondents responded that they expect the feature of view residents' feedback.

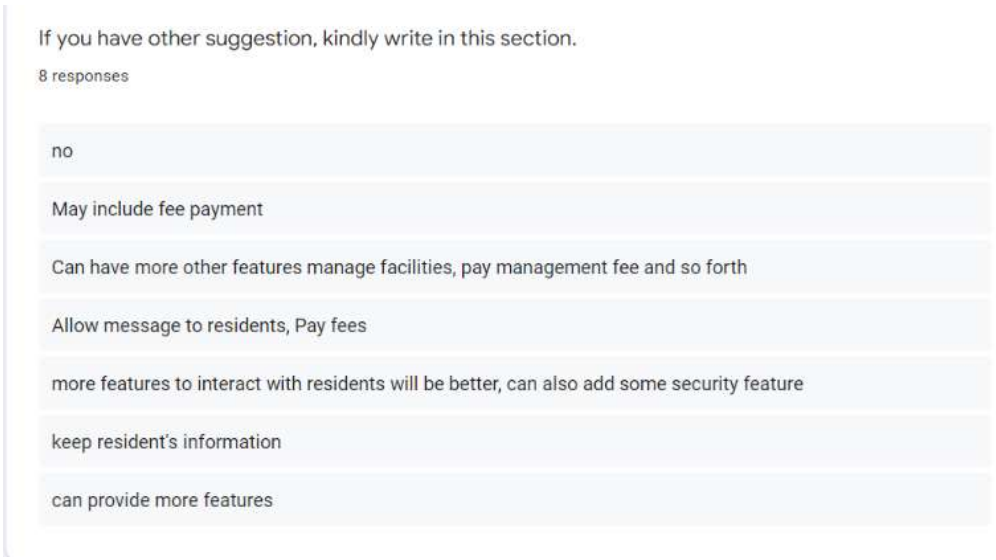


Figure 4.20 Other Suggestion of Feature oof Management Application

The questionnaire's final question asked the respondents to suggest further features. There are 8 responses received in total however there are only 6 responses providing information. A total of 3 respondents suggested fee payment as one of the features. 1 respondent recommended including manage facilities features, and 2 respondents suggested having the features to interact with the residents. Another 1 respondent expected to include more security features and 1 respondent preferred to keep the residents' information.

4.2.2 Responses of Questionnaire from Residents

4.2.2.1 Section A

In this section, demographic information like age and gender are collected.

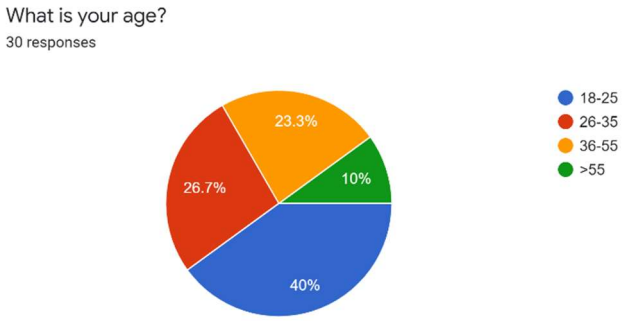


Figure 4.21 Age of Respondent (Residents)

The initial question of the survey is aimed to study the age range of the respondents. Based on the data collected above, most of the respondents are aged 18 to 25 with a total of 12 respondents out of 30 respondents. Then, followed by the age group from 26 to 35 with 26.7% (8 respondents) and the age group from 36 to 55 (7 respondents). There are only 3 respondents aged above 55 are found in this questionnaire. This question has shown the questionnaires are answered by all the age groups above 18 years old and different views from various age groups can be collected in this survey.

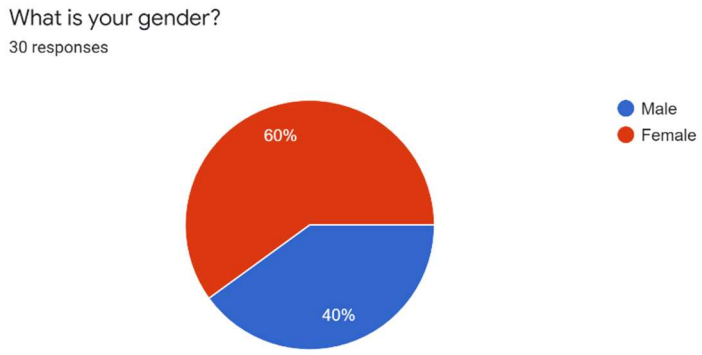


Figure 4.22 Gender of Respondents (Residents)

The second question asked about the gender of each respondent. According to the data shown, 18 respondents are female with a total of 60% and the remaining 40% are male respondents.

4.2.2.2 Section B

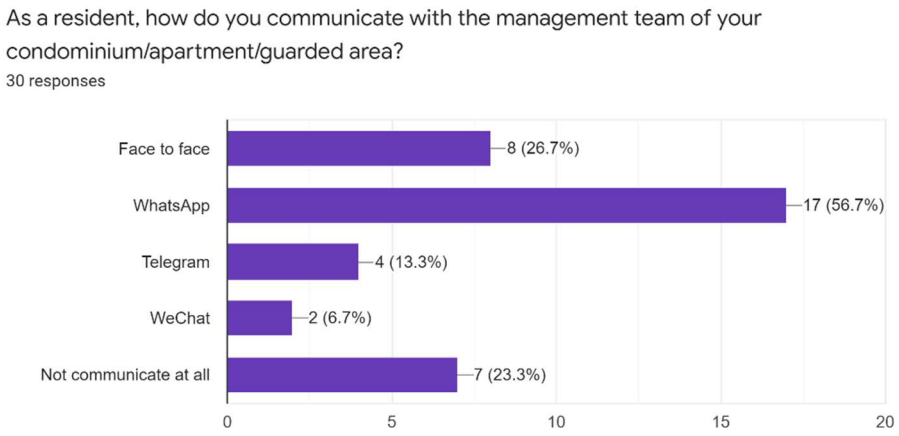


Figure 4.23 Communication Method Of Residents with Management Team

This question is to determine the method residents used to communicate with the management teams of their residential area. From the data collected, WhatsApp is the most used communication platform as it consists of 17 respondents out of 30 respondents. 8 respondents prefer face to face communication with the management teams and 7 respondents never communicate with the management teams. Telegram which consists of 4 respondents and WeChat with 2 respondents are the least 2 methods used by the respondents to communicate with management teams.

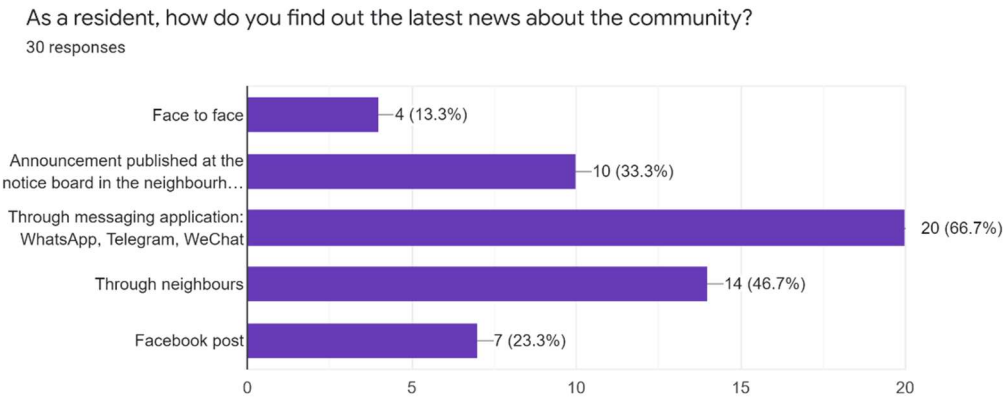


Figure 4.24 Methods of Respondents Learn the News of The Community

The next question is to discover how residents find out the latest news about the community. There are 5 multichoice checkboxes provided which are face to face, announcements published on the notice board in the neighbourhood, through messaging applications such as WhatsApp, Telegram, WeChat, through neighbours, and Facebook posts. Among these 5 selections, reading the latest news through messaging applications: WhatsApp, Telegram, and WeChat are chosen by the most respondents which are 20 out of 30 respondents. Followed by getting the news through neighbours with 14 respondents' responses and through the announcements published on the notice board in the neighbourhood with 10 respondents agreeing with it. Learning the news through face to face and Facebook post are the least popular among 30 respondents as only 4 and 7 responses are recorded respectively.

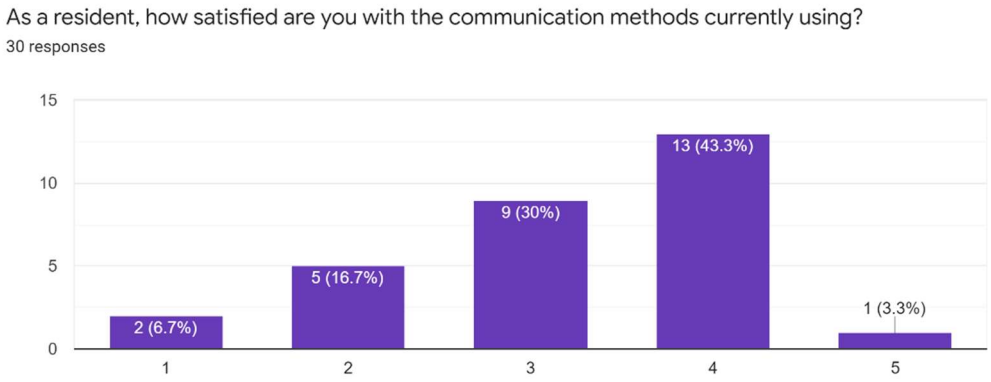


Figure 4.25 Satisfaction of Residents on the Current Communication Method Using

Followed by the question to study the satisfaction of residents with their current communication methods. Scales 1 (very not satisfied) to 5 (very satisfied) are given. Most of the users with 14 respondents are satisfied and very satisfied with the current communication method. 9 out of 30 responses are neutral. There are 5 respondents not satisfied and 2 respondents are very not satisfied with the current communication method.

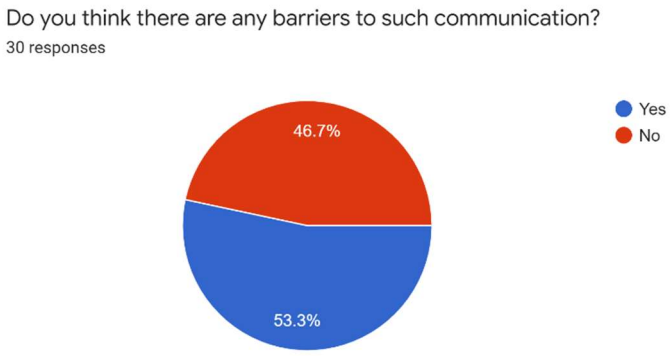


Figure 4.26 Opinion of Barriers of Communication Method Used

Question 4 in Section B is about whether the respondents find there are any barriers to the communication method they are using. 16 respondents responded "yes" and the remaining 14 respondents responded "no". The 16 respondents further respond to the barriers that they are facing in the next question while the rest will skip the next question.

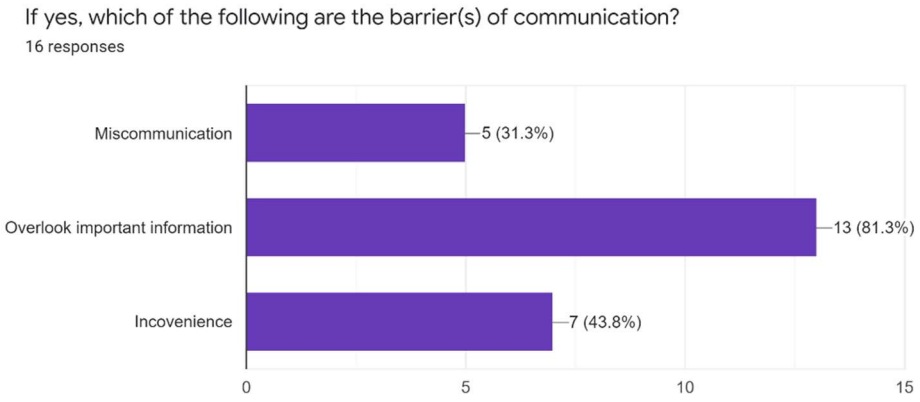


Figure 4.27 Barriers of Communication Method Using Faced

In this question, 3 barriers are provided and each respondent can choose more than 1 barrier for this question. From the 16 respondents, the barrier of the current communication method facing is overlooked important information recorded with 13 responses. Followed by inconvenience supported by 7 respondents. There are only 5 respondents selected miscommunication as the barrier of the communication that they are facing.

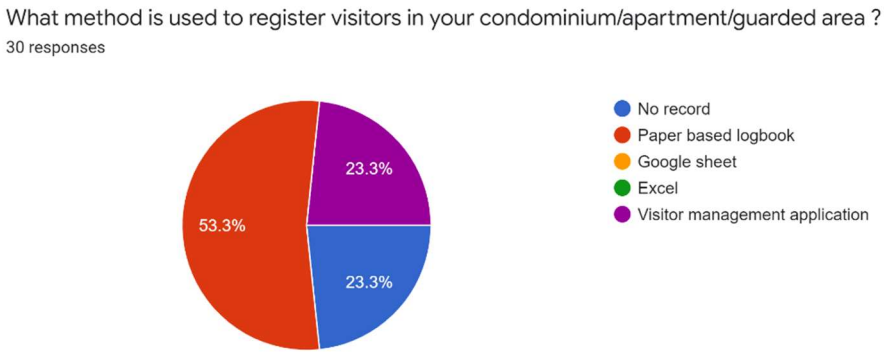


Figure 4.28 Method Used by Respondents to Register Visitor

Figure 4.28 shows question 6 which is about the method of registering visitors by the management team in the respondent's residential area. 53.3% of the respondents (16 respondents) reported using paper-based logbooks. There are 7 respondents reacted with no record of any registration of visitors and 7 respondents responded by using visitor management applications. Google Sheet and Excel are not selected by any of the respondents.

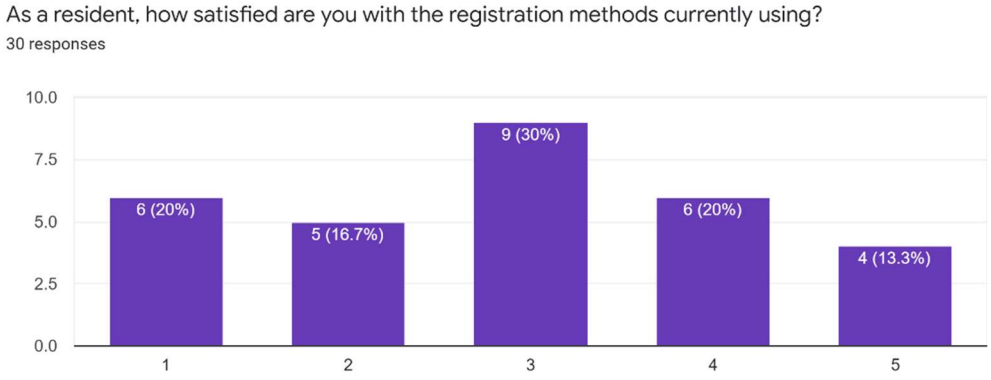


Figure 4.29 Satisfaction of Current Register Visitor Method

The question in Figure 4.29 above is to study if the respondents are satisfied with the current registration methods used. Scales 1(very not satisfied) to 5(very satisfied) are provided to respondents to answer. Most respondents with a number of 9 out of 30 respondents are neutral to the registration methods. 4 respondents are very satisfied with the current registration methods while 6 respondents are satisfied. The remaining 11 respondents react with very not satisfied and not satisfied.

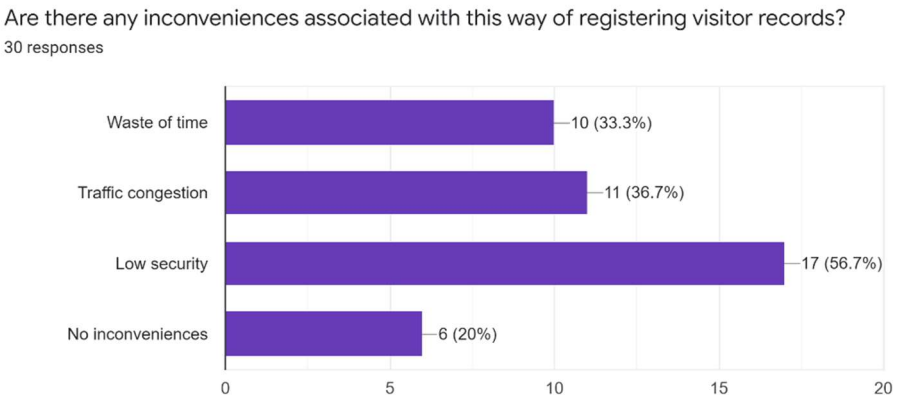


Figure 4.30 Incoveniences Faced when Register Visitors

This question is to discover whether there are any inconveniences associated with the registering visitor's method. 17 respondents found that the registering method they are using are having low security in ensuring the authenticity of the visitors' information. There are 11 respondents who reacted to the traffic congestion as inconvenience and 10 respondents reacted with waste of time as inconveniences faced while registering visitors. However, 6 respondents out of 30 respondents have different opinions as they found no convenience in their registering method.

4.2.2.3 Section C

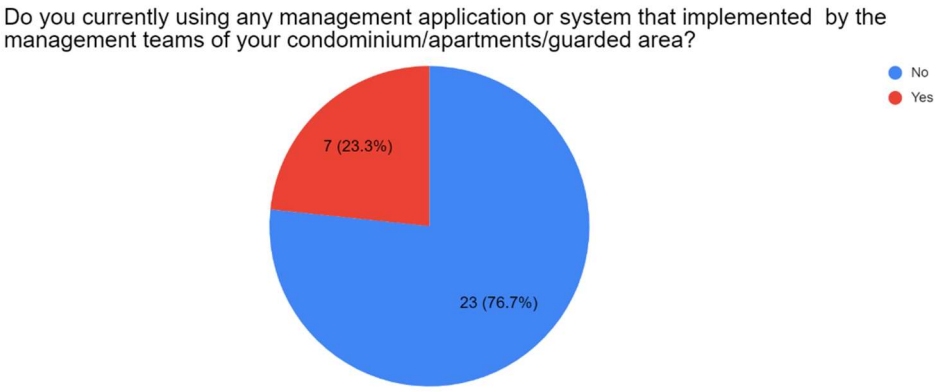


Figure 4.31 Usage of Any Management Applications

The first question in Section C is to study if the management team of a residential area of respondents are using any management application or system. There are 23 respondents responded with “no”. The remaining 7 respondents reacted with “yes” to this question. The 7 respondents who have experience in the resident management application are required to fill in the next question and the others may skip the question.

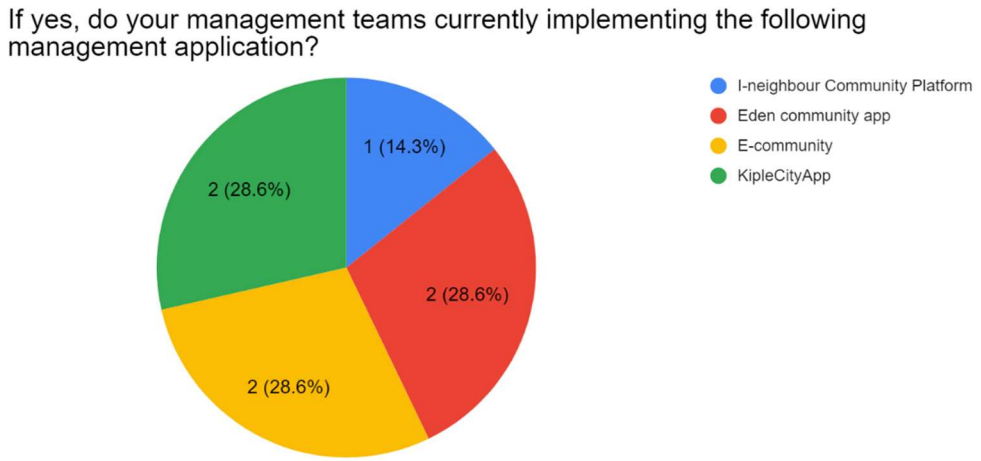


Figure 4.32 Examples of Management Application Used

Figure 4.32 illustrates the management application that the management team of the respondents uses. There are Eden Community App, E-community, and KipleCityApp are used by a total of 6 respondents with 2 respondents each. I-neighbour Community are reported by 1 respondent.

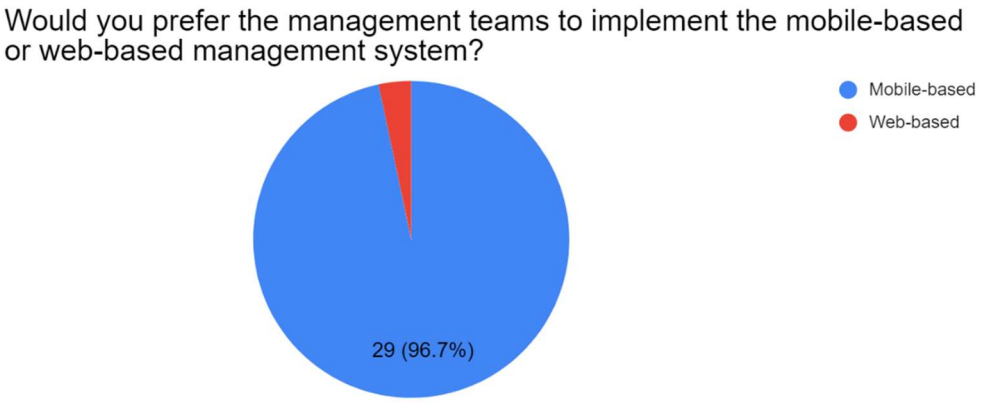


Figure 4.33 Opinions of Respondents on Web Applications and Mobile Application

The third question is to collect the opinion of respondents on whether they prefer their management team to implement mobile-based or web-based management applications. Almost all respondents reacted they prefer mobile-based applications and only 1 respondent prefer web-based applications.

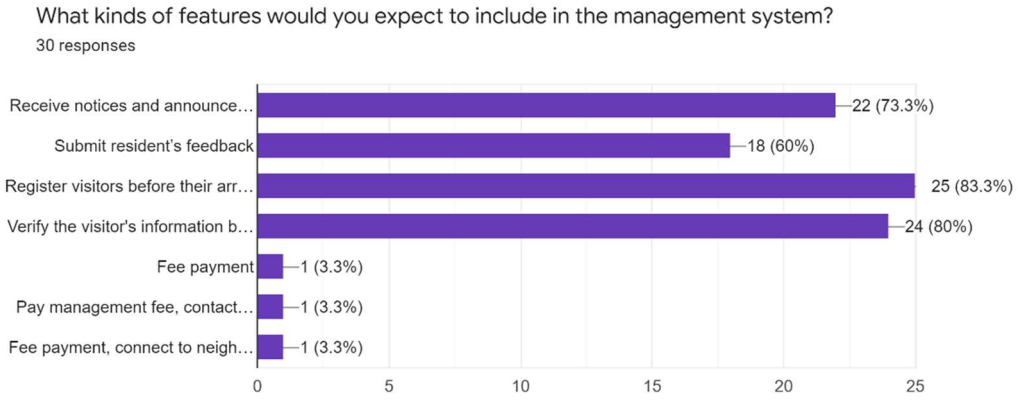


Figure 4.34 Features Expected by Respondents on the System

The last question of this questionnaire collected the features expected in the management team by the respondents. Five options provided to respondents included receiving notices and announcements, submitting residents' feedback, registering visitors before their arrival, verifying the visitor's information before entry, and others. Register visitors before their arrival are the most supported answer by 25 respondents. Followed by the feature of verifying the visitor's information before entry recorded with 24 respondents. There are 22 respondents supported the feature of receiving notices and announcements and 18 respondents supported the feature of submitting residents' feedback. Besides, 3 respondents mentioned the fee payment features should be included in the management application. 1 respondent prefers to have the feature to contact the security guards and another 1 respondent prefers to have the feature to connect with neighbours and contact security guards when emergency.

4.3 Requirement Specification

The functional and non-functional needs were divided into the requirements specification. Furthermore, the functional requirements were classified into two categories: mobile application functional requirements and web application functional requirements. Each requirement was linked to the project scope modules stated in Chapter 1.

4.3.1 Functional Requirement

4.3.1.1 Web Application

W-1. The web application shall allow the management team to login their account.

W-2. The web application shall allow the management team to manage resident's registration.

W-3. The web application shall allow the management team to track resident's information.

W-4. The web application shall allow the management team to manage administrators.

W-5. The web application shall allow the management team to manage security guards.

W-6. The web application shall allow the management team to modify their user profile.

W-7. The web application shall allow the management team to track the visitor's records.

W-8. The web application shall allow the management team to manage announcements to all residents.

W-9. The web application shall allow the management team to manage feedback from residents.

4.3.1.2 Mobile Application

M-1. The mobile application shall allow residents to submit registration form to the management team.

M-2. The mobile application shall allow residents and security guards to login their accounts.

M-3. The mobile application shall allow residents to modify their user profiles.

M-4. The mobile application shall allow residents and security guards to register for visitors.

M-5. The mobile application shall allow residents to manage their registered visitations.

M-6. The mobile application shall allow visitors to check in their visitations.

M-7. The mobile application shall allow security guards to verify check in visitations.

M-8. The mobile application shall allow residents to view announcements published by management teams.

M-9. The mobile application shall allow residents to manage feedback.

Table 4.1 shows the linkages between the determined functional requirements of the Web and mobile applications (Section 4.4.1) and its related use case description (Section 4.4.2).

Table 4.1 Overview of Functional Requirement ID and Related Use Case ID

Functional Requirement ID	Use Case ID
W-1	1
W-2	2
W-3	3
W-4	4
W-5	5
W-6	6
W-7	7
W-8	8
W-9	9
M-1	10
M-2	11
M-3	12
M-4	13
M-5	14
M-6	15
M-7	16
M-8	17
M-9	18

4.3.2 Non-Functional Requirement

1. Performance requirements
    a. The web application's and mobile application's response time shall not be longer than 3 second.
    b. The web application and mobile application shall be able to load the user interfaces within 5 seconds when opening the applications.
2. Security requirements
    a. The web application and mobile application shall authenticate the users with valid email and password before login.
    b. The web application and mobile application shall only allow the authorised users to access the features of the applications.
3. Usability requirements
    a. The web application shall support responsive views in which the user interfaces automatically scale its content and elements to match the screen size.
    b. The number of errors found should not be more than 3 when users attempt to complete a task in the web application and mobile application.
    c. The user interfaces of the web application and mobile application shall be easy to learn and navigate with no prior training.
4. Reliability requirements
    a. The web application and mobile application shall provide warning messages when users perform destructive actions such as deletion of data from the database.
5. Availability requirements
    a. The web application and mobile application shall be available all the time (24/7 online) for all users.
    b. The updated version of the web application and mobile application shall be available for the users within 24 hours.

4.4 Use Case Modelling

4.4.1 Use Case Diagram

1. Web Application

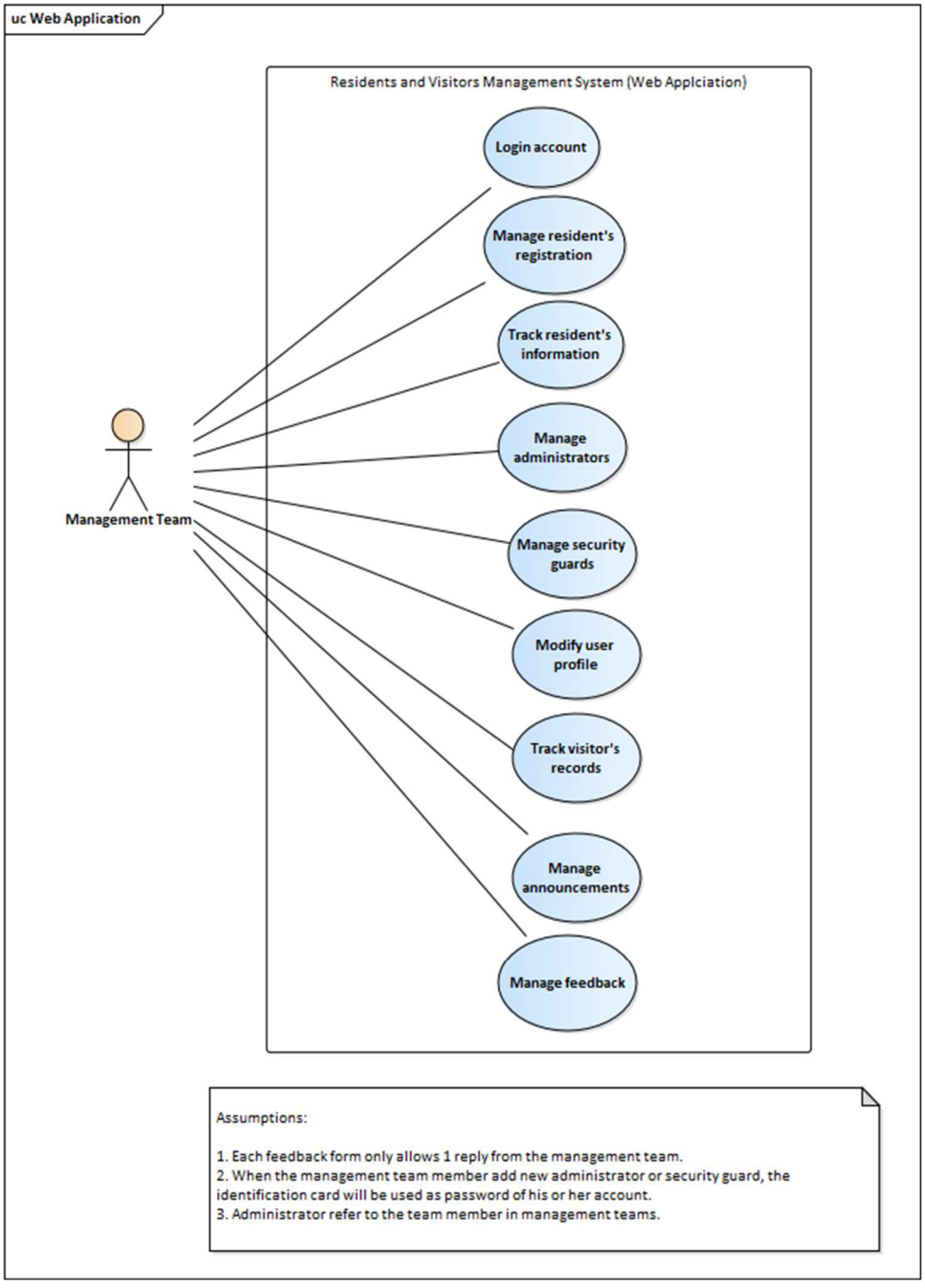


Figure 4.35 Use Case Diagram (Web Application)

2. Mobile Application

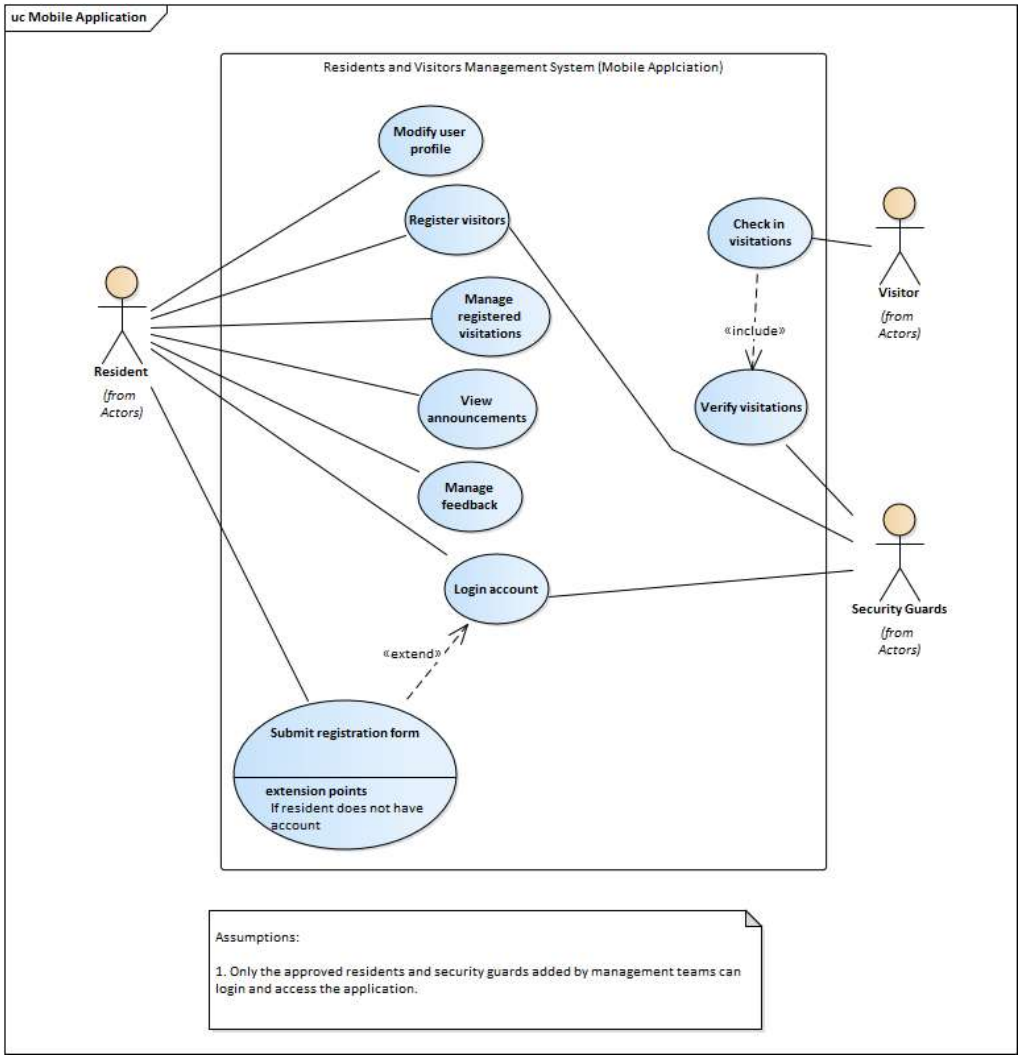


Figure 4.36 Use Case Diagram (Mobile Application)

4.4.2 Use Case Description

Web Application

Table 4.2 Use Case of Login Account

Use Case Name: Login account	ID: 1	**Importance Level: High**
Primary Actor: Management teams	Use Case Type: Detail, Real	
Stakeholders and Interests: Management teams - Management teams can login their account and access their application.		
Brief Description: This use case describes how management teams log into their account before accessing the application.		
Trigger: When a management team member wants to access the application.		
Relationships: Association : Management teams Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The management team member opens the web application. 2. The management team member logs in his or her account with email and password. 3. The application verifies the email and password.		
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.3 Use Case of Manage Resident's Registration

<table>
<tr><td>Use Case Name: Manage resident’s registration</td><td>ID: 2</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: Management teams</td><td>Use Case Type: Detail, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
Management teams - Management teams can manage their resident’s registration.</td></tr>
<tr><td colspan="3">Brief Description:
This use case describes how management teams manage their resident’s registration.</td></tr>
<tr><td colspan="3">Trigger: When a management team member wants to check and update registration status of the residents’ application after successfully logins into his or her account.</td></tr>
<tr><td colspan="3">Relationships:

Association : Management teams
Include : -
Extend : -
Generalization : -</td></tr>
<tr><td colspan="3">Normal Flow of Events:

1. The management team member clicks on the “resident” button on the navigation bar.
2. The web application shows the new applications of residents by default.
3. The management team member views the resident’s information from the list.
4. The management team member downloads the supporting document to verify the residents’ information.
5. The management team member chooses to approve or reject the resident’s registration.</td></tr>
<tr><td colspan="3">Sub-flows:</td></tr>
<tr><td colspan="3">Alternate/Exceptional Flows:</td></tr>
</table>

Table 4.4 Use Case of Track Resident's Information

<table>
<tr><td colspan="2">Use Case Name: Track resident’s information</td><td>ID: 3</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: Management teams</td><td colspan="2">Use Case Type: Detail, Real</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
Management teams - Management teams can track their resident’s information.</td></tr>
<tr><td colspan="4">Brief Description: This use case describes how management teams view, or remove the residents.</td></tr>
<tr><td colspan="4">Trigger: When a management team member wants to view or remove the resident from the application after successfully logins into his or her account.</td></tr>
<tr><td colspan="4">Relationships:

Association : Management teams
Include : -
Extend : -
Generalization : -</td></tr>
<tr><td colspan="4">Normal Flow of Events:

1. The management team member enters the homepage of the web application.
2. The management team member selects the resident's list on the navigation bar.
3. The management team member clicks on the existing residents tab.
4. The web application shows all the existing residents.
5. The management team member views all existing residents’ information.
6. The management team member chooses to remove the residents from the list.
7.1 The S-1 suubflow is performed.</td></tr>
</table>

Sub-flows: S-1: 1. If the management team member chooses to remove the resident from the list, he / she clicks on the “remove” button and the resident will be removed from the database. 2. If the management team member chooses not to remove the resident from the list, no removal operation will be performed.
Alternate/Exceptional Flows:

Table 4.5 Use Case of Manage Administrators

<table>
<tr><td colspan="2">Use Case Name: Manage administrators</td><td>ID: 4</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: Management teams</td><td colspan="2">Use Case Type: Detail, Real</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
Management teams - Management teams can manage administrators in the application.</td></tr>
<tr><td colspan="4">Brief Description: This use case describes how management teams view, add, remove the administrator from administrator’s list.</td></tr>
<tr><td colspan="4">Trigger: When a management team member wants to view or add or remove the resident from the application after successfully login his or her account.</td></tr>
<tr><td colspan="4">Relationships:

Association : Management teams
Include : -
Extend : -
Generalization : -</td></tr>
<tr><td colspan="4">Normal Flow of Events:

1. The management team member enters the homepage of the web application.
2. The management team member selects the administrator on the navigation bar.
3. The web application shows all the administrator’s information.
4. The management team member is able to view all administrators’ information.
5. Only the management team member who has the highest authority can choose the operation (add or remove administrator).
5.1 The S-1 subflow is performed.</td></tr>
</table>

Sub-flows:

S-1:
1. If the add administrator operation is chosen, the S-1.1 flow is performed.
2. If the remove administrator operation is chosen, the S-1.2 flow is performed.

S-1.1: Create new administrator
1. The management team member with higher authority clicks on the “add admin” button.
2. The management team member with higher authority enters name, identification number, phone number, email address, and role.
3. The web application saves the administrator’s information and automatically assigns the identification card as password.

S-1.2: Remove administrator
1. The management team member with higher authority clicks on the “remove” button on the specific administrator.
2. The web application will prompt to confirm the removal action.
3. If yes is selected, the administrator will be removed from the web application. If no is selected, no removal operation will be performed.

Alternate/Exceptional Flows:

Table 4.6 Use Case of Manage Security Guards

<table>
<tr><td colspan="2">Use Case Name: Manage security guards</td><td>ID: 5</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: Management teams</td><td colspan="2">Use Case Type: Detail, Real</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
Management teams - Management teams can manage security guards' information.</td></tr>
<tr><td colspan="4">Brief Description: This use case describes how management teams can add or view or remove security guards.</td></tr>
<tr><td colspan="4">Trigger: When a management team member wants to add or view or remove security guards after successfully logins his or her account.</td></tr>
<tr><td colspan="4">Relationships:

Association : Management teams
Include : -
Extend : -
Generalization : -</td></tr>
<tr><td colspan="4">Normal Flow of Events:

1. The management team member enters the homepage of the web application.
2. The management team member selects security guards on the navigation bar.
3. The web application shows all security guards' information.
4. The management team member chooses to add a new security guard or remove a security guard.
5.1The S-1 subflow is performed.</td></tr>
</table>

Sub-flows:

S-1:
1. If the add new security guard operation is chosen, S-1.1 flow is performed.
2. If the remove security guard operation is chosen, S-1.2 flow is performed.

S-1.1: Add security guard
1. The management team member clicks on the “add security guard” button.
2. The management team member enters the security guard information including name, identification number, email and phone number.
3. The management team member clicks on the “add” button.
4. The web application saves the security guard’s information and automatically assigns the identification card as password.

S-1.2: Remove security guard
1. The management team member clicks on the “remove” button on the specific security guard.
2. The web application will prompt to confirm the removal action.
3. If yes is selected, the security guard will be removed from the web application. If no is selected, no removal operation will be performed.

Alternate/Exceptional Flows:

Table 4.7 Use Case Of Modify User Profile

Use Case Name: Modify User Profile	ID: 6	Importance Level: High
Primary Actor: Management teams	Use Case Type: Detail, Real	
Stakeholders and Interests: Management teams - Management teams can modify their user profile.		
Brief Description: This use case describes how management teams can modify their own user profile.		
Trigger: When a management team member wants to modify their user profile after successfully logins his or her account.		
Relationships: Association : Management teams Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The management team member enters the homepage of the web application. 2. The management team member clicks on the profile photo. 3. The web application shows the administrator's information. 4. The management team member updates his or her information or password. 5. The management team member clicks on the "update" button to update his or her information.		
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.8 Use Case of Track Visitor's Recors

Use Case Name: Track visitor's records	ID: 7	Importance Level: High
Primary Actor: Management teams	Use Case Type: Detail, Real	
Stakeholders and Interests: Management teams - Management teams can track visitor's information.		
Brief Description: This use case describes how management teams view or search the visitors.		
Trigger: When a management team member wants to view or search the visitors from the application after successfully logins his or her account.		
Relationships: Association : Management teams Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The management team member enters the homepage of the web application. 2. The management team member selects visitors on the navigation bar. 3. The web application shows all visitors' lists. 4. The management team clicks on the "total visitors" tab to view all the visitors. 5. The web application shows the total visitors' list. 6. The management team member clicks on the "search" button. 7. The management team member inserts visitors name or unit id or visitation id to search the visitor. 8. The web application shows the searched result.		
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.9 Use Case of Manage Announcements

Use Case Name: Manage Announcements	ID: 8	Importance Level: High
Primary Actor: Management teams	Use Case Type: Detail, Real	
Stakeholders and Interests: Management teams - Management teams can manage the announcements.		
Brief Description: This use case describes how management teams add, view, remove the announcements.		
Trigger: When a management team member wants to add, view, remove the announcements from the application after successfully logins his or her account.		
Relationships: Association : Management teams Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The management team member enters the homepage of the web application. 2. The management team member selects announcements on the navigation bar. 3. The web application shows all the announcements. 4. The management team member views all the announcements. 5. The management team member chooses to add an announcement or edit draft announcement or remove announcement operation. 6.1The S-1 subflow is performed.		

Sub-flows:

S-1 :
1. If add post operation is chosen, S-1.1 flow is performed.
2. If edit draft post operation is chosen, S-1.2 flow is performed.
3. If remove post operation is chosen, S-1.3 flow is performed.

S-1.1: Add announcement
1. The management team member clicks on the “add announcement” button.
2. The management team member enters the announcements information including title, descriptions, images.
3. The management team member clicks on “save as draft” button or “publish” button.
4. The web application saves the announcement as draft if the “save as draft” button is selected. The web application saves and publishes the announcement when the “publish” button is selected.

S-1.2: Edit draft announcement
1. The management team member clicks on the “edit” button on a draft announcement.
2. The web application shows all the details of the selected draft announcement.
2. The management team member enters the updated announcements information including title, descriptions, images.
3. The management team member clicks on the “save” button or “publish" button.
4. The web application saves the announcement as draft if the “save as draft” button is selected. The web application saves and publish the announcement when the “publish” button is selected.

S-1.3: Remove announcement
1. The management team member clicks on the “remove” button on a particular announcement.
2. The web application will prompt a message to confirm the management team member wants to remove the announcement.
3. If yes is selected, the web application will remove the announcement. If not is selected, no removal operation will be performed.

Alternate/Exceptional Flows:

Table 4.10 Use Case of Manage Feedback

Use Case Name: Manage Feedback	ID: 9	Importance Level: High
Primary Actor: Management teams	Use Case Type: Detail, Real	
Stakeholders and Interests: Management teams - Management teams can manage residents' feedback.		
Brief Description: This use case describes how management teams view, reply the feedback from residents.		
Trigger: When a management team member wants to view, reply residents's feedback from the application after successfully logins his or her account.		
Relationships: Association : Management teams Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The management team member enters the homepage of the web application. 2. The management team member selects feedback on the navigation bar. 3. The web application shows all the new feedback from residents by default. 4. If the management team member clicks on replied feedback, the S-1 subflow is performed. 5. The management team member views all the not replied feedback. 6. The management team member clicks on the "view" button to view the specific feedback. 7. The web application shows the feedback's title and description. 8. The management team member clicks on the "reply" button to reply to the feedback. 9. The management team member enters the reply message. 10. The management team member clicks on the "send reply" button to send the reply to the resident.		

Sub-flows: S-1: 1. The web application shows all the replied feedback. 2. The management team member clicks on the specific feedback to view the feedback and reply 3. The web application shows the feedback's title and description and reply from the management team.
Alternate/Exceptional Flows:

Mobile Application

Table 4.11 Use Case of Submit Registration Form

Use Case Name: Submit Registration Form	ID: 10	Importance Level: High
Primary Actors: Residents	Use Case Type: Detail, Real	
Stakeholders and Interests: Residents - Residents can submit their registration form.		
Brief Description: This use case describes how residents submit their registration form.		
Trigger: When a resident wants to submit registration form.		
Relationships: Association : Residents Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The resident opens the mobile application. 2. The resident clicks on the" register as a resident" button. 3. The mobile application shows the registration form to the residents. 4. The resident fills in his or her user and unit information including name, identification number, email address, phone number, password , unit id, address, supporting documents of the unit (example: electricity bill, water bill, etc.) , and car plates number. 5. The resident clicks on the "submit" button to submit the form and wait for the approval from the management teams.		
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.12 Use Case of Login Account

<table>
<tr><td colspan="2">Use Case Name: Login Account</td><td>ID: 11</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actors: Residents, Security guards</td><td colspan="2">Use Case Type: Detail, Real</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
Residents - Residents can login their accounts.
Security guards – Security guards can login their accounts.</td></tr>
<tr><td colspan="4">Brief Description: This use case describes how residents and security guards log into their accounts before entering the application.</td></tr>
<tr><td colspan="4">Trigger: When a resident or security guards wants to logins their account to access functions in the application</td></tr>
<tr><td colspan="4">Relationships:

Association : Residents
Include : -
Extend : Submit Registration Form
Generalization : -</td></tr>
<tr><td colspan="4">Normal Flow of Events:

1. The resident or security guard opens the mobile application.
2. The resident or security guard clicks on the “login” button.
2.1 The S-1 subflow is performed
3. The resident or security guard logs in their accounts with email and password.
3.1 The password for the security guard is his or her identification number.
4. The mobile application verifies the email and password.</td></tr>
<tr><td colspan="4">Sub-flows:

S-1:
1. If the user is resident, the resident clicks on login as the “resident” button.
2. If the user is security guard, the security guard clicks on “login as security guard” button.</td></tr>
</table>

Alternate/Exceptional Flows: 2. The resident without an account needs to submit their registration form to management teams. The security guards without an account need to inform management team member to create an account.

Table 4.13 Use Case of Modify User Profile

<table>
<tr><td>Use Case Name: Modify User Profile</td><td>ID: 12</td><td>Importance Level: High</td></tr>
<tr><td>Primary Actors: Residents</td><td colspan="2">Use Case Type: Detail, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
Residents - Residents can modify their user profile.</td></tr>
<tr><td colspan="3">Brief Description: This use case describes how residents modify their user profile.</td></tr>
<tr><td colspan="3">Trigger: When a resident wants to modify their user profile after successfully logins his or her account.</td></tr>
<tr><td colspan="3">Relationships:

Association : Residents
Include : -
Extend : -
Generalization : -</td></tr>
<tr><td colspan="3">Normal Flow of Events:

1. The mobile application shows the home page.
2. The resident clicks on the profile picture on the home page.
3. The mobile application shows the user information includes name, identification number, email address, phone number, password, car plates.
4. The resident edit the information that he or she wants to modify.
5. The resident clicks on the “update” button.
6. The mobile application updates his or her user profile.</td></tr>
<tr><td colspan="3">Sub-flows:</td></tr>
<tr><td colspan="3">Alternate/Exceptional Flows:</td></tr>
</table>

Table 4.14 Use Case of Register Visitors

<table>
<tr><td>Use Case Name: Register visitors</td><td>ID: 13</td><td>Importance Level: High</td></tr>
<tr><td>Primary Actors: Residents, Security guards</td><td colspan="2">Use Case Type: Detail, Real</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
Residents - Residents can pre-register for their visitors.
Security guards – Security guards can register for ad-hoc visitors.</td></tr>
<tr><td colspan="3">Brief Description: This use case describes how residents pre-register for their visitors and how security guards register for ad-hoc visitors.</td></tr>
<tr><td colspan="3">Trigger: When a visitor wants to visit the resident, the resident needs to pre-register the visitor for his/her access to the condominium/serviced apartment/guarded area. When a adhoc visitor wants to access to the condominium/serviced apartment/guarded area without any pre-registration by the resident, the security guards need to register the visitor from their end.</td></tr>
<tr><td colspan="3">Relationships:

Association : Residents, Security guards
Include : -
Extend : -
Generalization : -</td></tr>
<tr><td colspan="3">Normal Flow of Events:

1. The mobile application shows the home page.
2. The user clicks on register visitor button.
3. The user enters visitor’s information.
3.1 The S-1 subflow is performed
including name, identification number, phone number, email address, date and time visited and car plate.
4. The user clicks on the “add visitor” button.
5. The mobile application saves the visitor and visitation information.</td></tr>
<tr><td colspan="3">Sub-flows:

S-1:
1. If the user is residents, the visitor’s information to be fill including name, identification number, phone number, email address, date and time visited and car plate.
2. If the user is residents, the visitor’s information to be fill including name, identification number, phone number, email address, car plate and unit id.</td></tr>
</table>

Alternate/Exceptional Flows:

Table 4.15 Use Case of Manage Registered Visitations

Use Case Name: Manage registered visitations	ID: 14	Importance Level: High
Primary Actors: Residents	Use Case Type: Detail, Real	
Stakeholders and Interests: Residents – Residents can manage their registered visitations.		
Brief Description: This use case describes how residents view, or cancel the registered visitation.		
Trigger: When a resident wants to view or cancel the visitation being registered after successfully logins his or her account.		
Relationships: Association : Residents Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The mobile application shows the home page. 2. The resident clicks on the “view all visitation” button. 3. The mobile application shows all the visitations including upcoming visitation and the completed visitations. 4. The resident chooses to view specific visitation or cancel specific visitation. 5.1 The S-1 subflow is performed.		
Sub-flows: S-1: 1. If view specific visitation is chosen, S-1.1 flow is performed. 2. If cancel specific visitation is chosen, S-1.2 flow is performed.		

S-1.1: View specific visitation. 1. The resident clicks on the “view” button on the specific visitation. 2. The mobile application shows the details of the visitation. S-1.2: Cancel specific visitation. 1. The resident clicks on the “cancel” button on the specific visitation. 2. The mobile application will prompt the resident to confirm his or her removal action. 3. If yes is selected, the mobile application will remove the specific visitation. If no is selected, no removal operation will be performed.
Alternate/Exceptional Flows:

Table 4.16 Use Case of Check in Visitations

Use Case Name: Check in visitations	ID: 15	Importance Level: High
Primary Actors: Visitors	Use Case Type: Detail, Real	
Stakeholders and Interests: Visitors – Visitors can check in their visitation through the visitation id and unit id given by the residents.		
Brief Description: This use case describes how visitors check in their visitations.		
Trigger: When a visitor arrives at the condominium/serviced apartments/guarded area and wants to verify his or her visitations.		
Relationships: Association : Visitors Include : Verify Visitation Extend : - Generalization : -		
Normal Flow of Events: 1. The visitor opens the application. 2. The visitor clicks on the “use as visitor” button. 3. The visitor enters the visitation id and unit id provided by the residents. 4. The visitor shows the qr code to the security guard. 5. The security guard verifies the qr code. 6. The mobile application prompts the visitation is recorded.		
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.17 Use Case of Verify Visitations

Use Case Name: Verify visitations	ID: 16	Importance Level: High
Primary Actors: Security guards	Use Case Type: Detail, Real	
Stakeholders and Interests: Security guards – Security guards can verify the visitor's check-in.		
Brief Description: This use case describes how security guards verify the visitor's check in when visitors arrive at the entry of the condominium/serviced apartments/guarded area.		
Trigger: When a visitor shows his or her visitation qr code to the security guards.		
Relationships: Association : Security guards Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The security guard opens the mobile application. 2. The security guard logins his or her account. 3. The security guard clicks on the "verify registered visitors" button. 4. The security guard scans the qr code shown by the visitor. 5. The mobile application shows the visitation id, unit id, and address of the registered visitors. 6. The security guard clicks on the "verify" button to verify the check in. 7. The mobile application will record the visitation time and the related security guard id in the visitation.		
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.18 Use Case of View Announcements

Use Case Name: View announcements	ID: 17	Importance Level: High
Primary Actors: Residents	Use Case Type: Detail, Real	
Stakeholders and Interests: Residents – Residents can receive and view the announcements publish by the management teams.		
Brief Description: This use case describes how residents view announcements publish by the management teams.		
Trigger: When the resident wants to check the latest information of his or her condominium/service apartment/guarded area after logins his or her account.		
Relationships: Association : Residents Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The mobile application shows the homepage. 2. The resident clicks on the “announcement” button. 3. The resident views all the announcements published by the management teams. 4. The resident clicks on a specific announcements to view the details of the announcements.		
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.19 Use Case of Manage Feedback

Use Case Name: Manage Feedback	ID: 18	Importance Level: High
Primary Actors: Residents	Use Case Type: Detail, Real	
Stakeholders and Interests: Residents – Residents can manage their feedback to the management teams.		
Brief Description: This use case describes how residents submit their feedback to the management teams, view the specific feedback's details.		
Trigger: When a resident has some opinion and wants to share or report to the management teams and when the resident wants to view the specific feedback details after logins his or her account.		
Relationships: Association : Residents Include : - Extend : - Generalization : -		
Normal Flow of Events: 1. The mobile application shows the homepage. 2. The resident clicks on the "feedback form" button. 3. The mobile application shows all feedback. 4. The resident chooses to add feedback or view feedback. 4.1 The S-1 subflow is performed.		
Sub-flows: S-1: 1. If add feedback operation is chosen, the S-1.1 alternate flow is performed. 2. If view feedback details operation is chosen, the S-1.2 alternate flow is performed. S-1.1: Add feedback 1. The resident selects the "add feedback" button. 2. The resident enters the feedback information including form title, descriptions and categories. 3. The resident clicks on the "submit" button.		

4. The mobile application saves the datetime and feedback into the database. S-1.2: View feedback details 1. The resident selects the specific feedback to view. 2. The mobile application shows the details of the feedback including, form title, descriptions, categories and reply from management teams.
Alternate/Exceptional Flows:

4.5 Interface Flow Diagram

The flow of web application and mobile application are shown in the interface flow diagrams.

4.5.1 Web Application

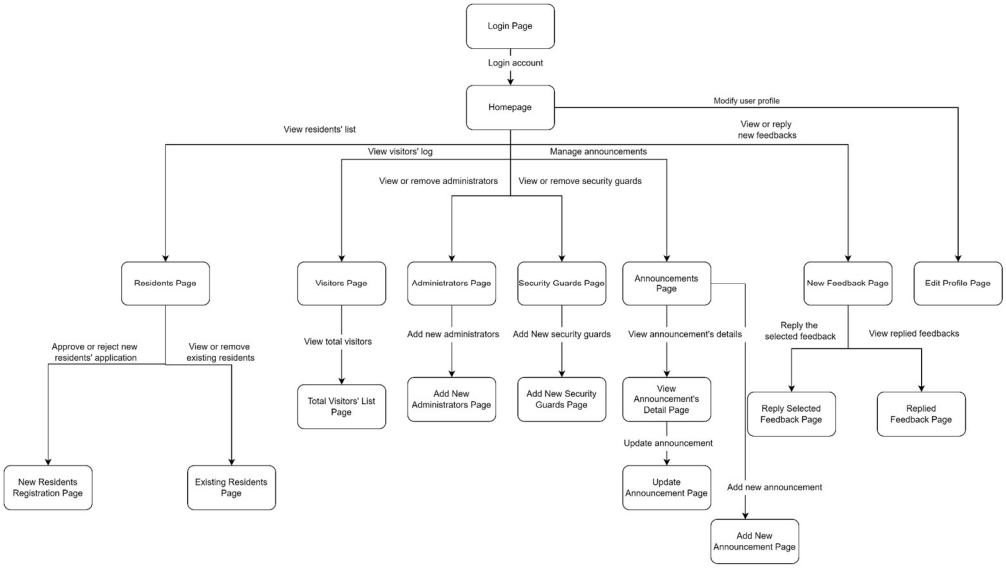


Figure 4.37 Interface Flow Diagram of Web Application

4.5.2 Mobile Application

1. Residents

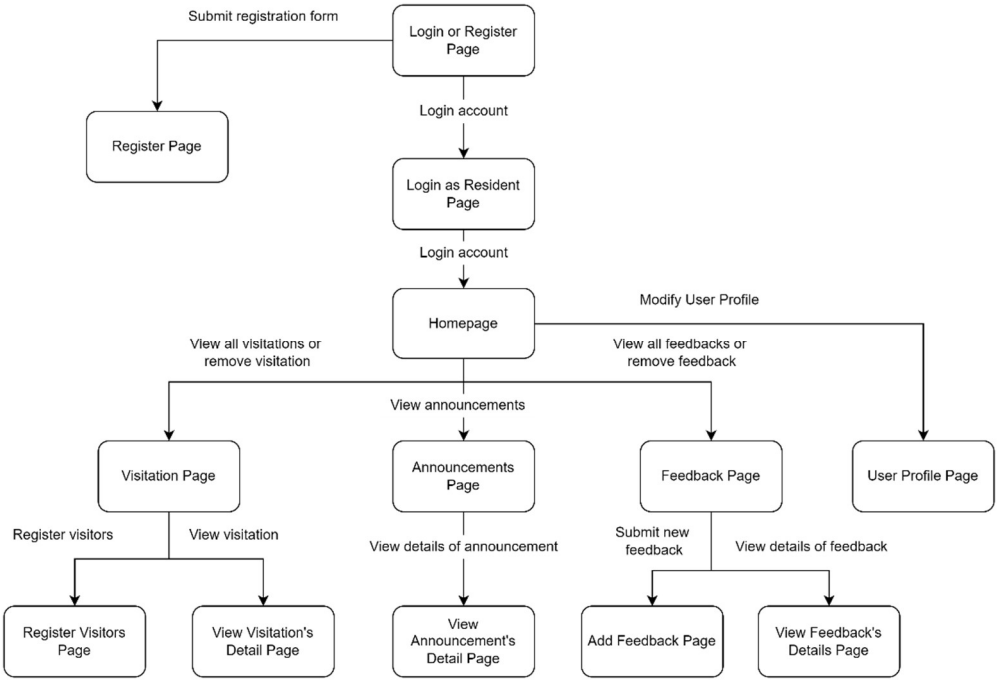


Figure 4.38 Interface Flow Diagram of Mobile Application (Resident)

2. Visitors

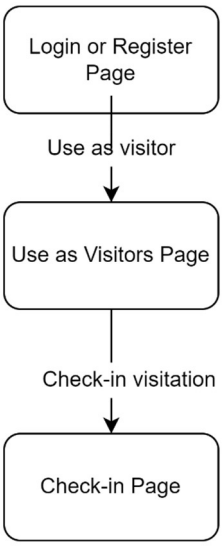


Figure 4.39 Interface Flow Diagram of Mobile Application (Visitor)

3. Security guards

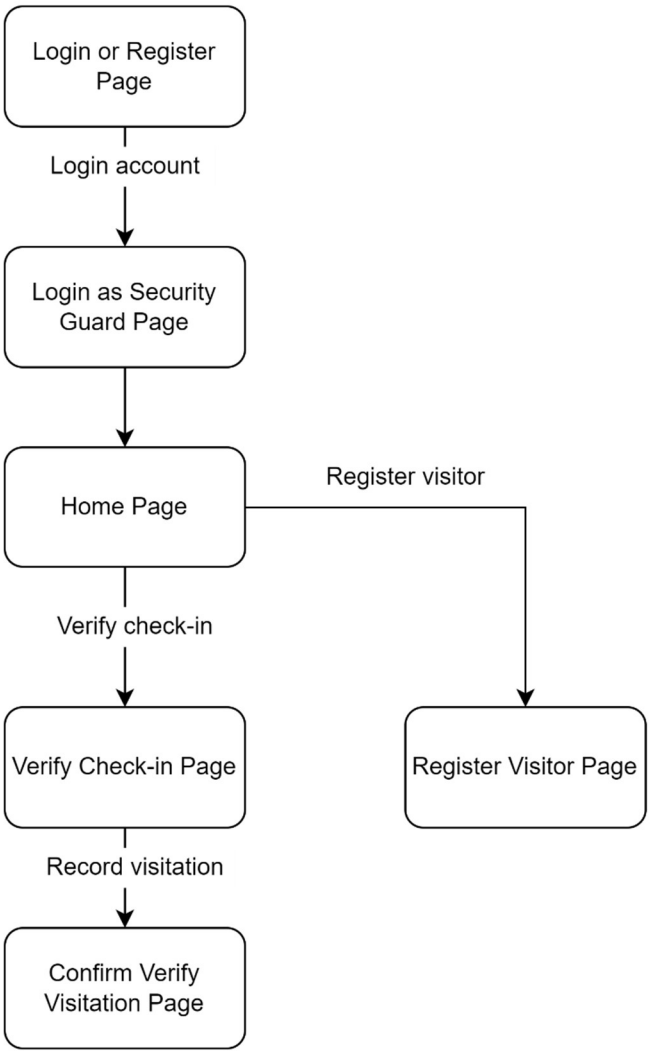


Figure 4.40 Interface Flow Diagram of Mobile Application (Security guard)

4.6 Prototype interface

4.6.1 Web Application

1. Login Page

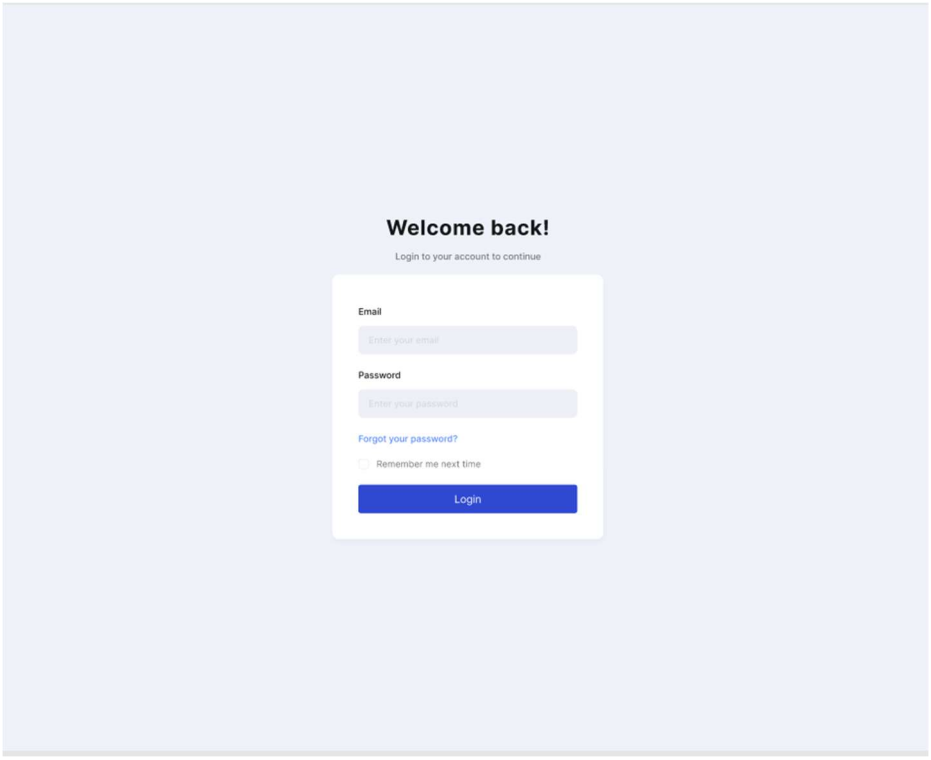


Figure 4.41 Login Page (Web)

2. Home page

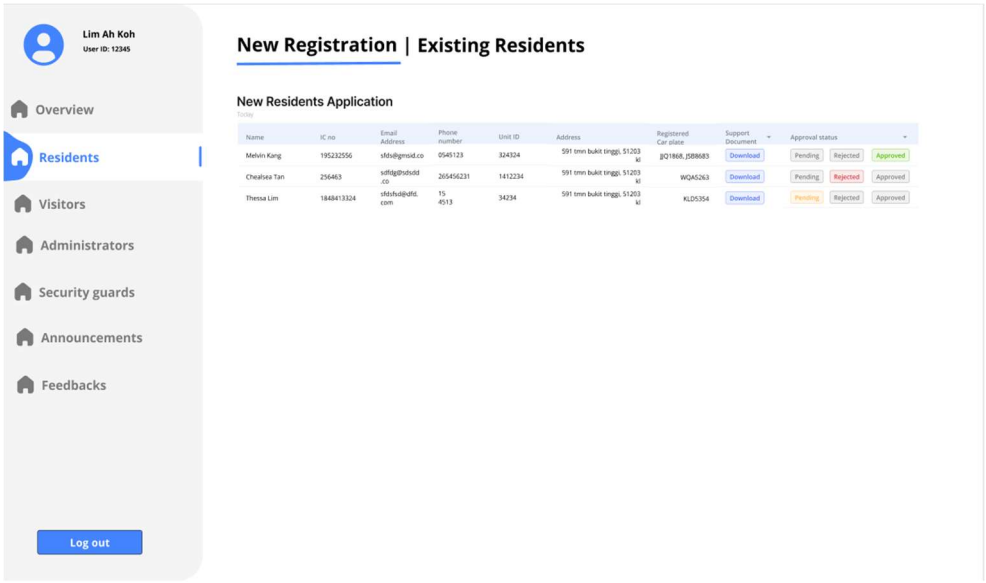


Figure 4.42 Home Page (Web)

3. New Resident Registration Page

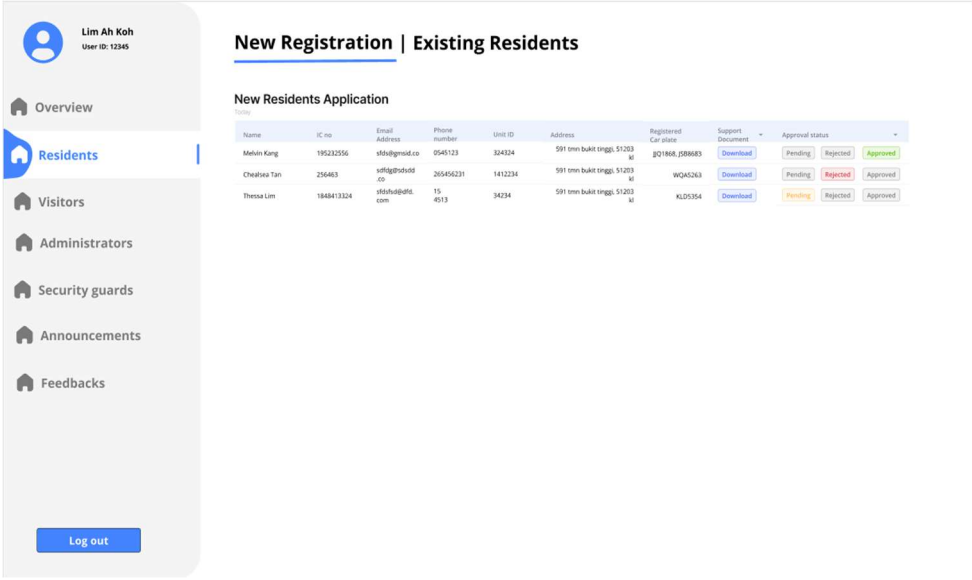


Figure 4.43 New Resident Registration Page

4. Existing Resident Page

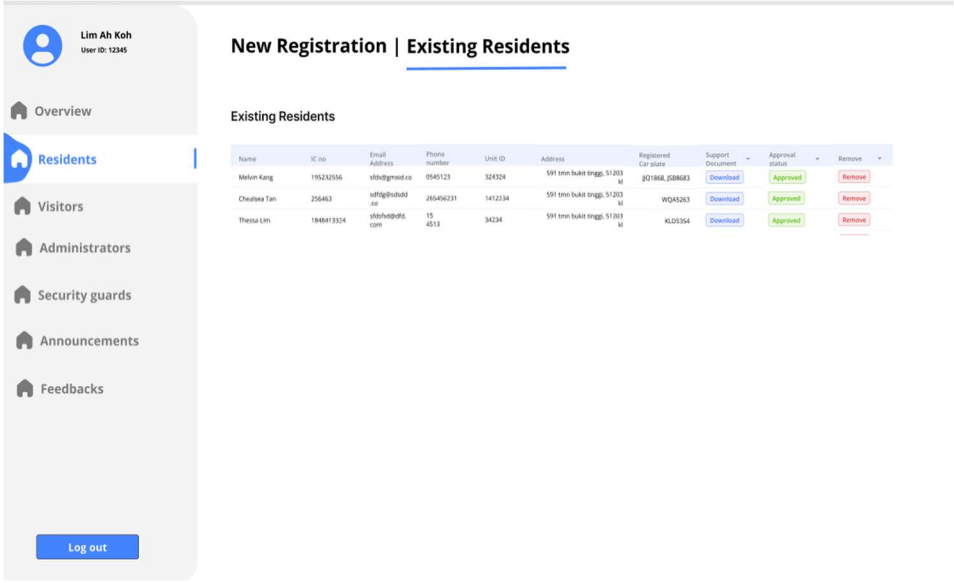


Figure 4.44 Existing Resident Page

5. Visitors Page

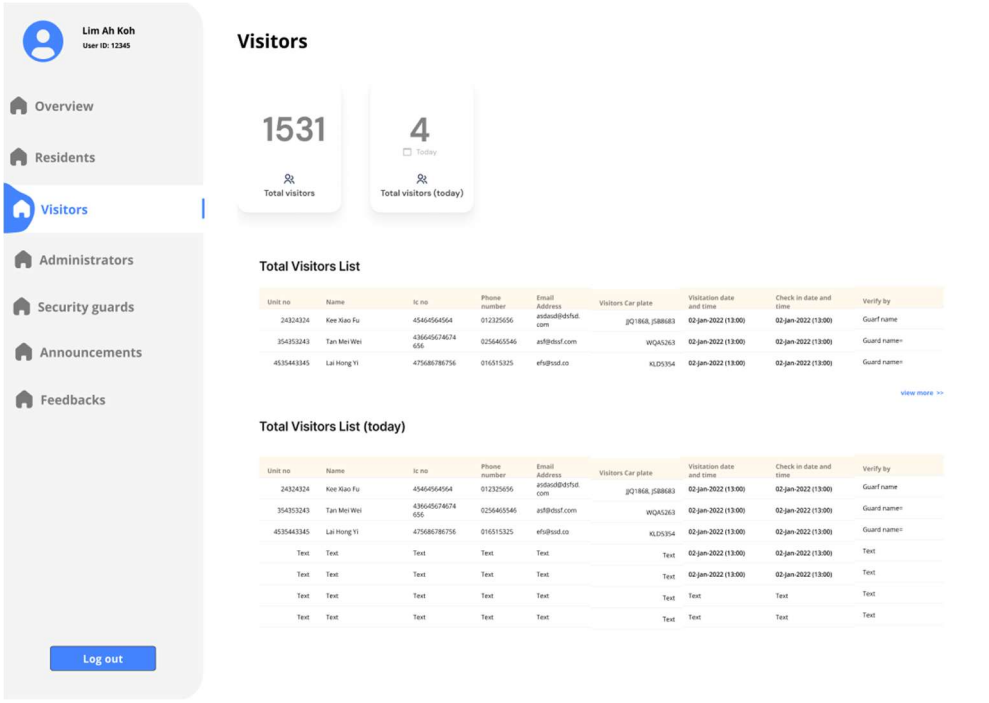


Figure 4.45 Visitor Page

6. Total Visitors List Page

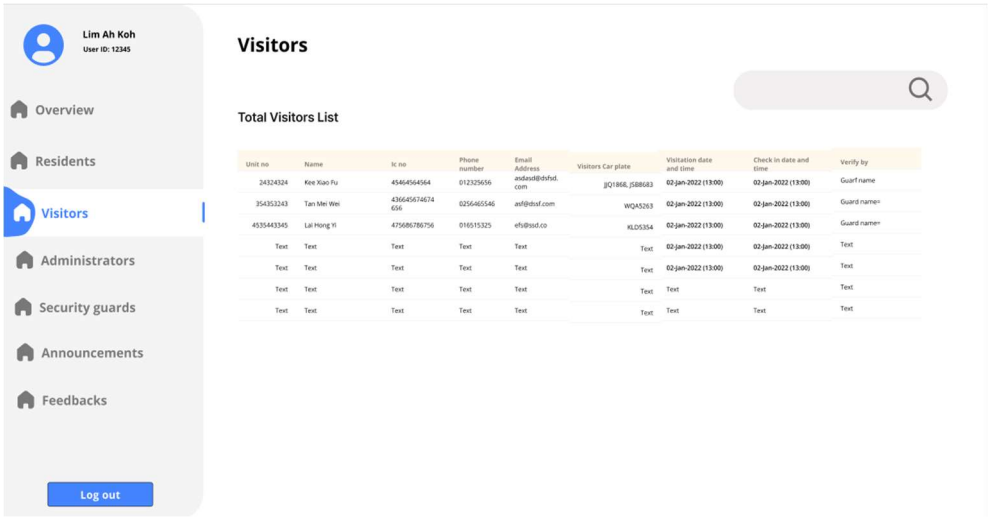


Figure 4.46 Total Visitor List Page

7. Administrators Page

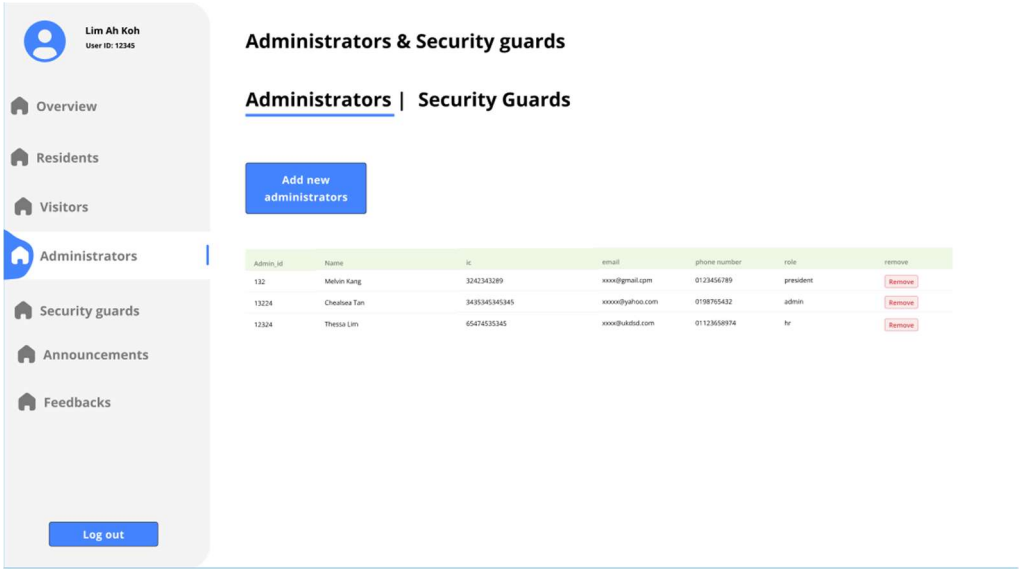


Figure 4.47 Administrators Page

8. Add New Administrator Page

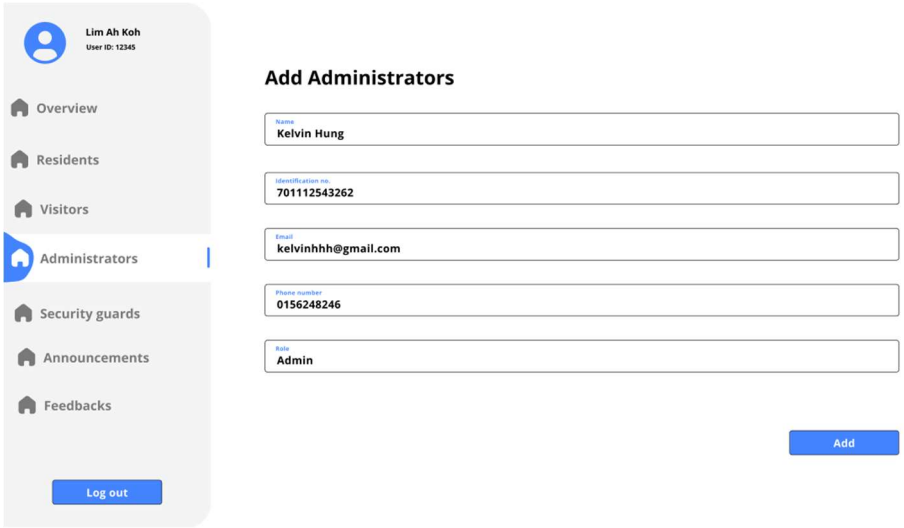


Figure 4.48 Add New Administrator Page

9. Security Guards Page

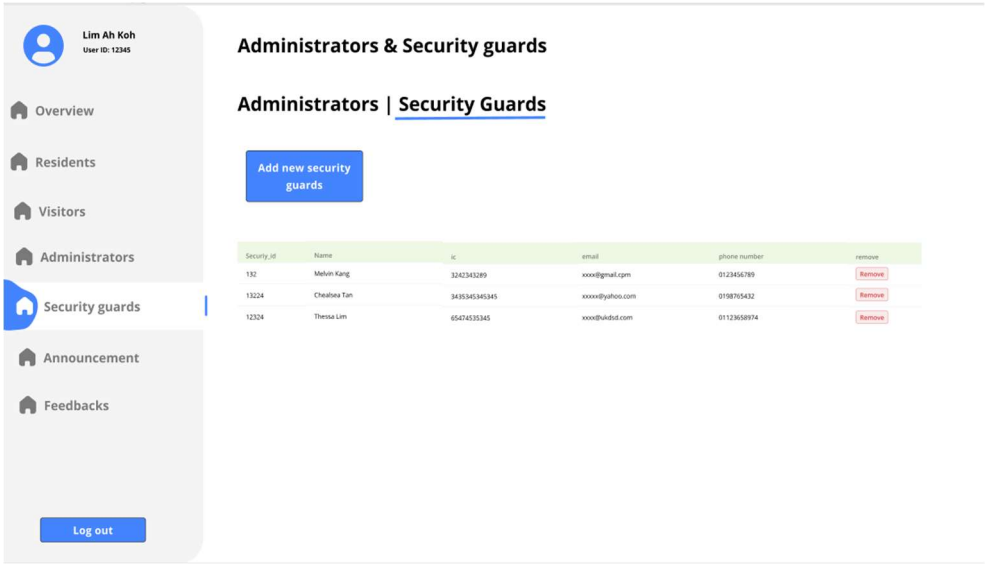


Figure 4.49 Security Guard Page

10. Add New Security Guard Page

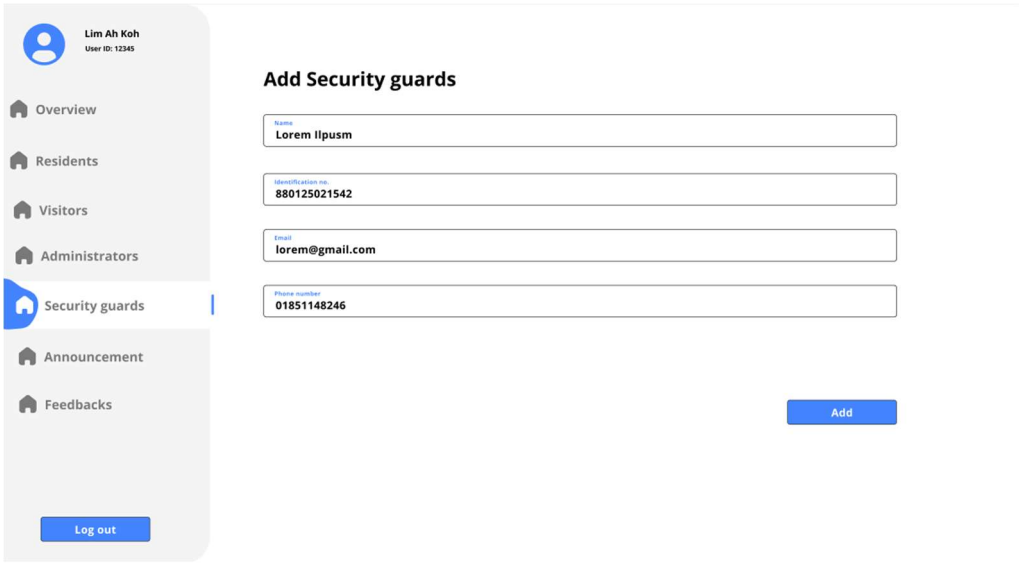


Figure 4.50 Add New Security Guard Page

11. Announcements Page

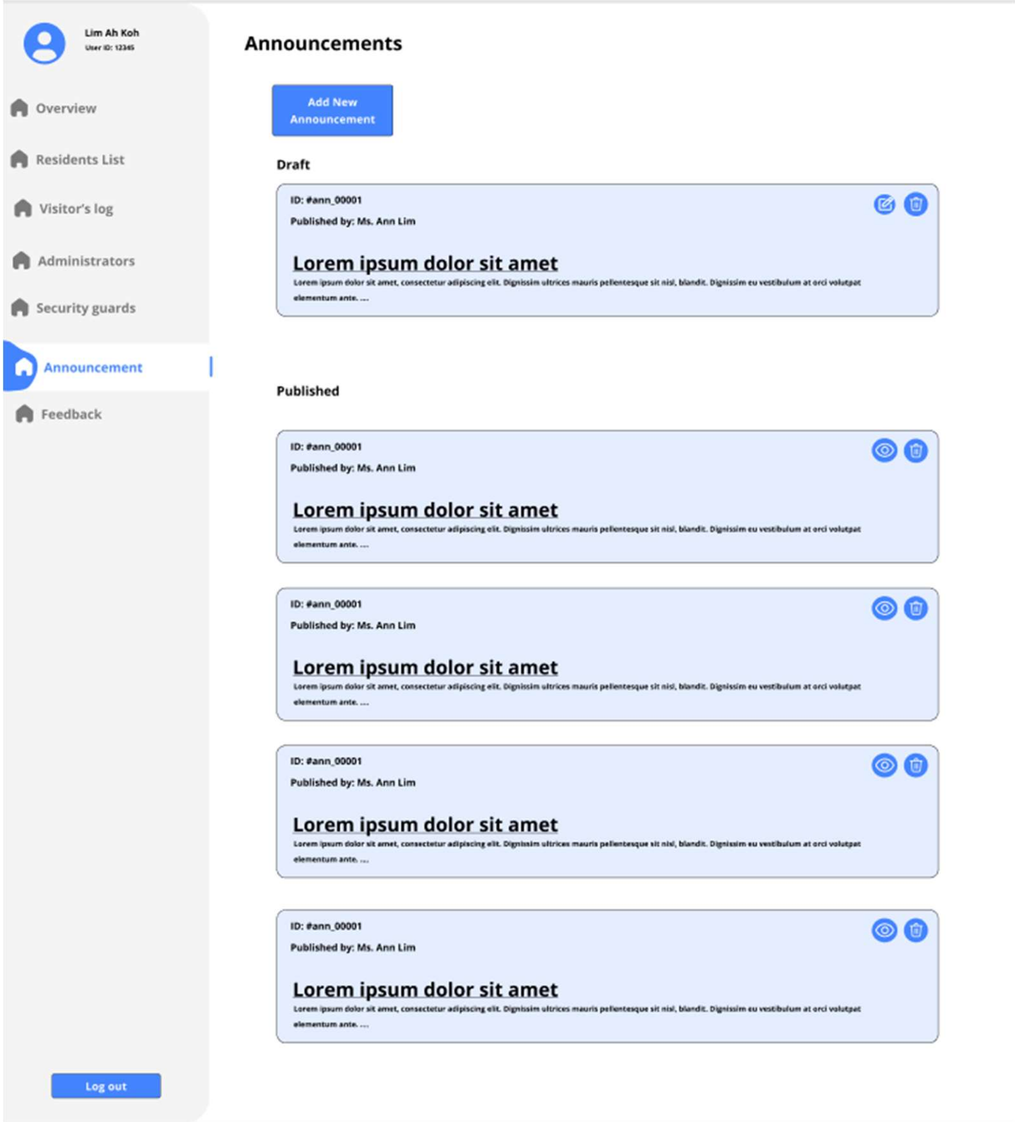


Figure 4.51 Announcements Page

12. View Announcements Detail Page

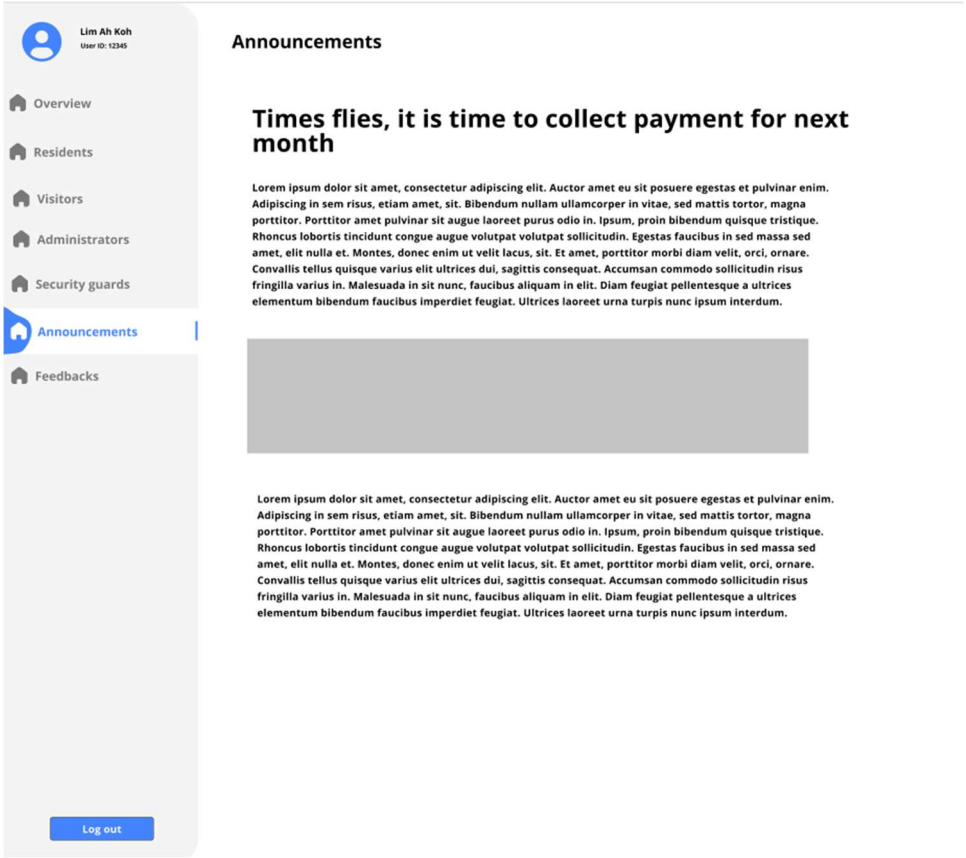


Figure 4.52 View Announcement Page

13. Update Announcement Page

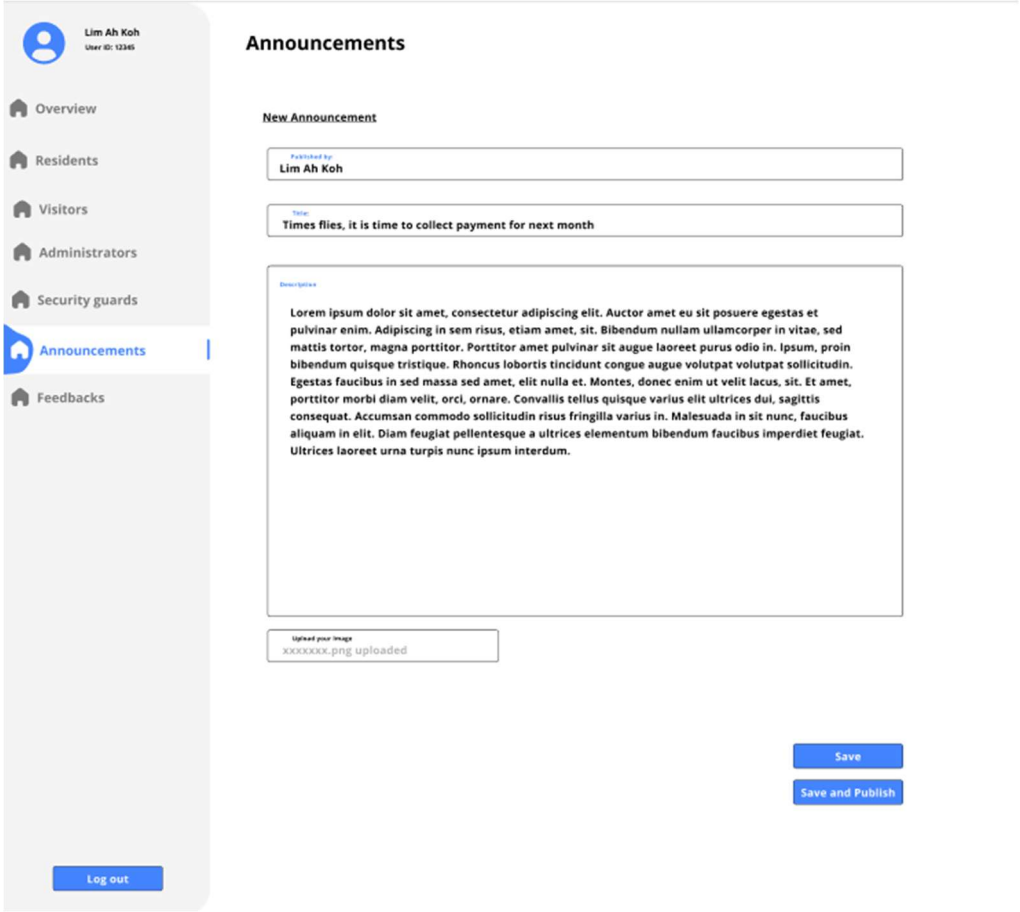


Figure 4.53 Update Announcement Page

14. Add New Announcement Page

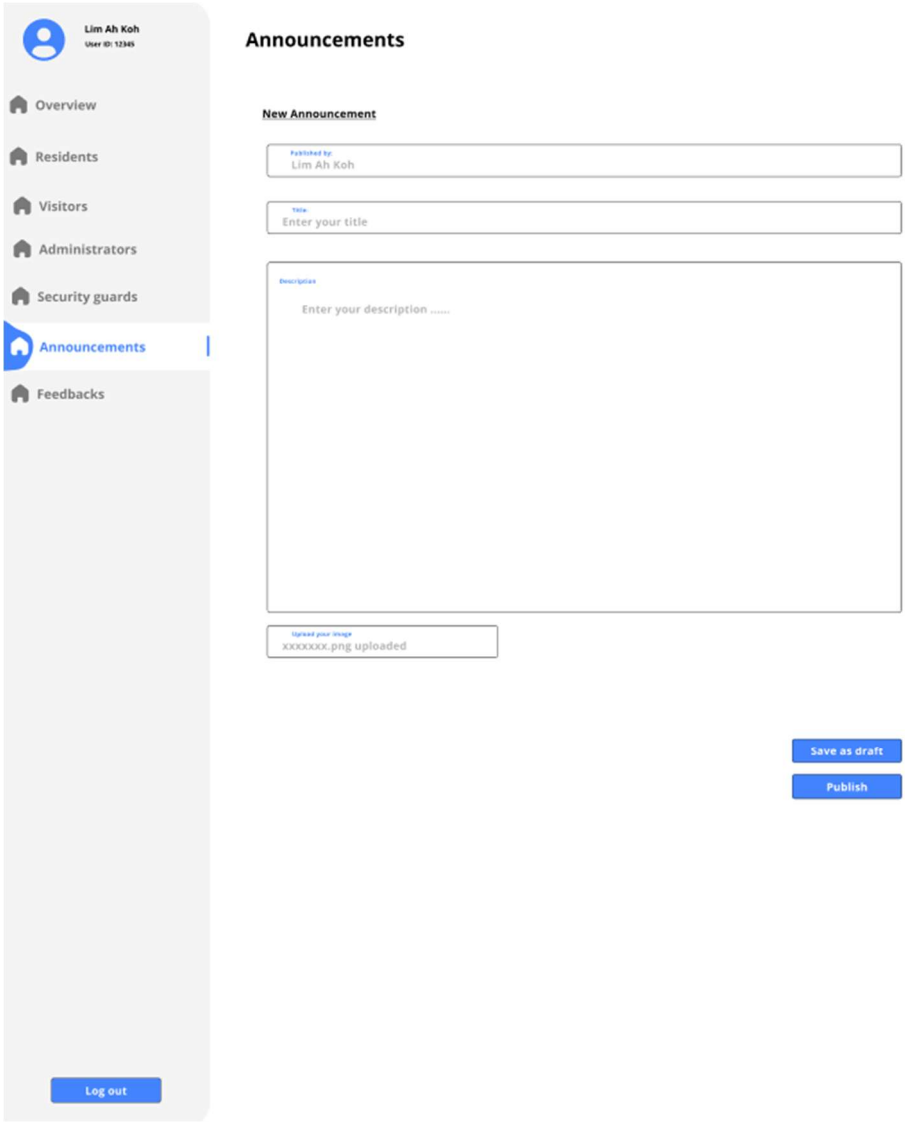


Figure 4.54 Add New Announcement Page

15. New Feedback Page

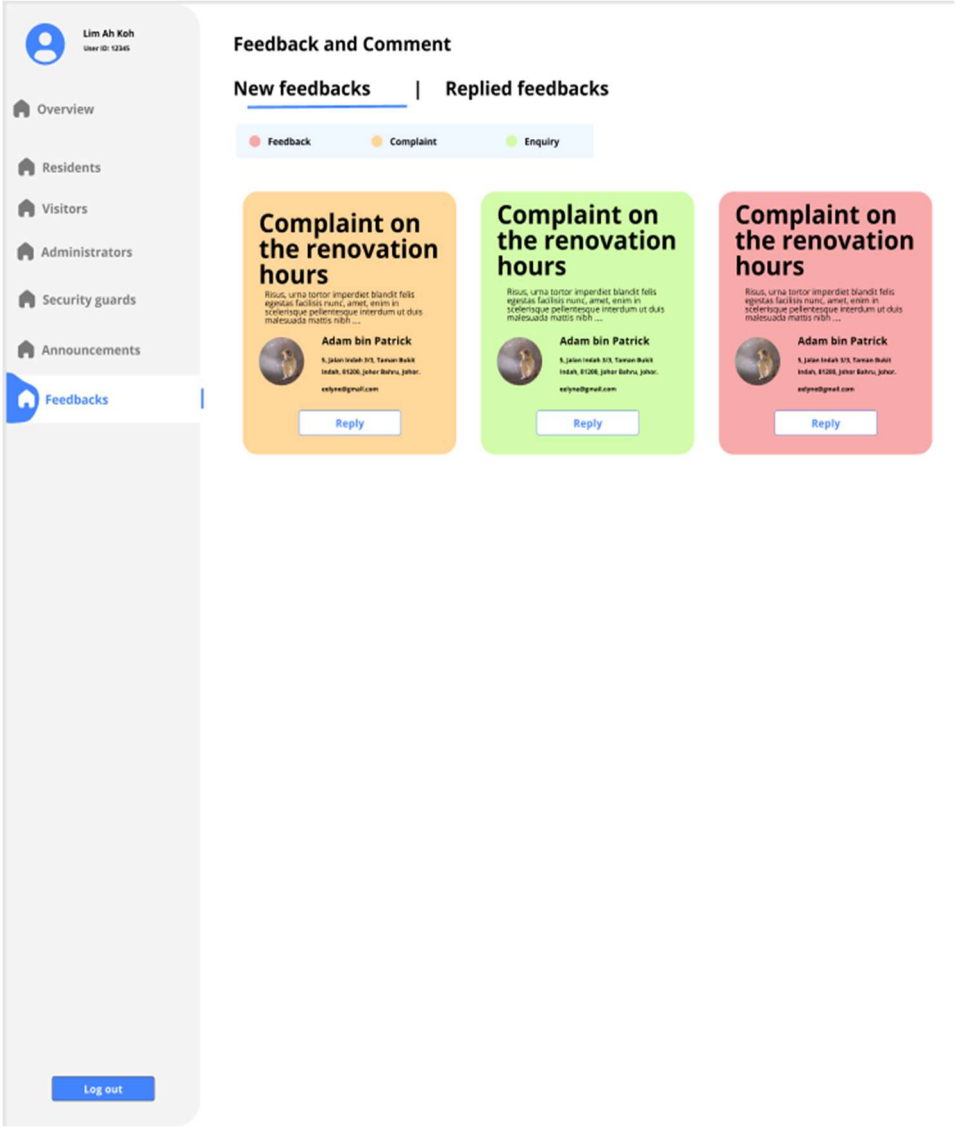


Figure 4.55 New Feedback Page

16. Replied Feedback Page

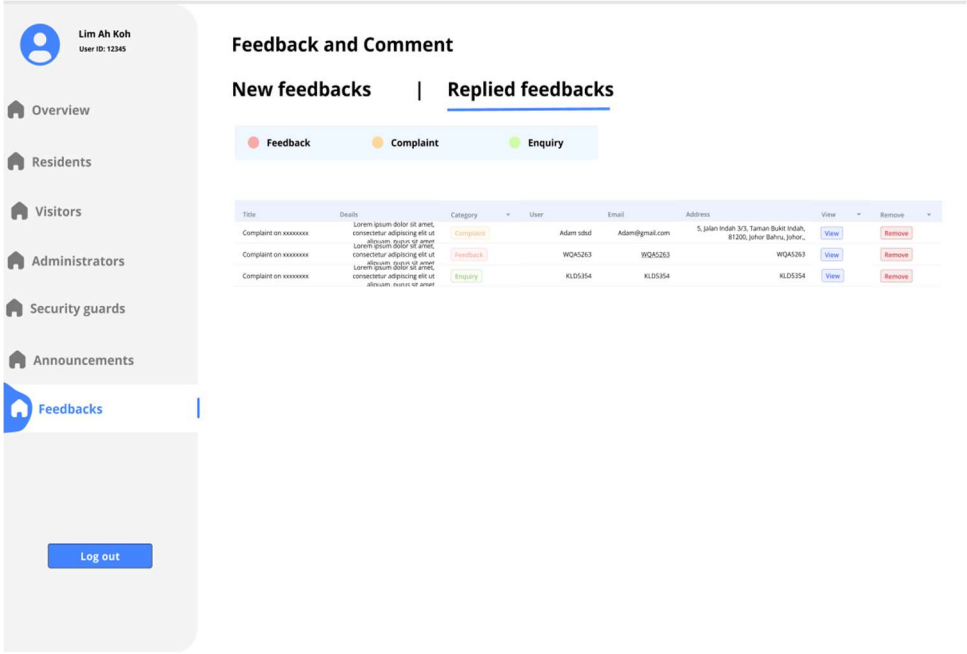


Figure 4.56 Replied Feedback Page

17. Reply Selected Feedback Page

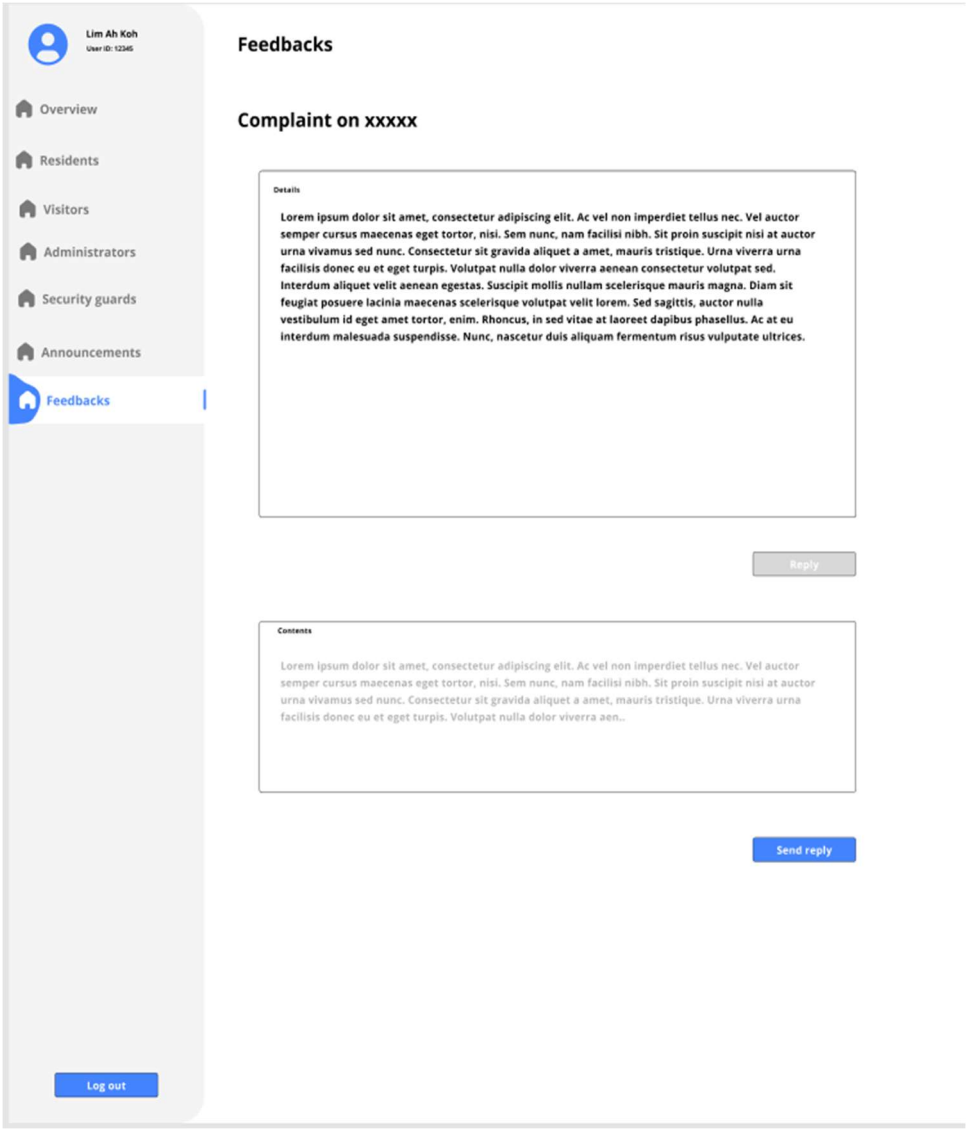


Figure 4.57 Reply Selected Feedback Page

18. Modify User Profile Page

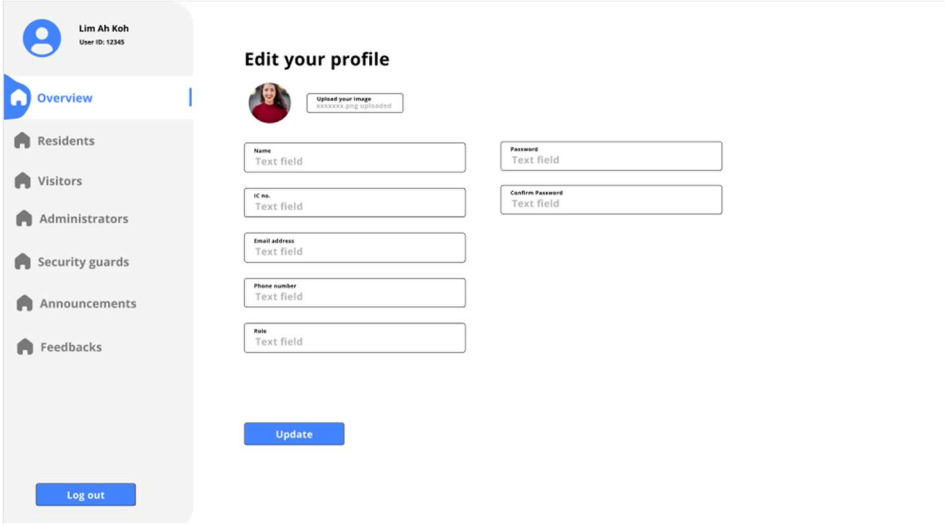


Figure 4.58 Modify User Profile Page

4.6.2 Mobile Application

1. Login or Register Page

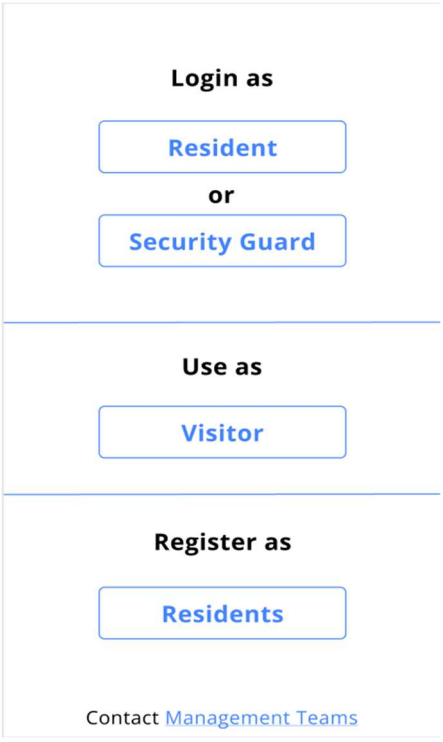


Figure 4.59 Login or Register Page

Residents

1. Register as Resident Page

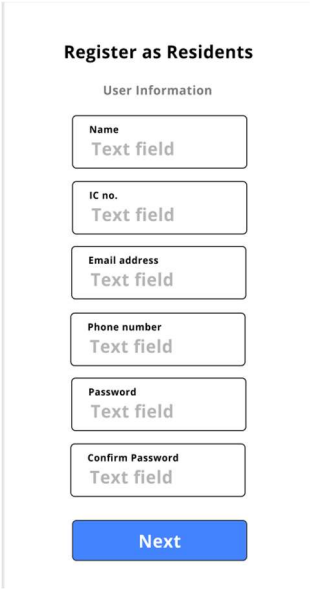


Figure 4.60 Register as Resident Page

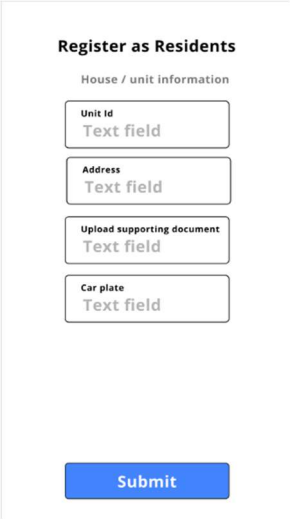


Figure 4.61 Register as Resident Page (cont.)

2. Login as Resident Page

Figure 4.62 Login as Resident Page

3. Home page

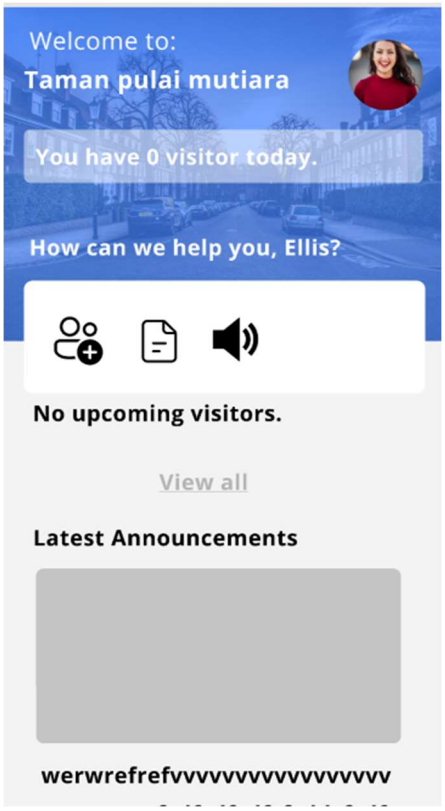


Figure 4.63 Home Page

4. Visitation Page

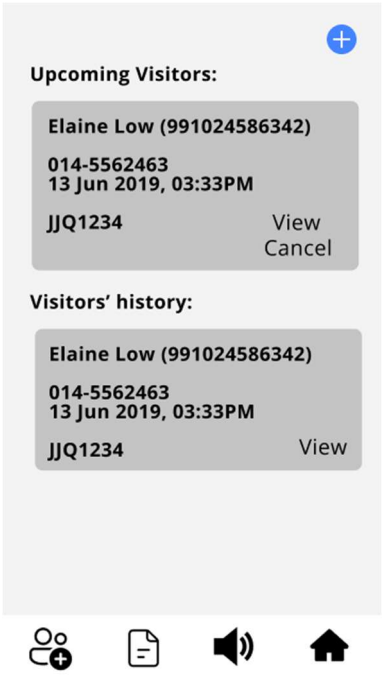


Figure 4.64 Visitation Page

5. Register Visitors Page

Figure 4.65 Register Visitors Page

6. View Visitation Details Page

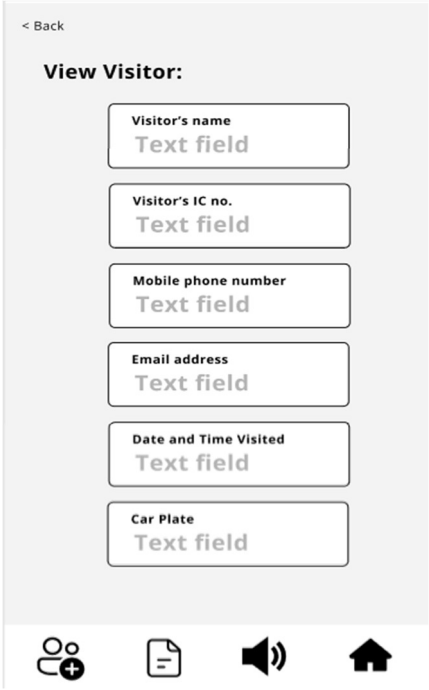


Figure 4.66 View Visitation Details Page

7. Announcements Page

Figure 4.67 Announcements Page

8. View Announcement Details Page

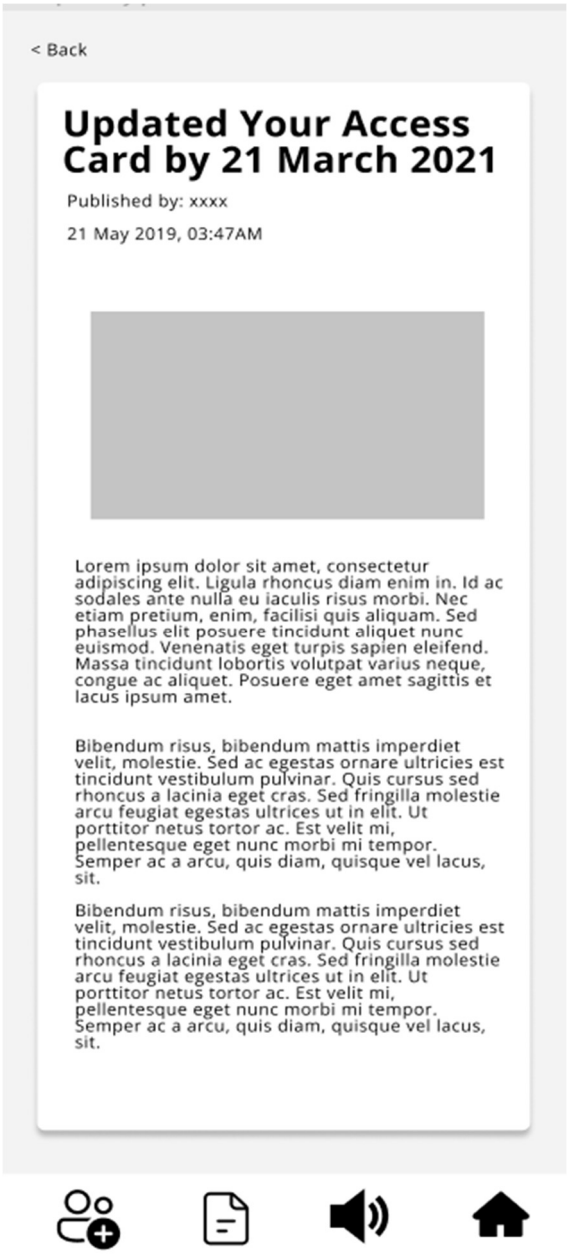


Figure 4.68 View Announcement Details Page

9. Feedback Page

Figure 4.69 Feedback Page

10. Add Feedback Page

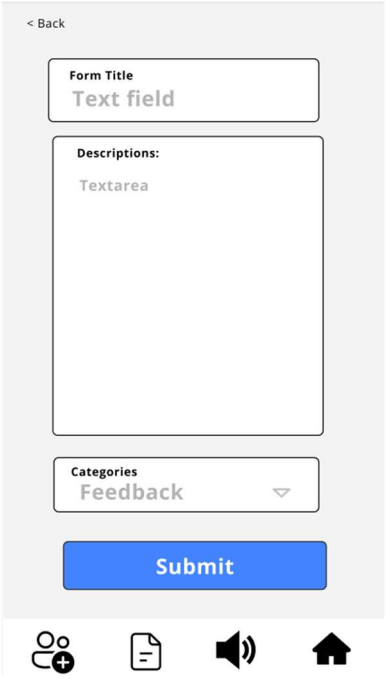


Figure 4.70 Add Feedback Page

11. View Feedback Detail Page

Figure 4.71 View Feedback Detail Page

12. Modify User Profile Page

Figure 4.72 Modify User Profile Page

Visitors

1. Use as Visitors Page

Figure 4.73 Use as Visitors Page

2. Check-in Page

Figure 4.74 Check-in Page

Security Guards

1. Login as Security Guard Page

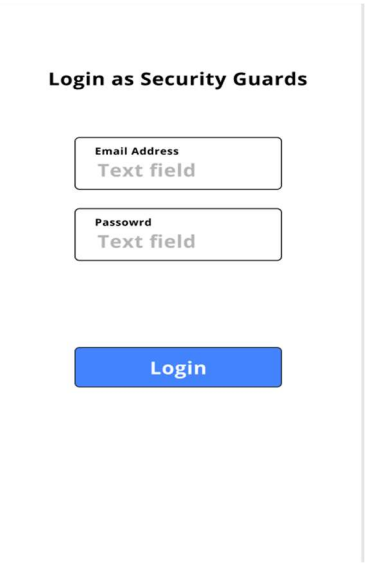


Figure 4.75 Login as Security Guard Page

2. Home Page

Figure 4.76 Home Page

3. Verify Check-in Page

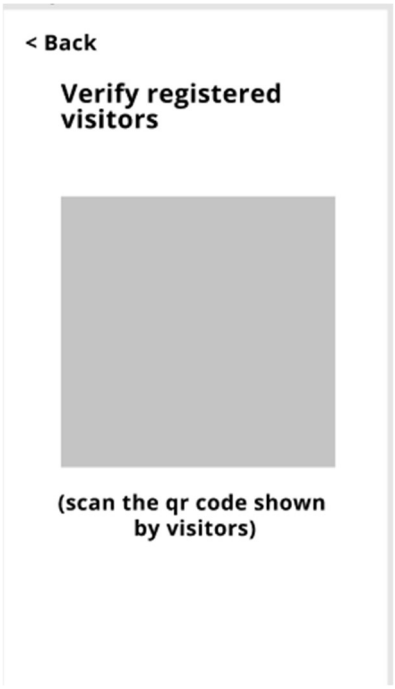


Figure 4.77 Verify Check-in Page

4. Confirm Verify Visitation Page

Figure 4.78 Confirm Verify Visitation Page

4.7 Summary

In general, this chaper discussed the analysis for questionnaires result. Besides, a total of 18 functional requirements and non-functional requirements were listed out and finalised. Next, the activities that the system is capable of performing were outlined in use case diagrams and use case descriptions.Interface flow diagram were also included to provide an overview of the systems. After the end of this phase, the project continued with the development and testing phase iteratively until the final system is completed.

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

In this chapter, the system architecture design used was discussed. Besides, data flow diagram and database schema were included. Furthermore, user interface designs were demonstrated to better visualise Resident and Visitor Management System.

5.2 System Architecture Design

In order to address the issues that developed as the software grows, a separate UI server architecture had been used. The API, for instance, might be accessible through mobile applications or third parties in addition to the web-based UI. Due to the separation of the UI and API, as seen in Figure 5.1, the mobile application will be able to access the API features.

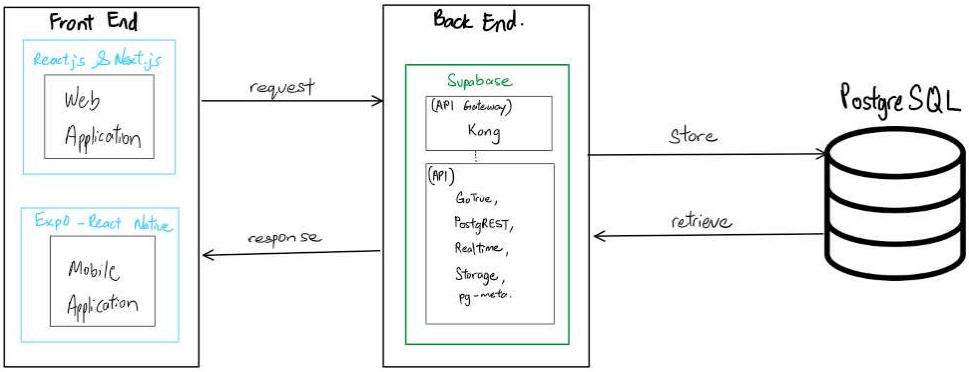


Figure 5.1 Separate UI Server Architecture

5.2.1 Front end architecture

React Components, which react to state changes, component interaction, and user input, form the foundation of both web and mobile applications. There is a render method included in each component that returns the HTML and JSX code needed to generate the component on a web page. React does not require the use of JSX, but doing so dramatically enhances the visual appeal of the UI component when using JavaScript.

Next.js which is production framework for React is used in this project. It is used on top of React, expanding its capabilitites and streamlining the development process. It also aims to reduce JavaScript fatigue by enabling developers to build web applications in a zero-config environment. At the heart of any Next.js application lies a collection of pages that provide content. Pages are loaded based on their path within the pages/ directory. Next.js handles routing URLs to their corresponding page content. It does not require explicitly create any sort of mapping between URLs and pages. Next.js takes care of rendering our components on the server for the initial page load. It also takes care of passing state to the browser so that when the user starts interacting with the page, the virtual DOM doesn't have to be completely recreated. API routes follow the same design principles as pages except they return JSON data that is consumed by pages that use the data to render content. Pages in Next.js applications are React components, meaning any React patterns or include any packages that allowed by React will work. One of the two pre-rendering techniques used by Next.js is known as Server-Side Rendering(SSR). The server loads user-specific data and prepares the page before sending it to the user's computer when a user requests a webpage. The page is then shown after the browser construct the contents. Server side rendering refers to the entire process of retrieving data from a database, creating an HTML page, and serving it to users. The HTML page is created by Next.js at build time, and it provides the pre-rendered page from the server to the browser with only a little amount of JavaScript code. When the page is loaded by the browser, the JavaScript code runs and creates a fully interactive page.

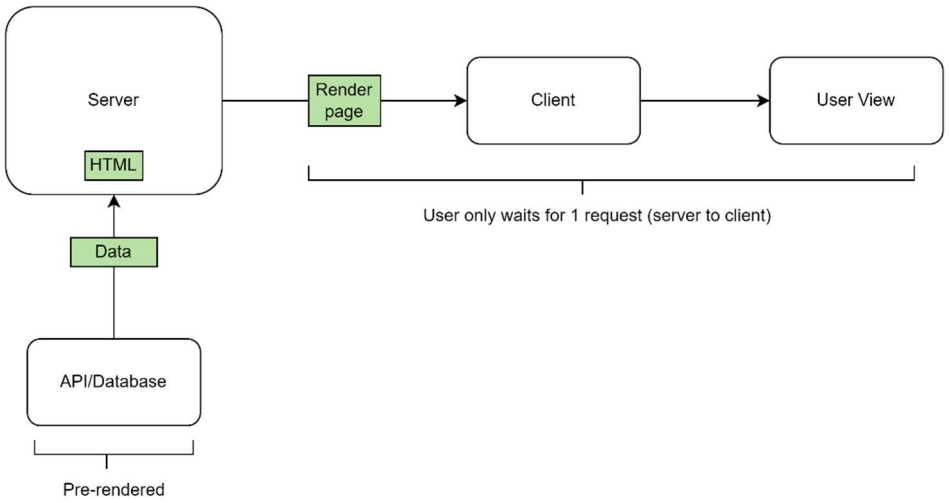


Figure 5.2 Server Side Rendering

React native is a cross-platform mobile framework that is used to build applications and webpages. JavaScript programmers can create mobile applications that can run on various platforms, including Windows, iOS and Android, thanks to React Native, which compiles to native app components. The native code is kept separate when a React Native application is launched, and the JavaScript code is bundled into a package called JS Bundle. The JavaScript thread runs the JS Bundle, and the native/UI thread runs the native modules and handles UI rendering. The communication between the native threads and javascript is enabled by a bridge, which sends data to the native threads after serializing as JSON. This bridge can only handle asynchronous communication.

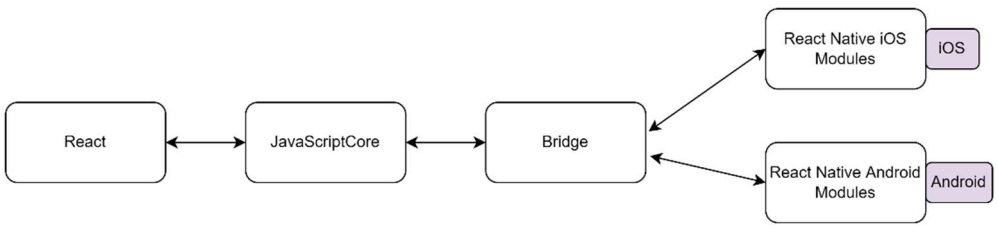


Figure 5.3 React Native Architecture

Expo is a set of tools created around React Native which is also used in this project. The mobile apps will automatically download and use the new JavaScript code on the following startup thanks to Expo's ability to post

JavaScript bundles to CDNs without asking a publication or store review. Any software that is compatible with Expo may be loaded using the generic Expo client. The native runtime used by all Expo apps (React Native + ExpoKit) is the same. The JavaScript code developers coded in the project is the only thing that differs. The Expo apps published to the app stores have the JavaScript bundle URL hardcoded in them. The Expo client is built in a particular approach with the purpose of letting developers to choose from which URL to load the Javascript where either by scanning a QR code or providing a URL. Additionally, Expo Client may load JavaScript bundles from localhost, simplifying the development process which without the use of Xcode or Android Studio and the process of getting the project to function is substantially quicker.

5.2.2 Back end architecture

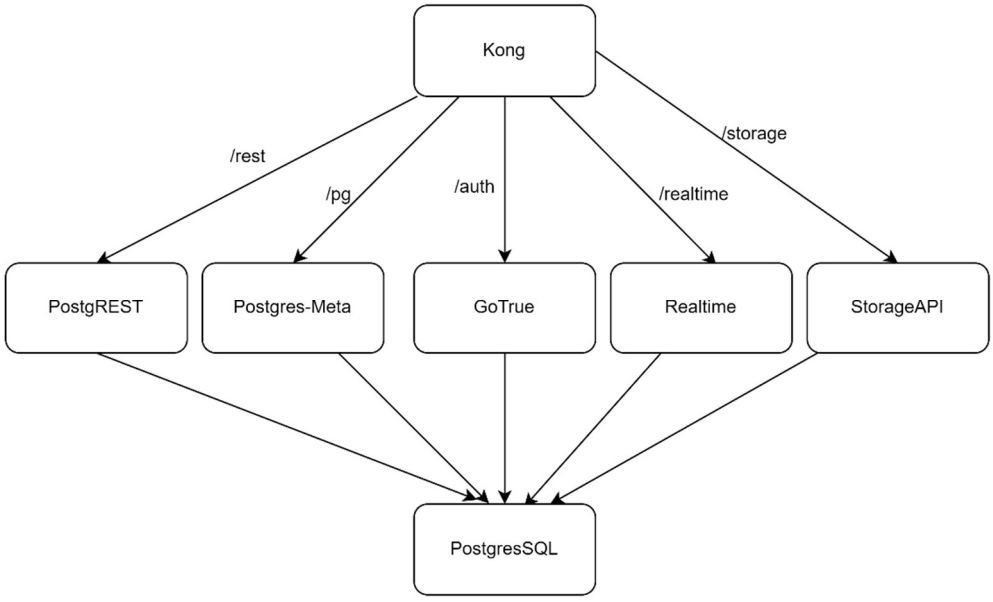


Figure 5.4 Supabase Architecture

Supabase is used as the backend and it is an app development platform built on top of PostgreSQL. It consists of the building blocks as shown in Figure 5.4. Kong serves as API gateway. GoTrue is used to manage users and issue authentication tokens. A RESTful API can be accessed from any generated Postgres database using PostgREST. WebSockets are made available through Realtime so they can listen to the target Postgres database. With Postgres and GoTrue, Storage-api functions as an S3-compatible object storage service. Fastify, a Node.js web framework, was used in its construction. In order to manage PostgreSQL databases, Postgres-meta offers a RESTful API that can be used to fetch tables, add roles, and execute queries. PostgreSQL provides persistence as the main part of Supabase, S3 storage provider for storing large files. The details of the implementation of the backend will be shown in Chapter 6.

5.3 Database architecture

5.3.1 Database Schema

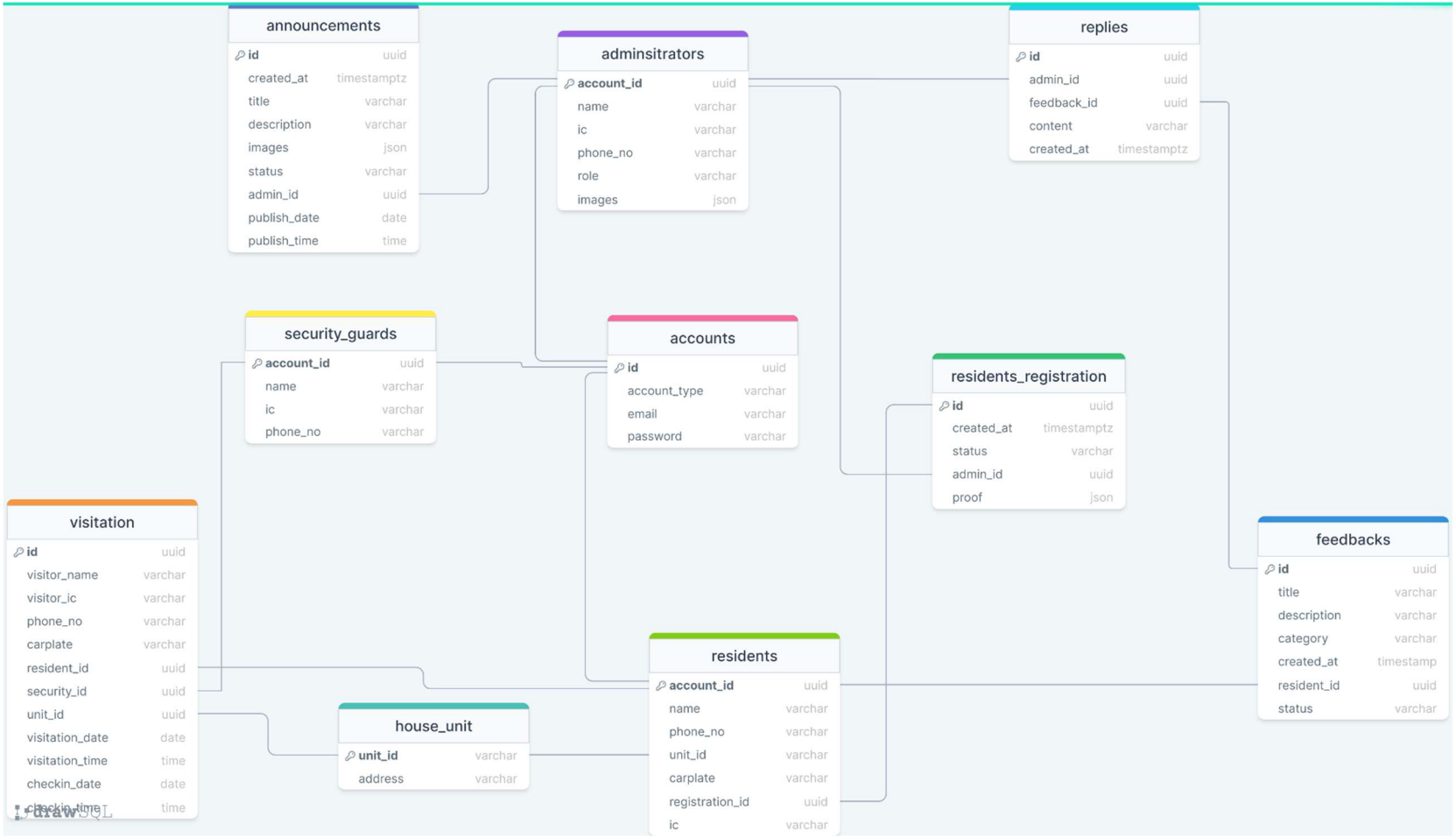


Figure 5.5 Database Schema

5.3.2 Table Description

Table 5.1 Table Description

Table Name	Description
Accounts	Contains the accounts details of every user which includes email address, password and account type.
Adminstrators	Contains the adminsitrators details of every administrator.
Announcements	Contains the details of announcement for evey announcement
Feedbacks	Contains the information of feedback for every feedback
House_unit	Contains the house unit information of each house unit
Replies	Contains the reply details of each feedback's reply
Residents	Contains the information of all residents
Residents_registration	Contains the residents' registration detail for every resident
Security_guards	Contains the security guard detail for each security guard
Visitation	Contains the information of all visitation.

5.3.3 Data Flow Diagram

5.3.3.1 Context Diagram

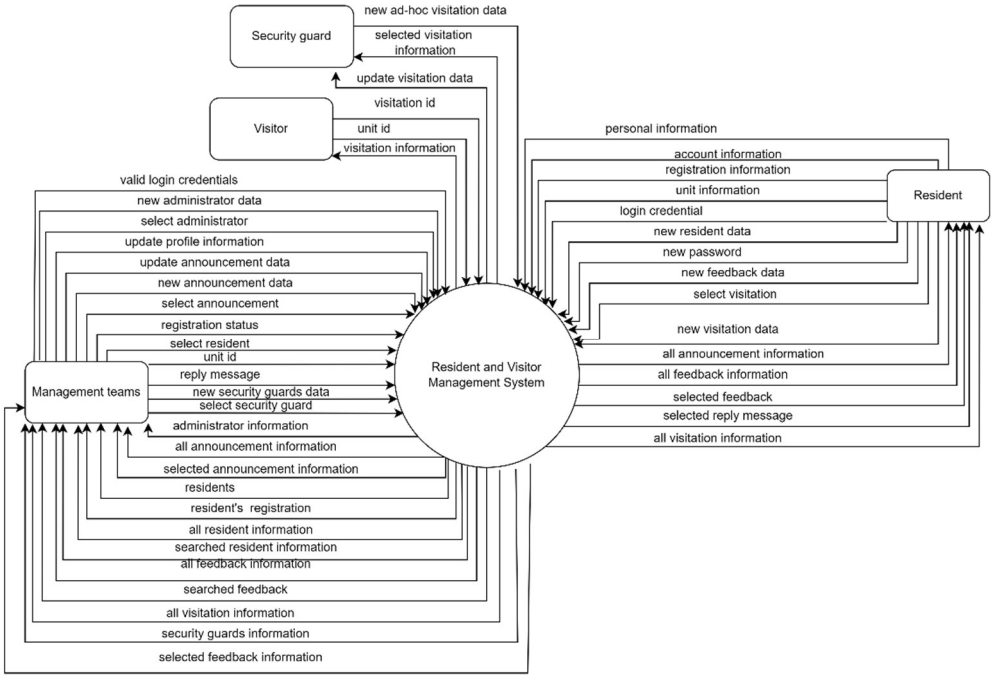


Figure 5.6 Context Diagram

5.3.3.2 Data Flow Diagram Level-0

Web application

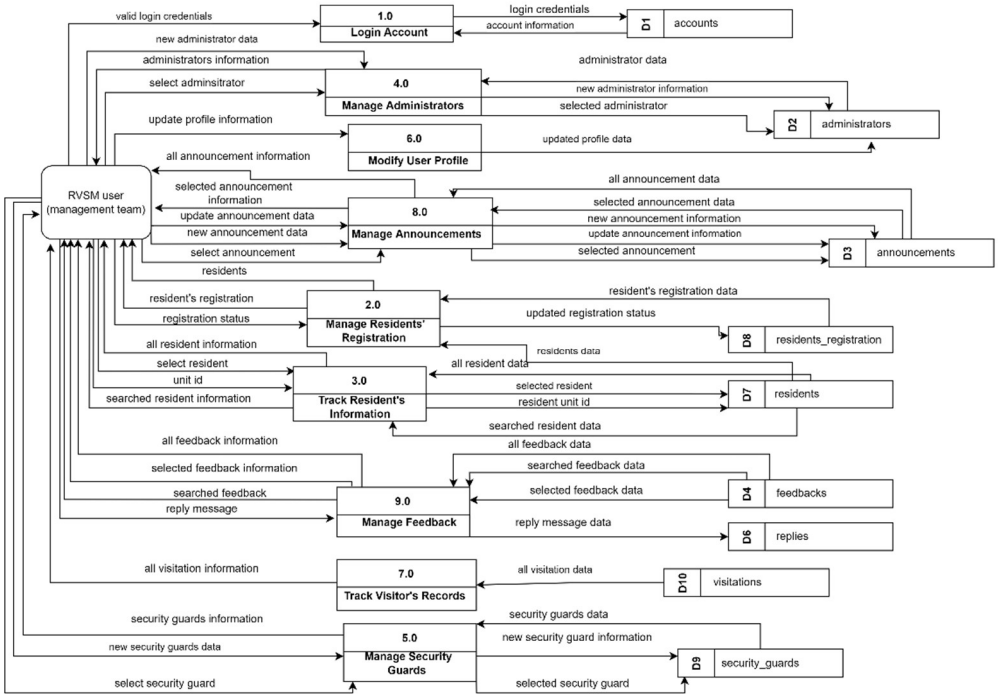


Figure 5.7 Data Flow Diagram Level-0 (Web application)

Mobile Application

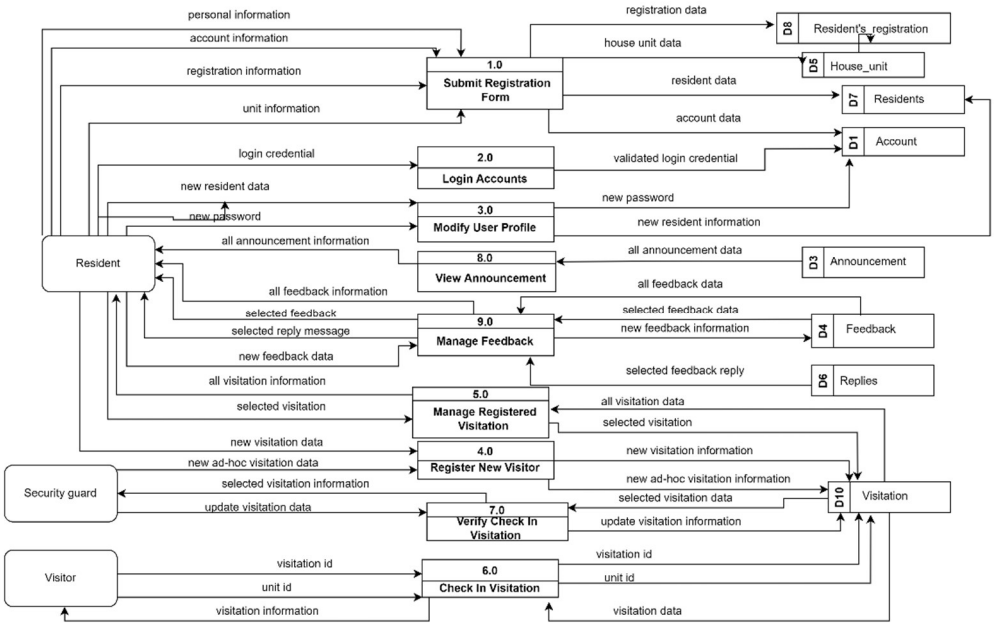


Figure 5.8 Data Flow Diagram Level-0 (Mobile Application)

5.3.3.3 Data Flow Diagram Level-1 (Web Application)

2.0 Manage Resident's Registration

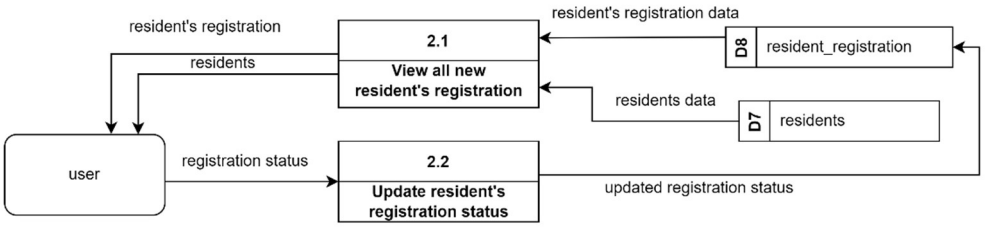


Figure 5.9 Data Flow Diagram Level-1 (2.0 Manage Resident's Registration)

3.0 Track Resident's information

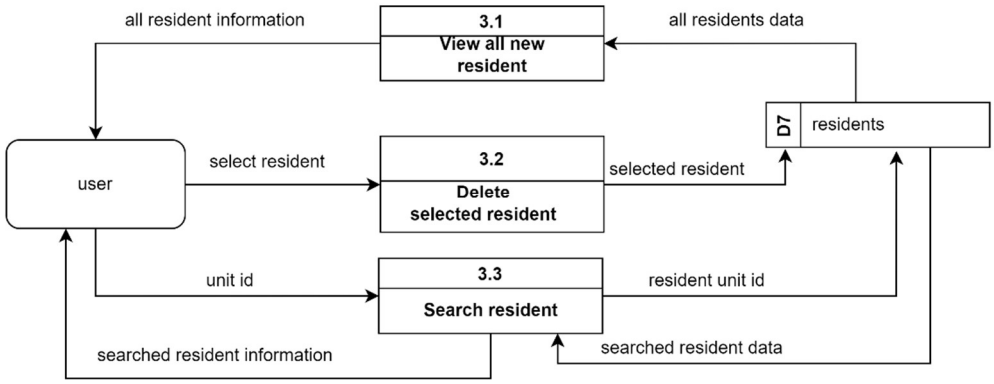


Figure 5.10 Data Flow Diagram Level-1 (3.0 Track Resident's information)

4.0 Manage Administrator

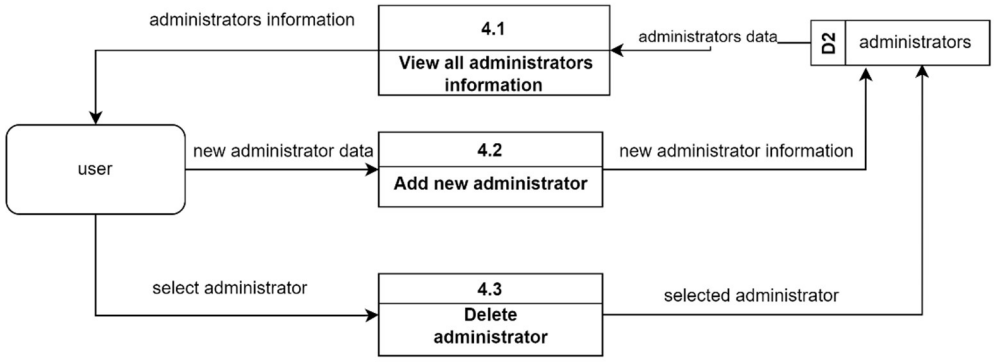


Figure 5.11 Data Flow Diagram Level-1 (4.0 Manage Administrator)

5.0 Manage Security Guard

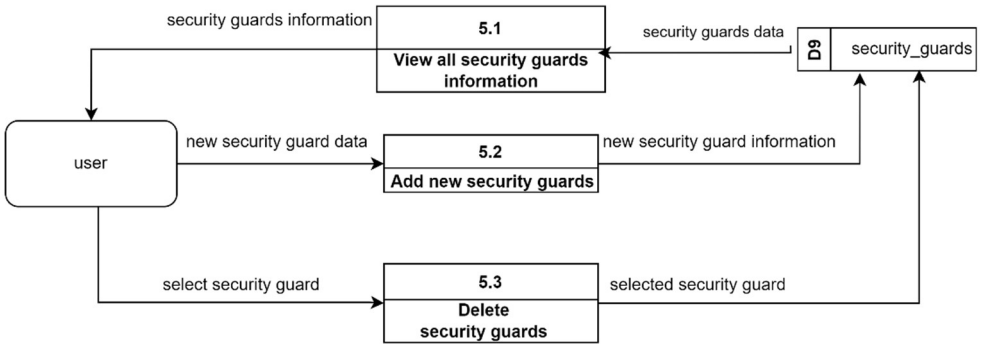


Figure 5.12 Data Flow Diagram Level-1 (5.0 Manage Security Guards)

8.0 Manage Announcement

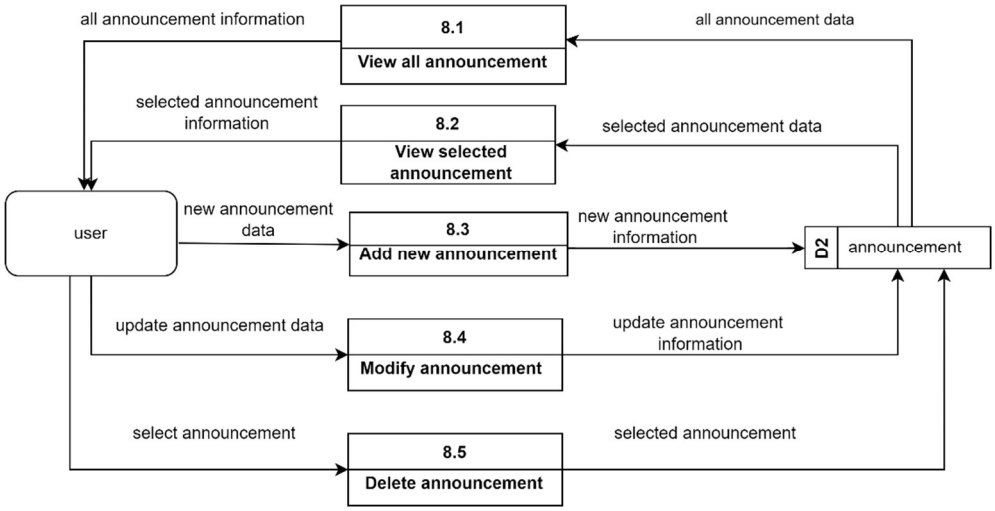


Figure 5.13 Data Flow Diagram Level-1 (8.0 Manage Announcement)

9.0 Manage Feedback

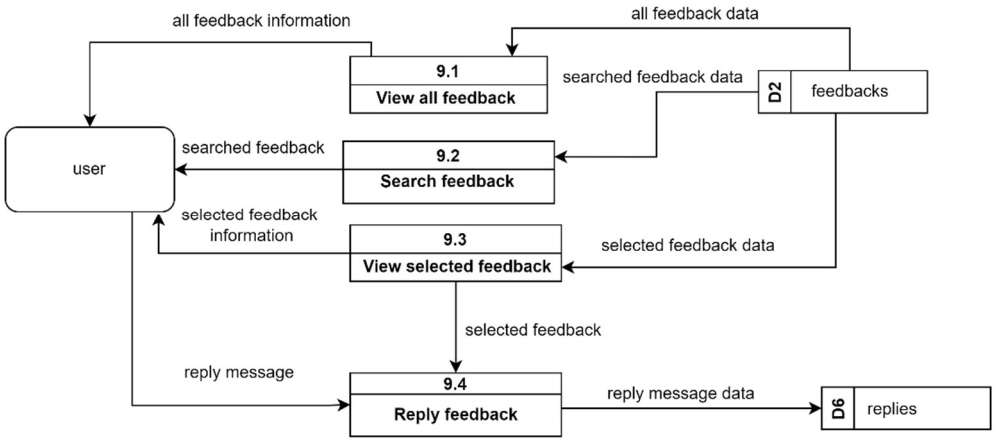


Figure 5.14 Data Flow Diagram Level-1 (9.0 Manage Feedback)

5.3.3.4 Data Flow Diagram Level-1 (Mobile Application)

5.0 Manage Registered Visitation

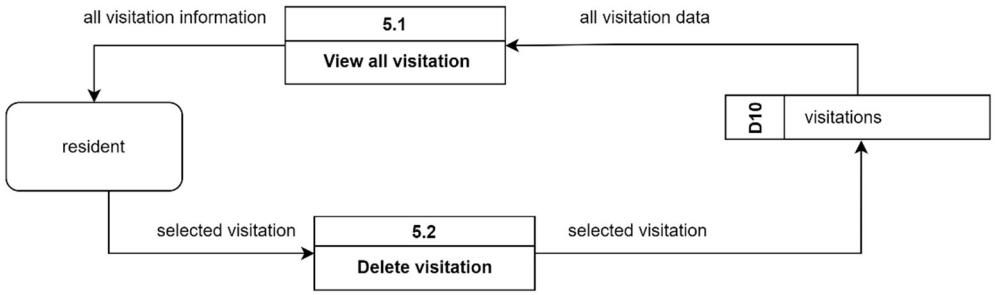


Figure 5.15 Data Flow Diagram Level-1 (5.0 Manage Registered Visitation)

9.0 Manage Feedback

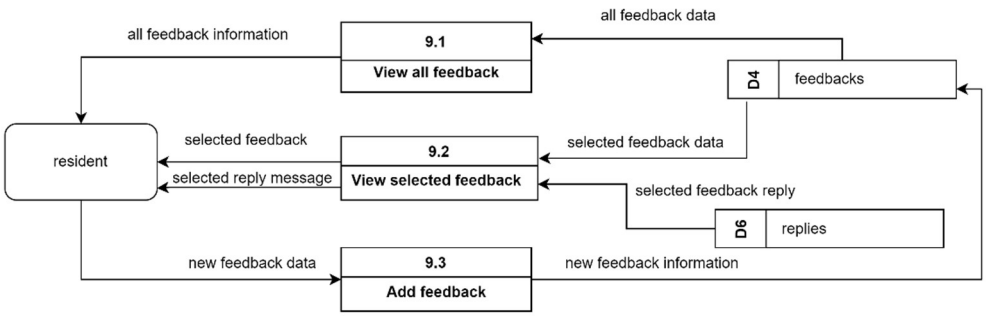


Figure 5.16 Data Flow Diagram Level-1 (9.0 Manage Feedback)

5.4 User Interface Designs

5.4.1 Web Application

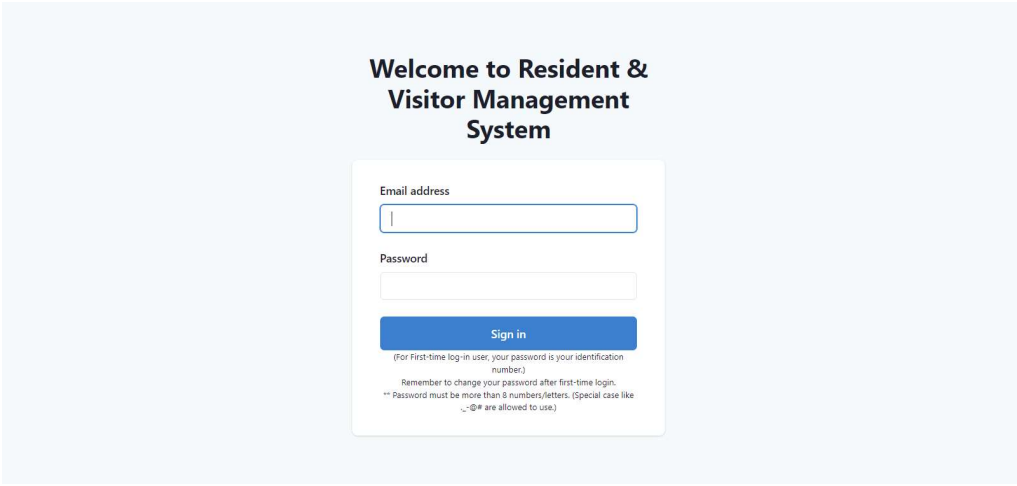


Figure 5.17 Login Page

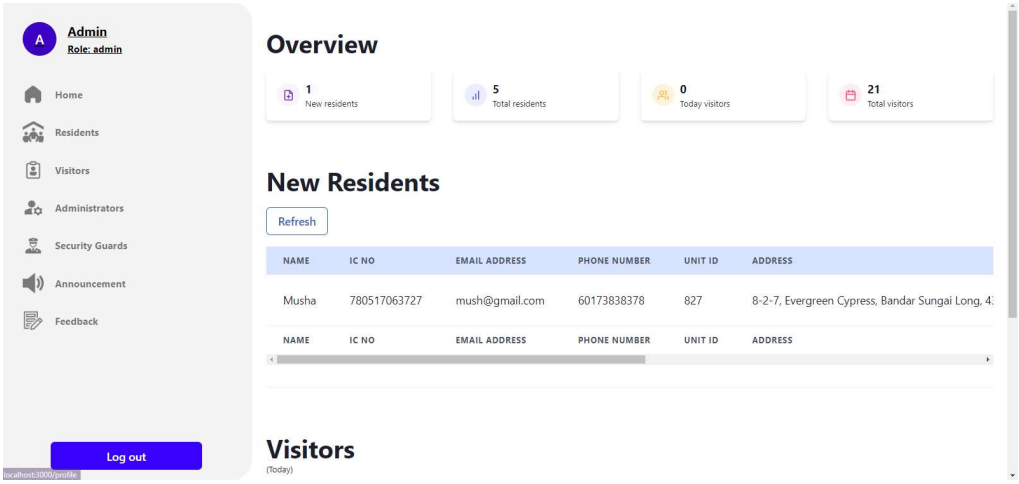


Figure 5.18 Home Page

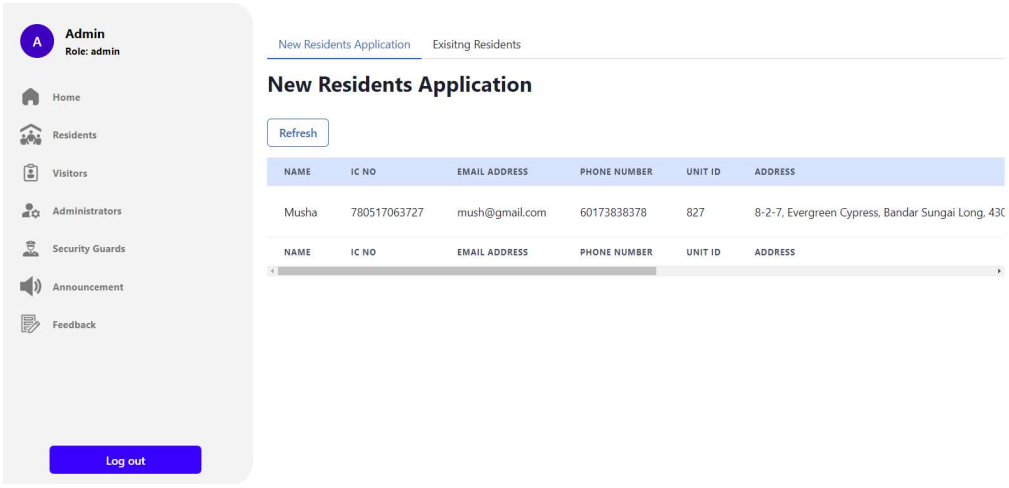


Figure 5.19 New Residents Registration Page

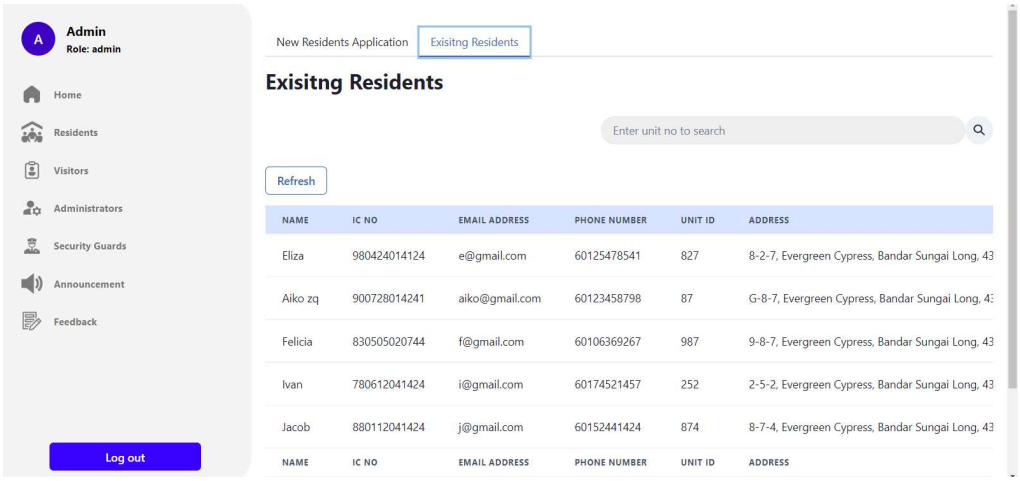


Figure 5.20 Exisitng Resident Page

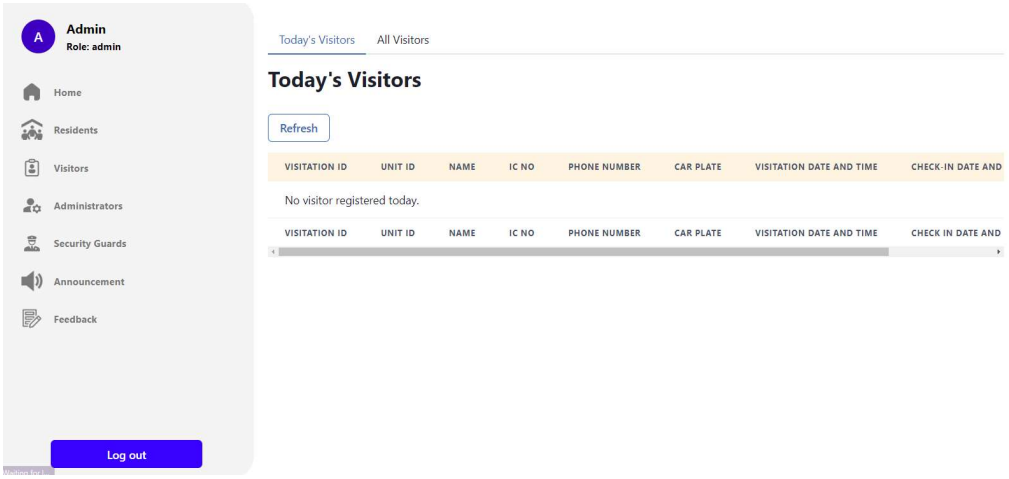


Figure 5.21 Today Visitor's Page

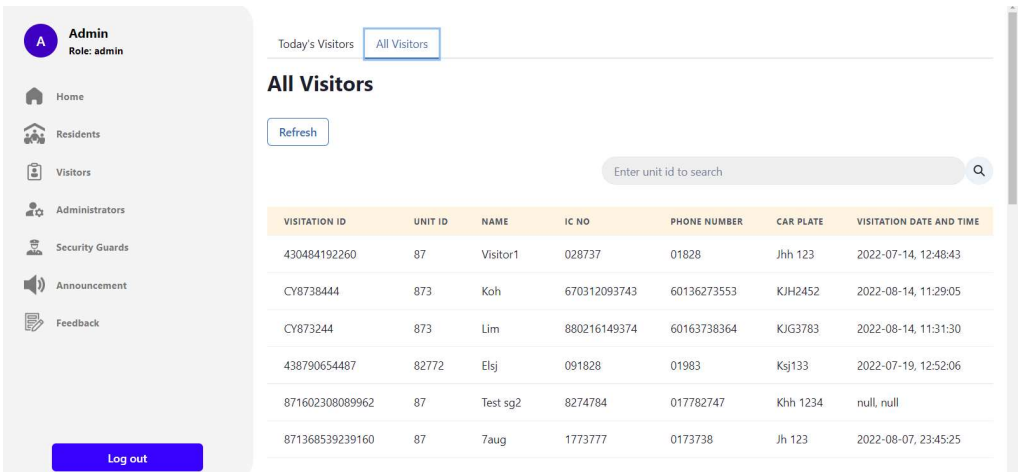


Figure 5.22 All Visitation Page

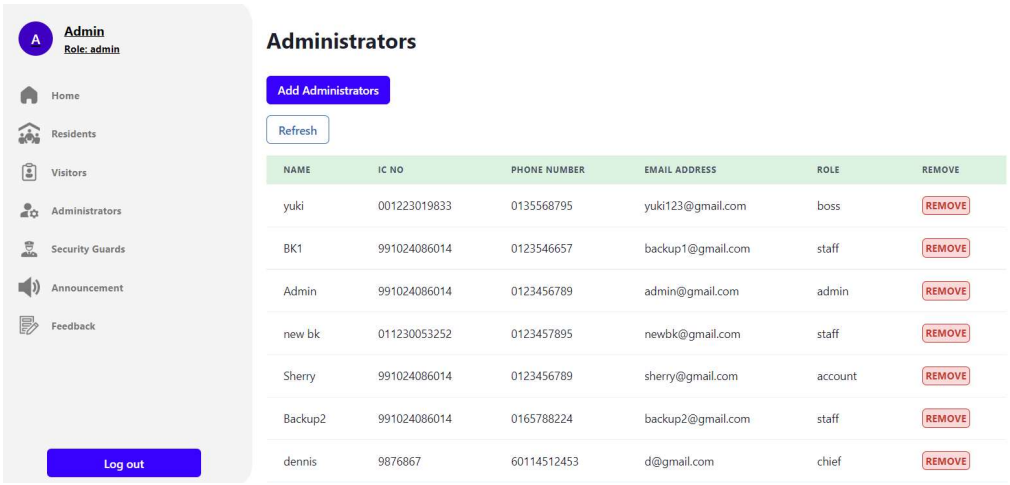


Figure 5.23 Adminsitrators Page

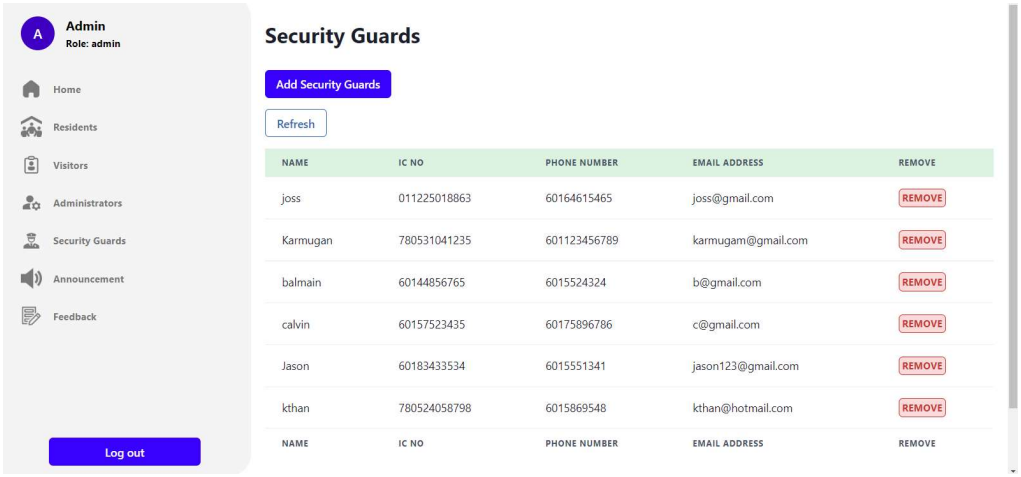


Figure 5.24 Security Guards Page

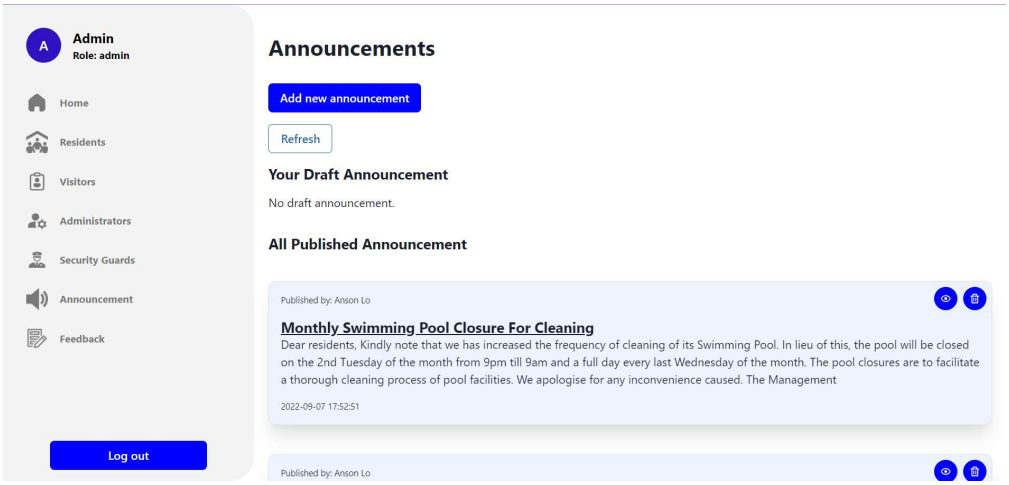


Figure 5.25 Announcements Page

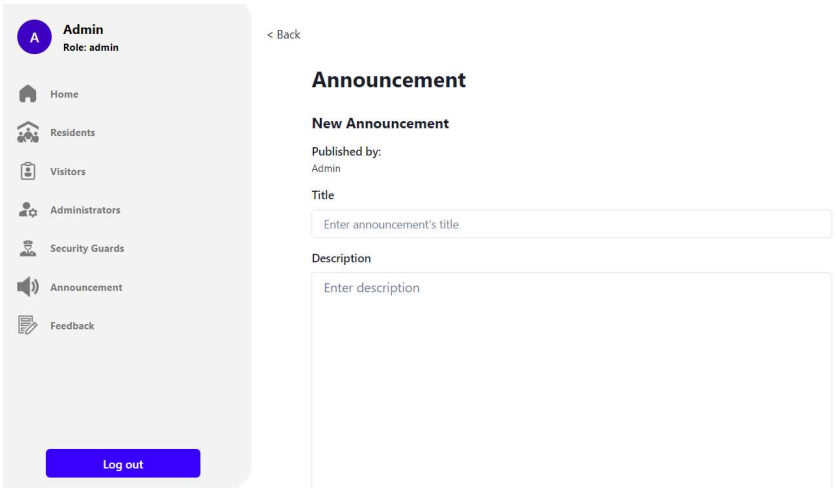


Figure 5.26 Add New Announcement Page

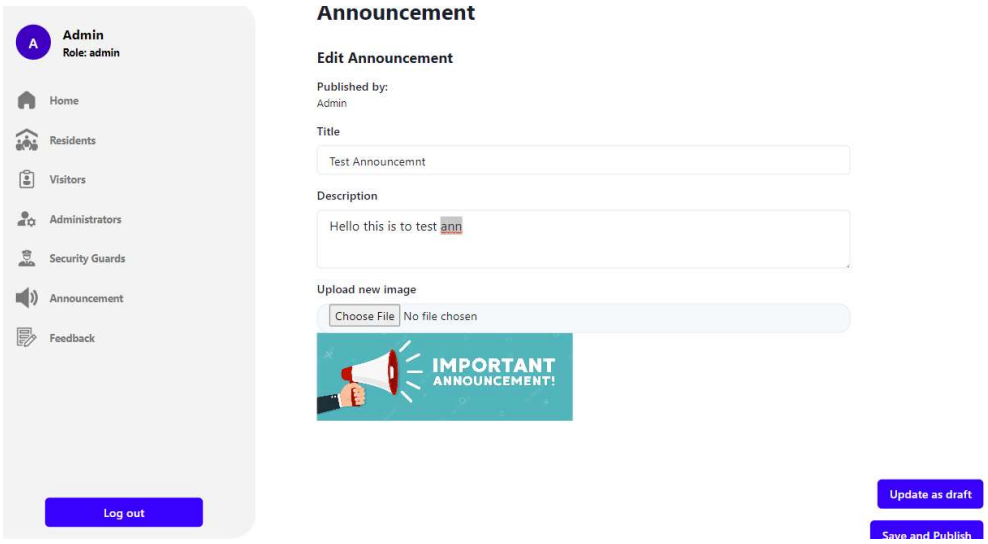


Figure 5.27 Edit Announcement Page

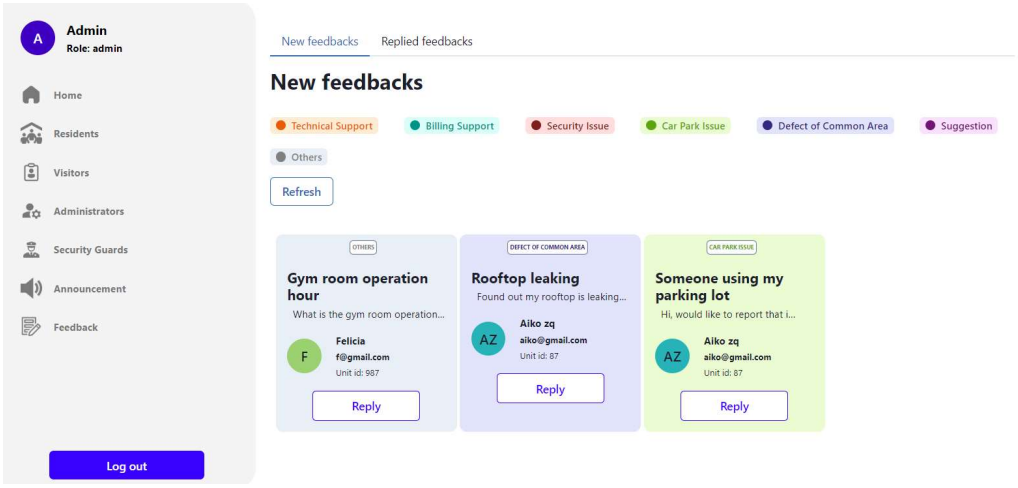


Figure 5.28 New Feedback Page

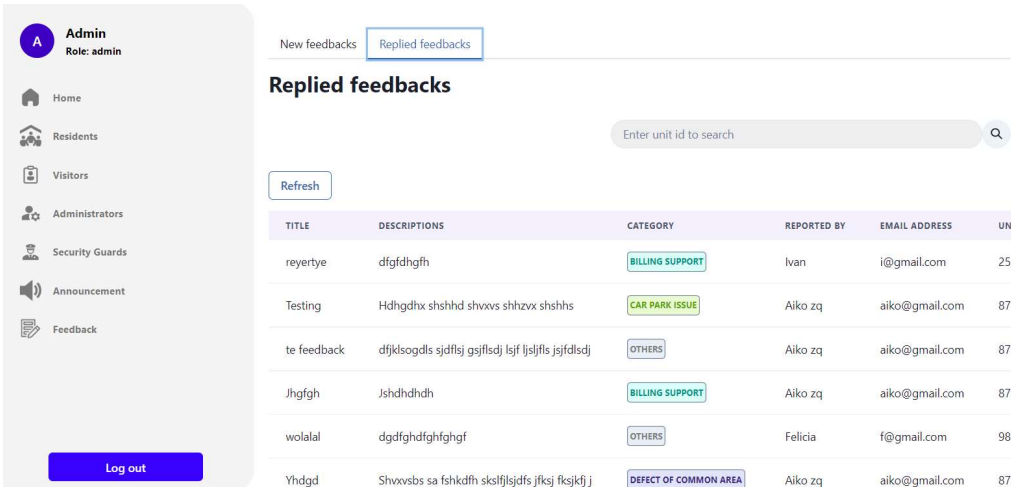


Figure 5.29 Replied Feedback Page

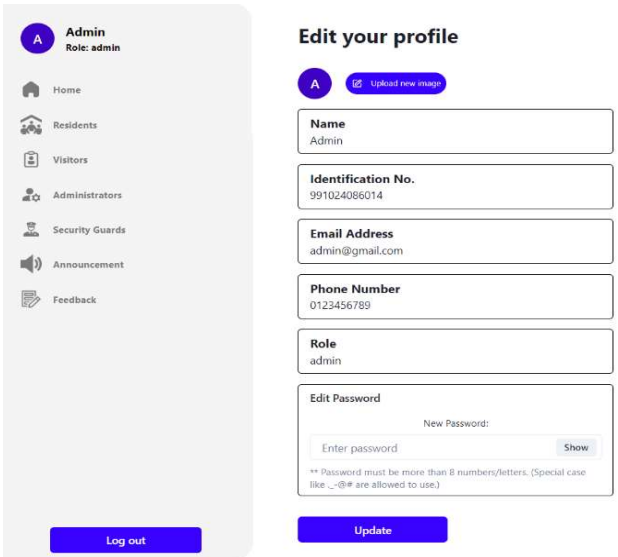


Figure 5.30 Edit Profile Page

5.4.2 Mobile Application

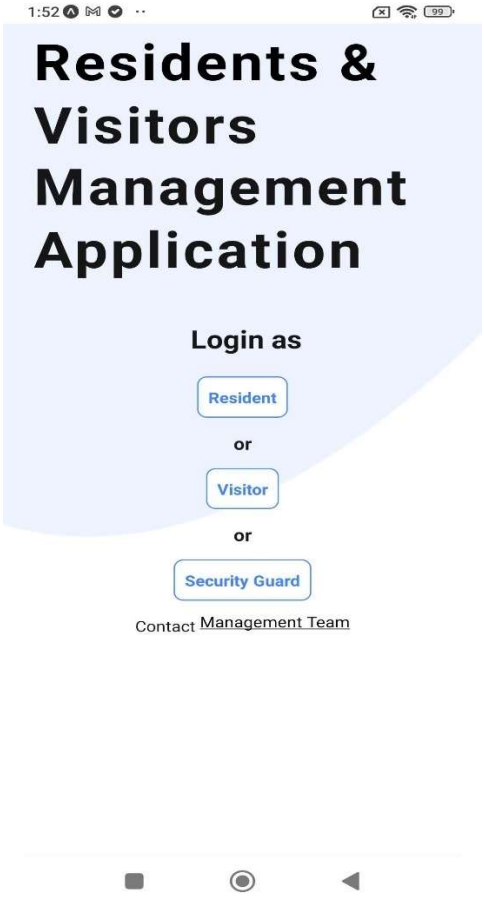


Figure 5.31 Welcome Page

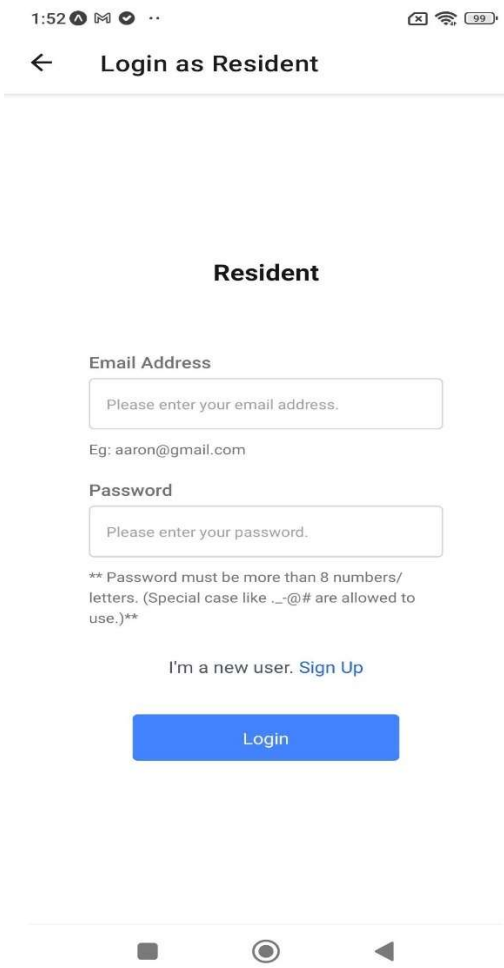


Figure 5.32 Login as Resident Page

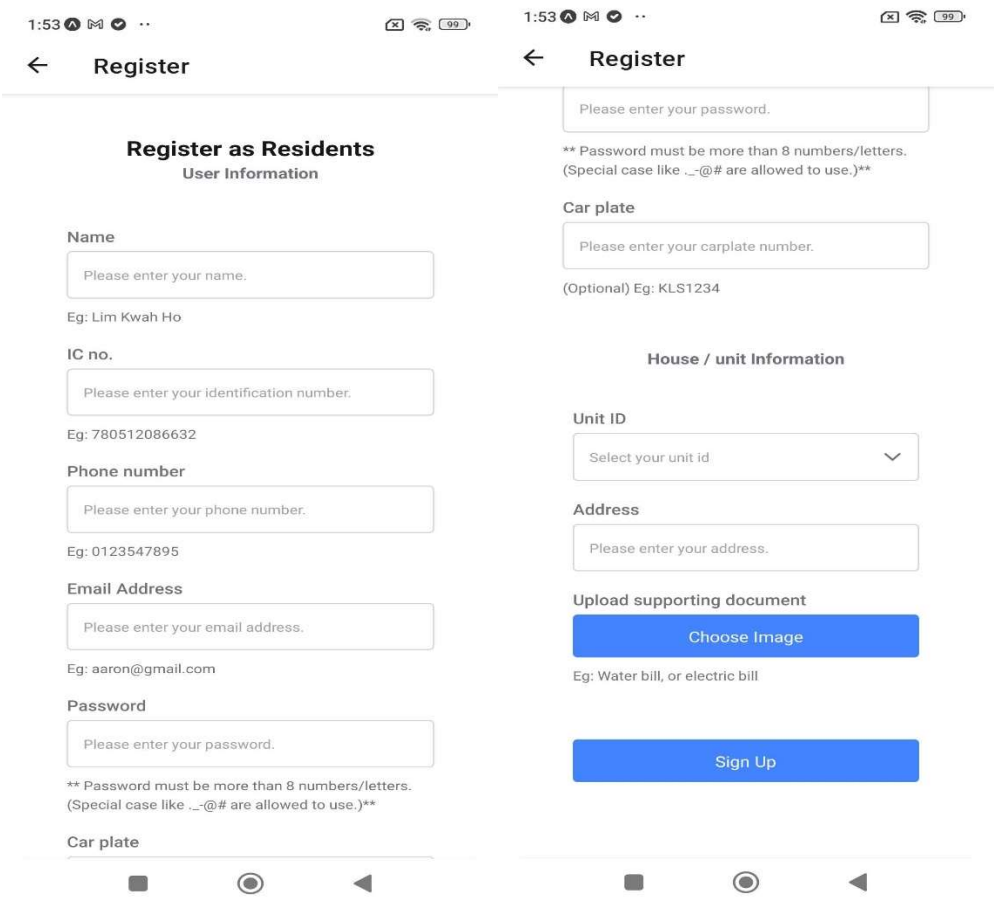


Figure 5.33 Signup Page

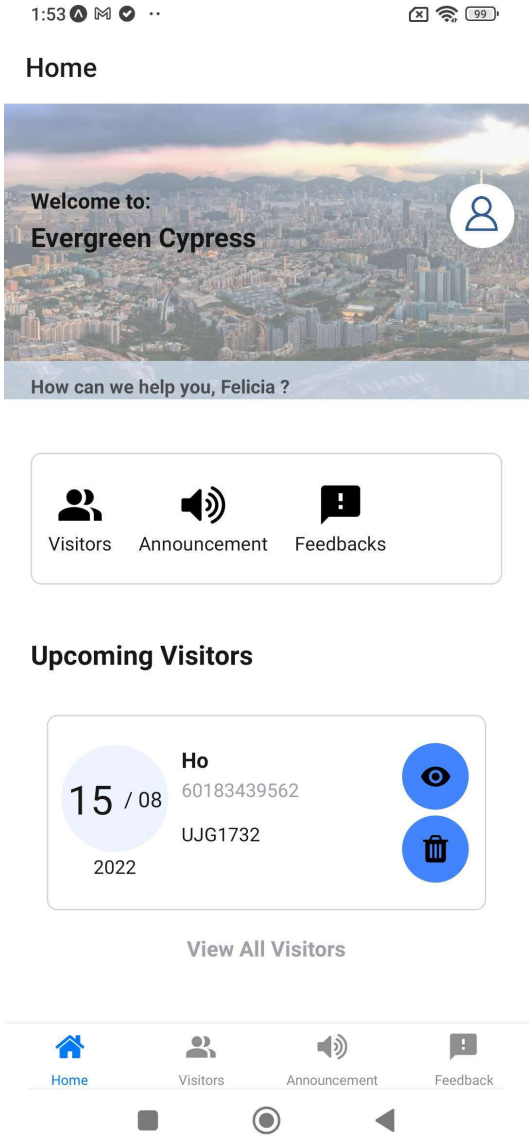


Figure 5.34 Home Page

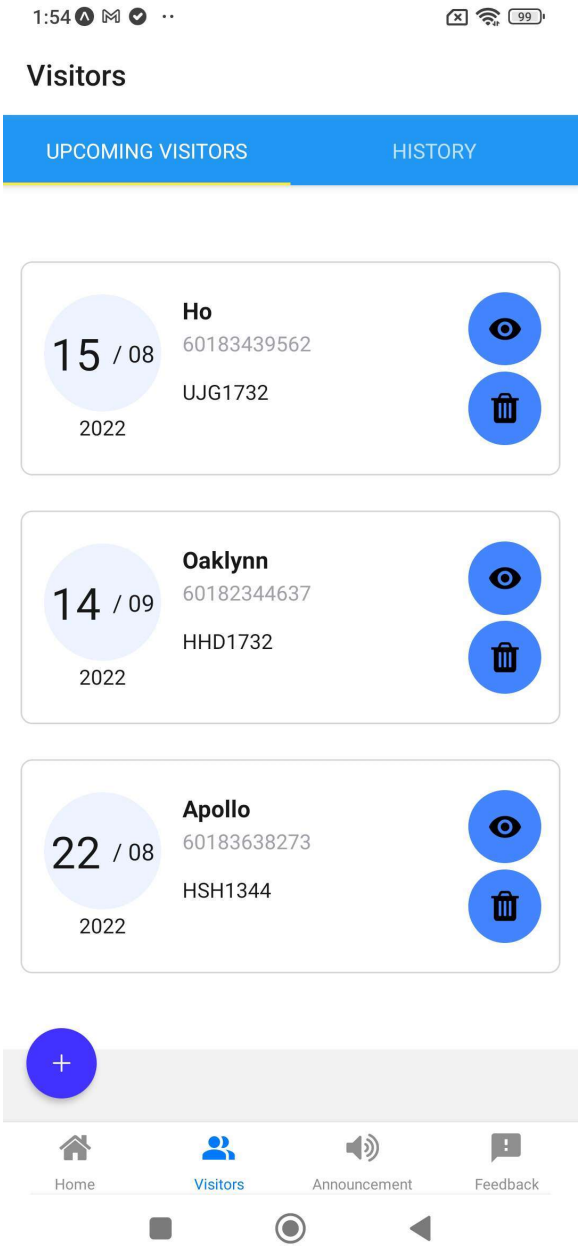


Figure 5.35 Upcoming Visitor Page

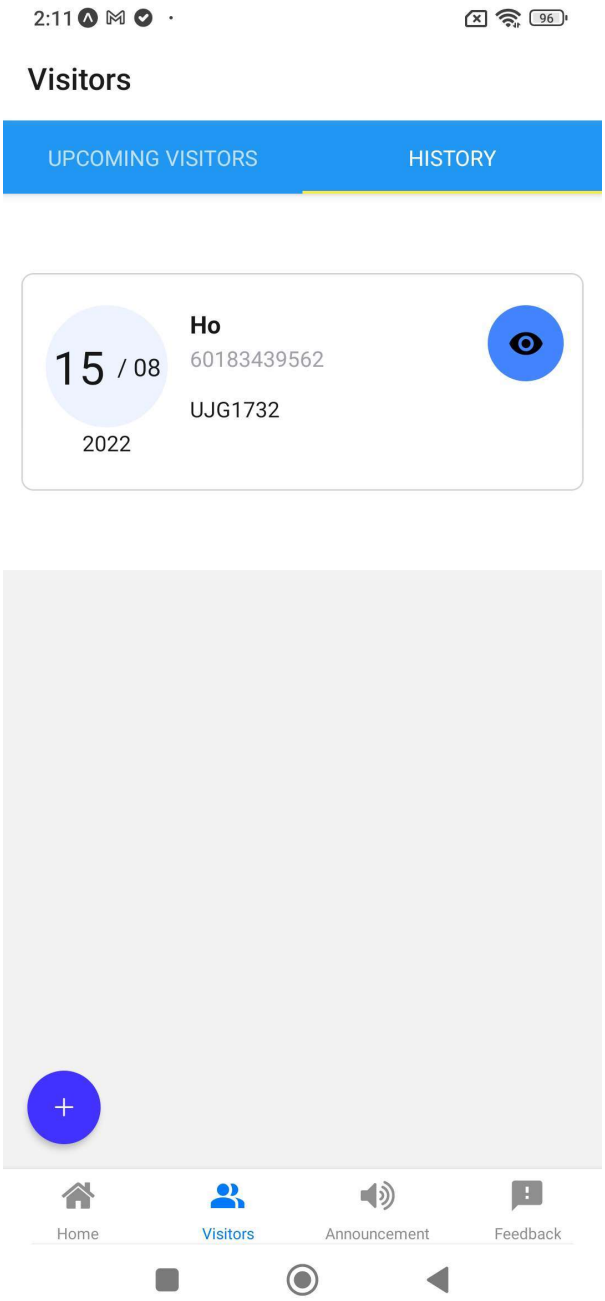


Figure 5.36 Visitation Hitory Page

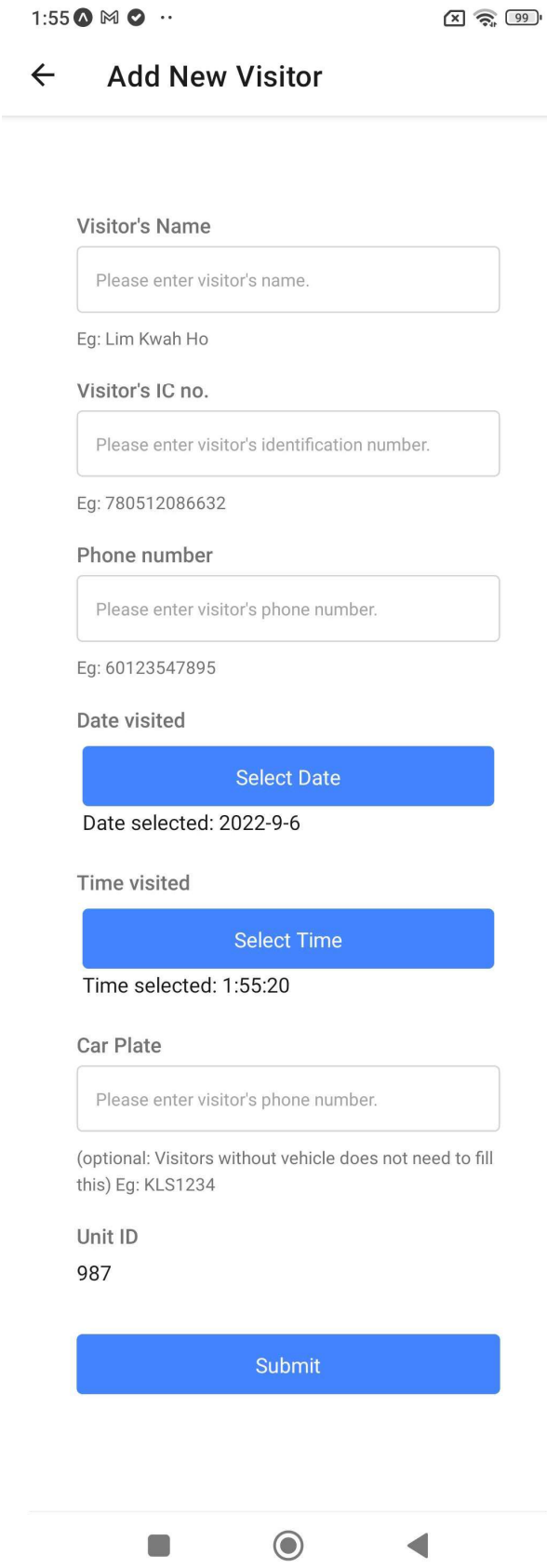


Figure 5.37 Add New Visitor Page

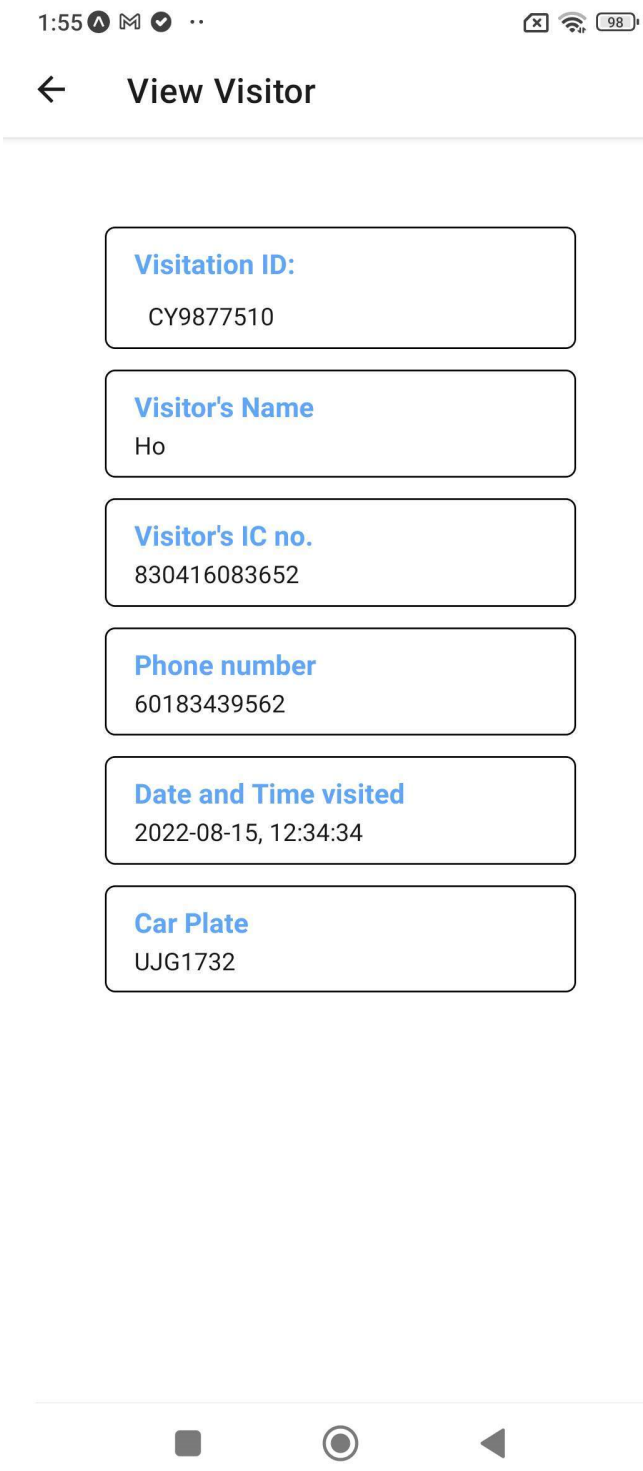


Figure 5.38 View Visitor Page

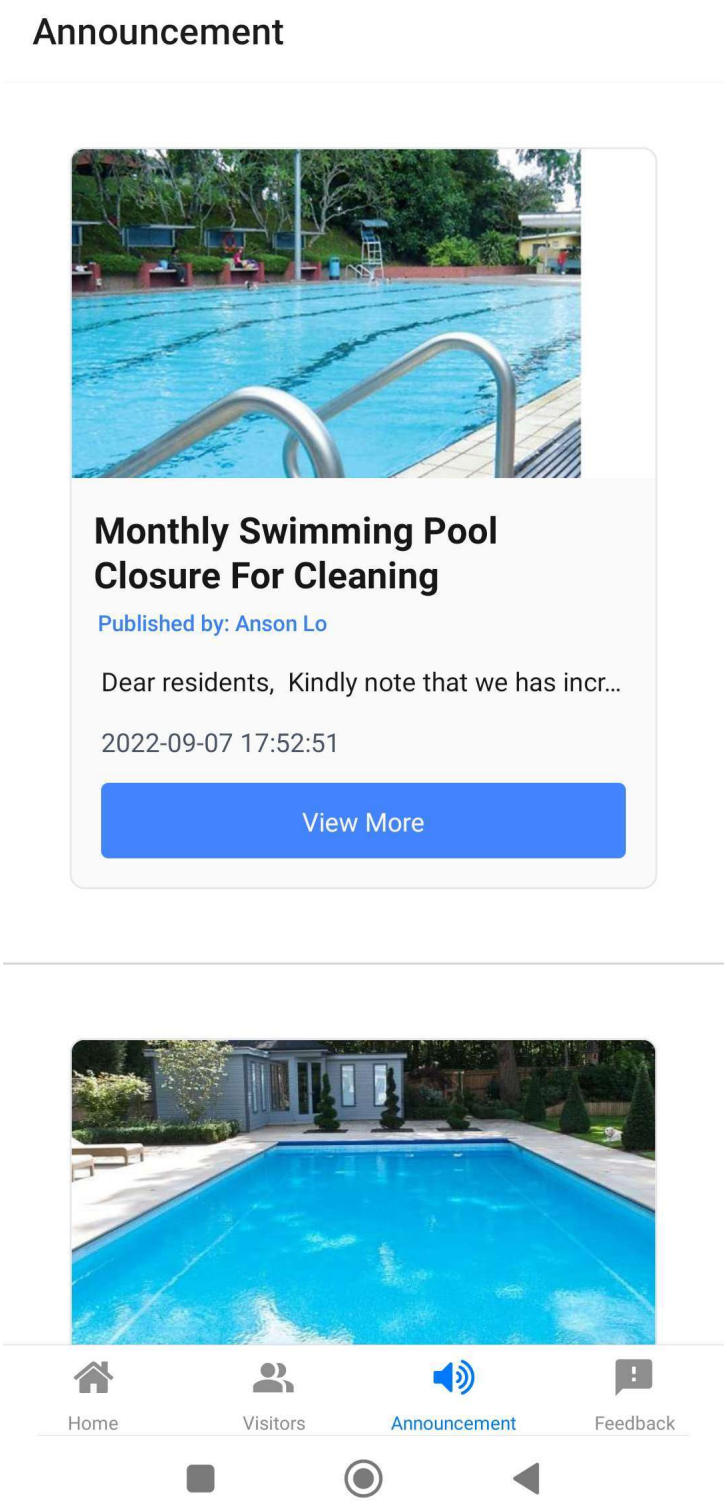


Figure 5.39 Announcement Page

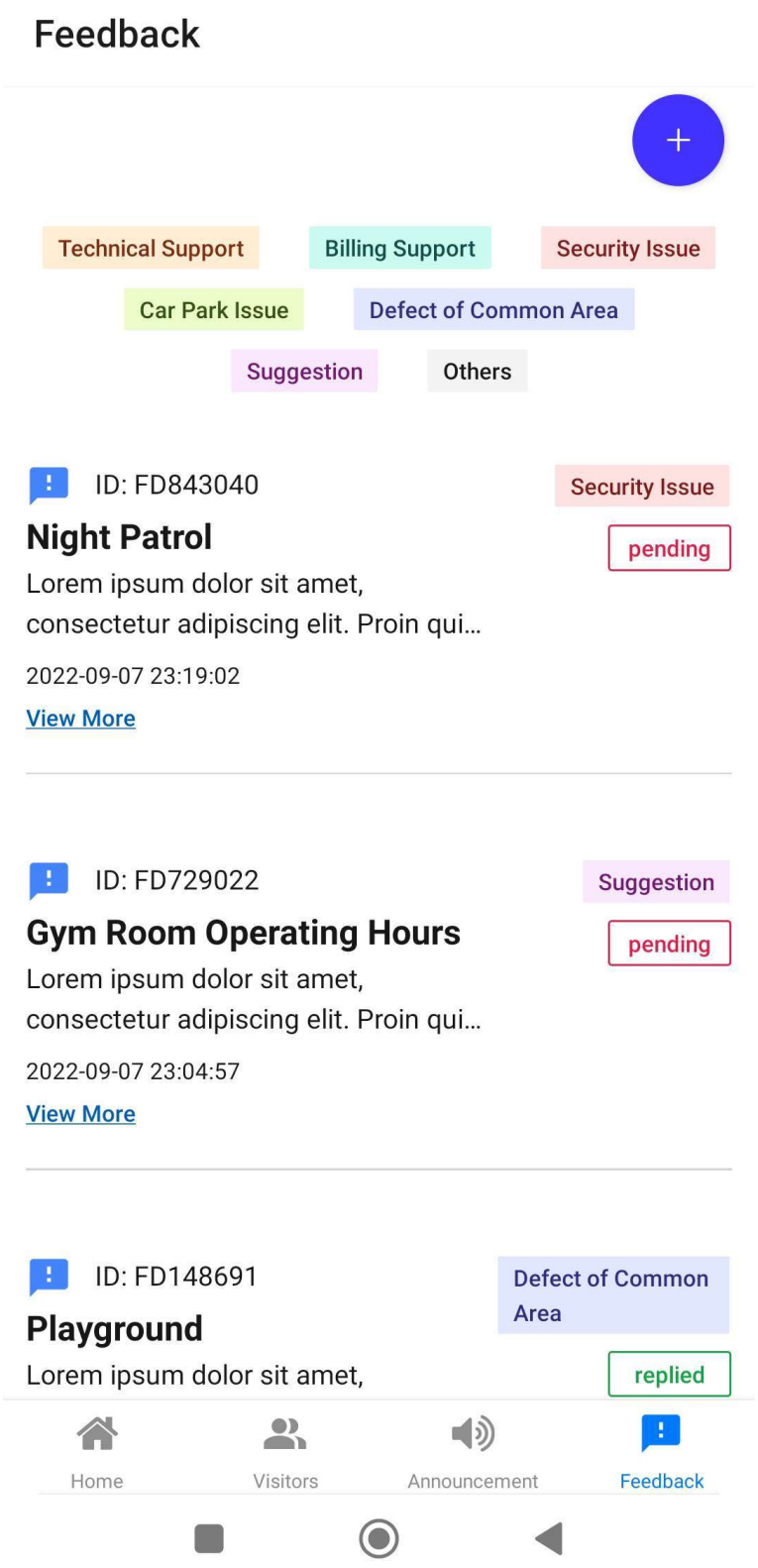


Figure 5.40 View Feedback Page

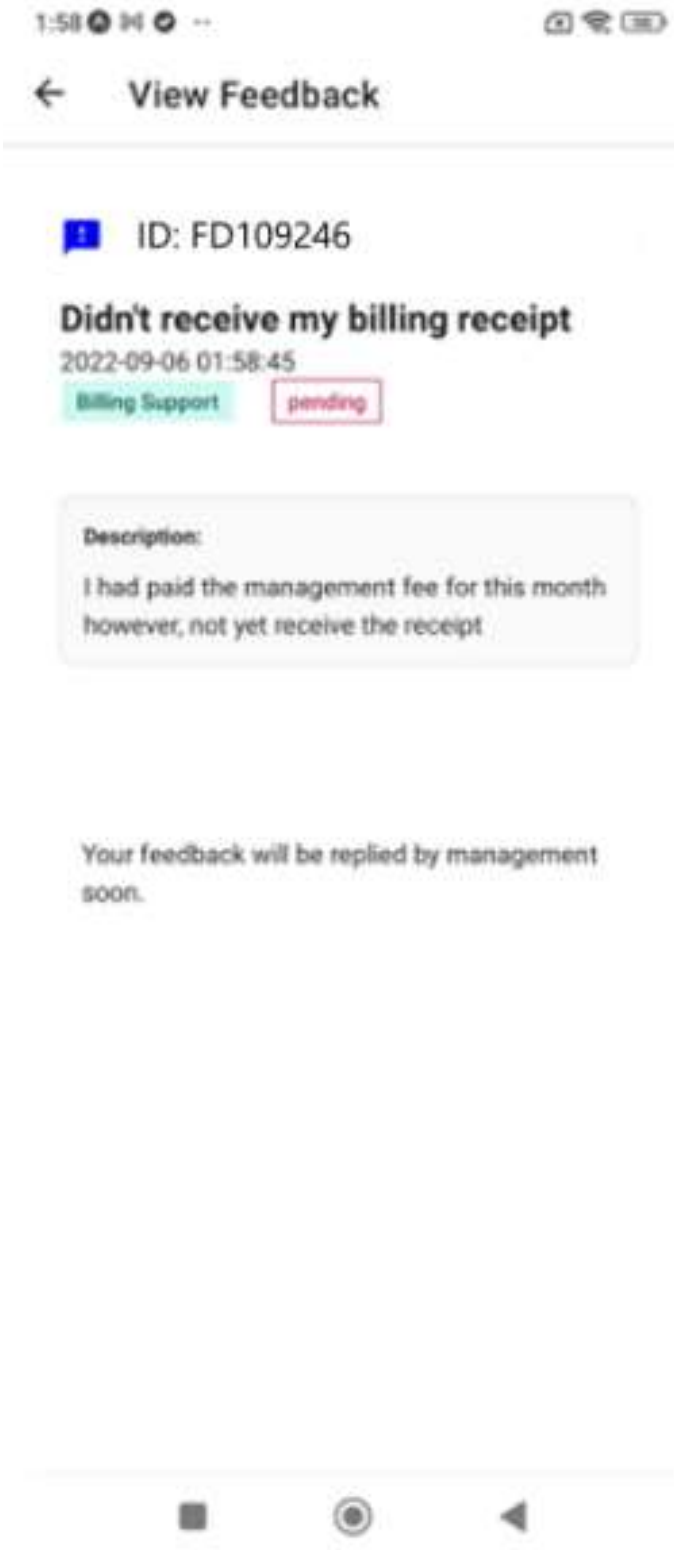


Figure 5.41 View Pending Feedback Page

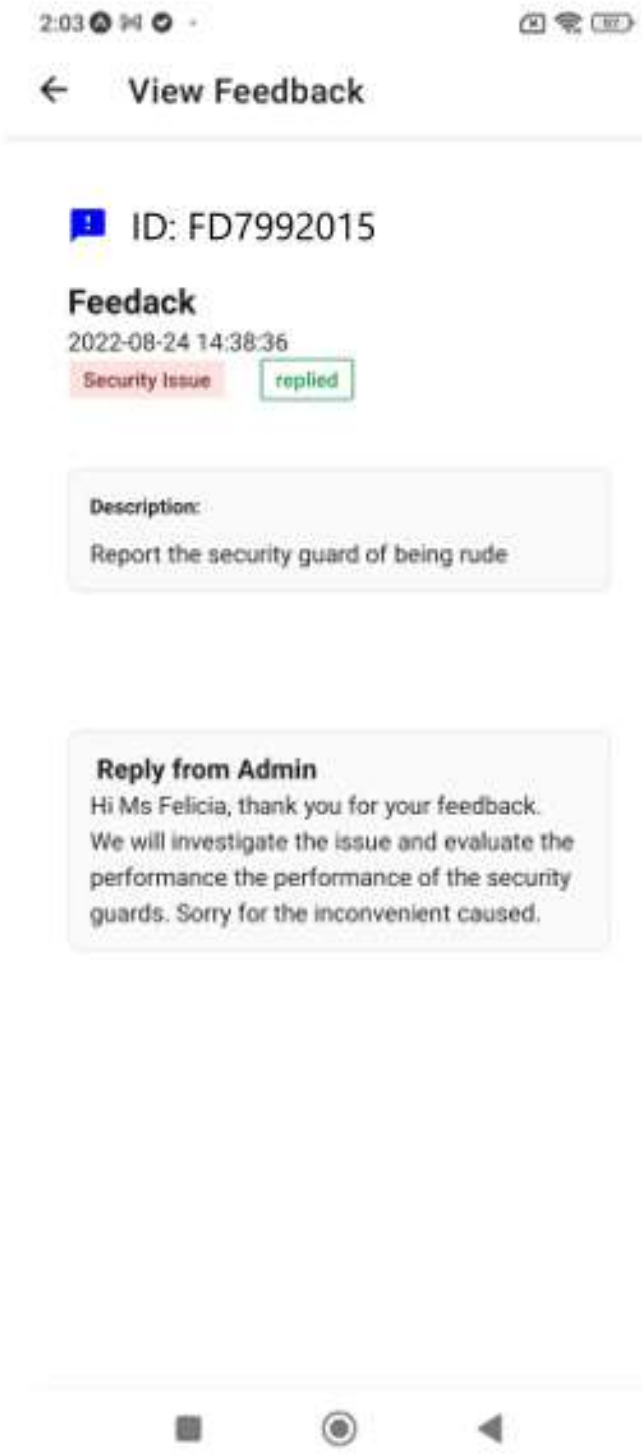


Figure 5.42 View Replied Feedback Page

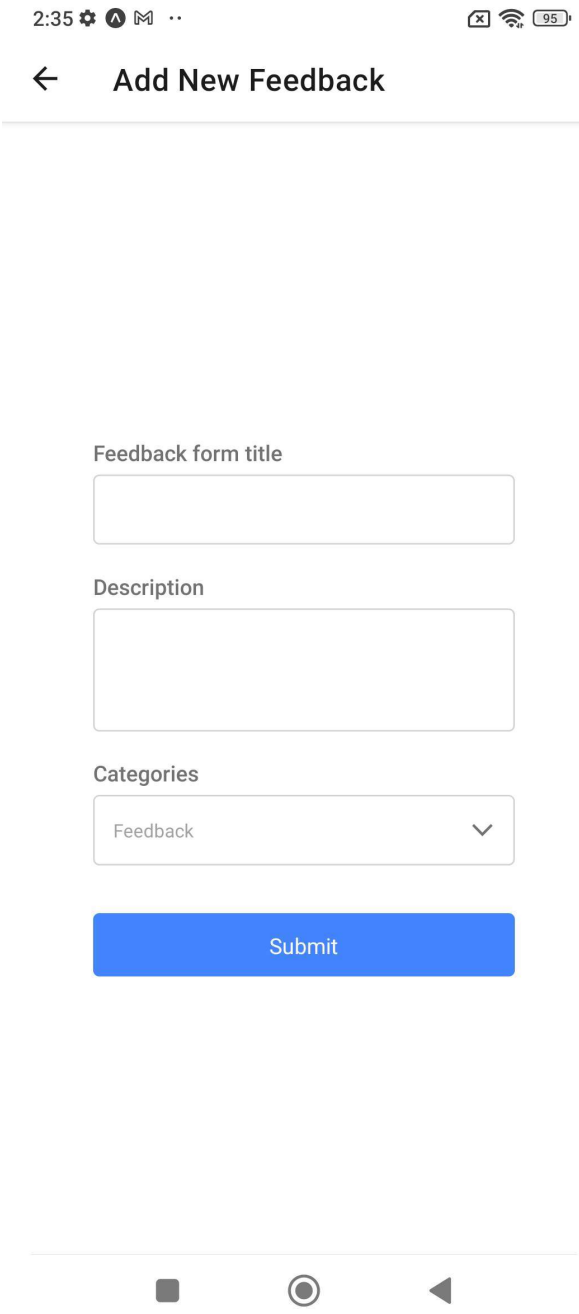


Figure 5.43 Add New Feedback Page

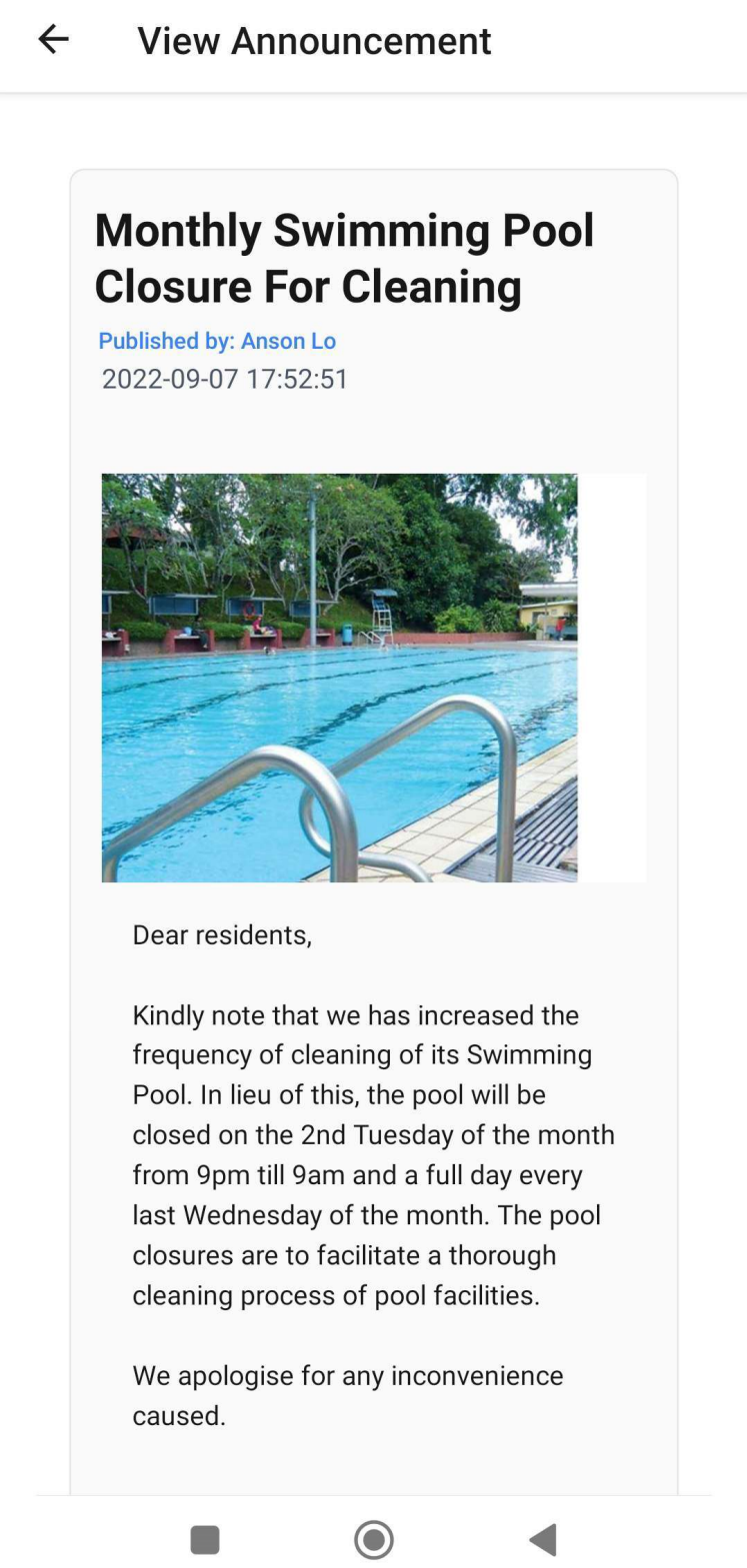


Figure 5.44 View Announcement Page

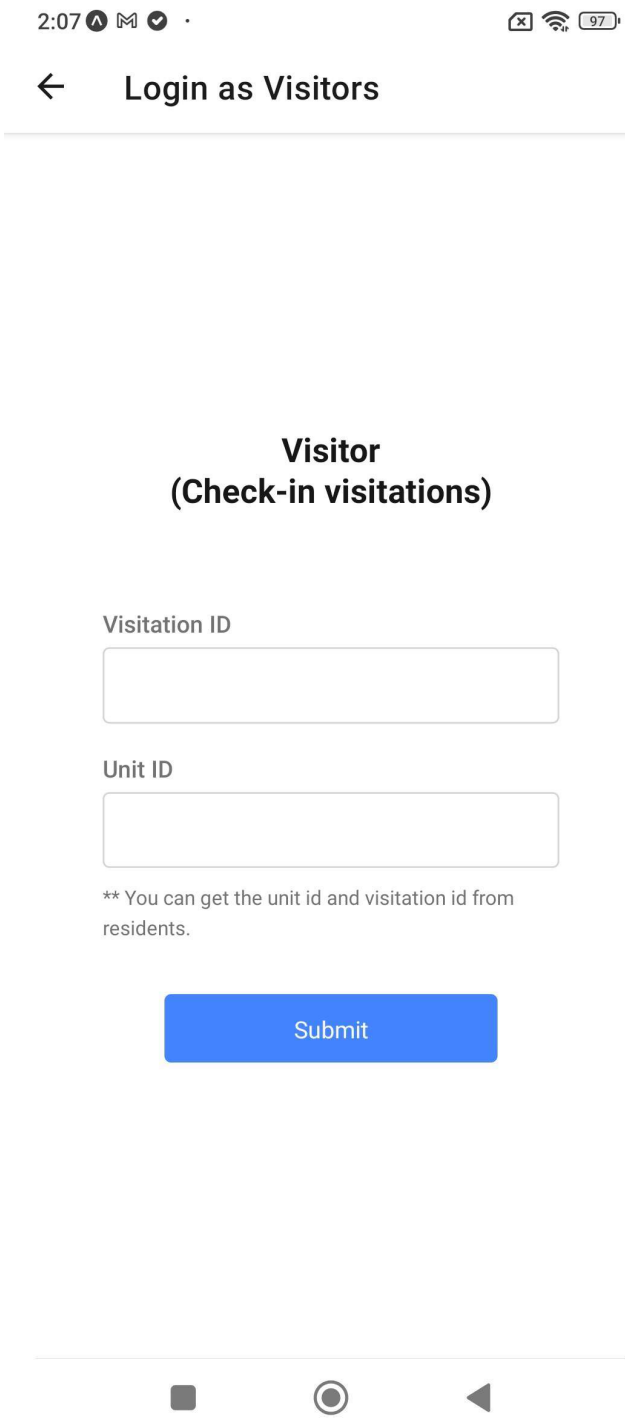


Figure 5.45 Login as Visitor Page

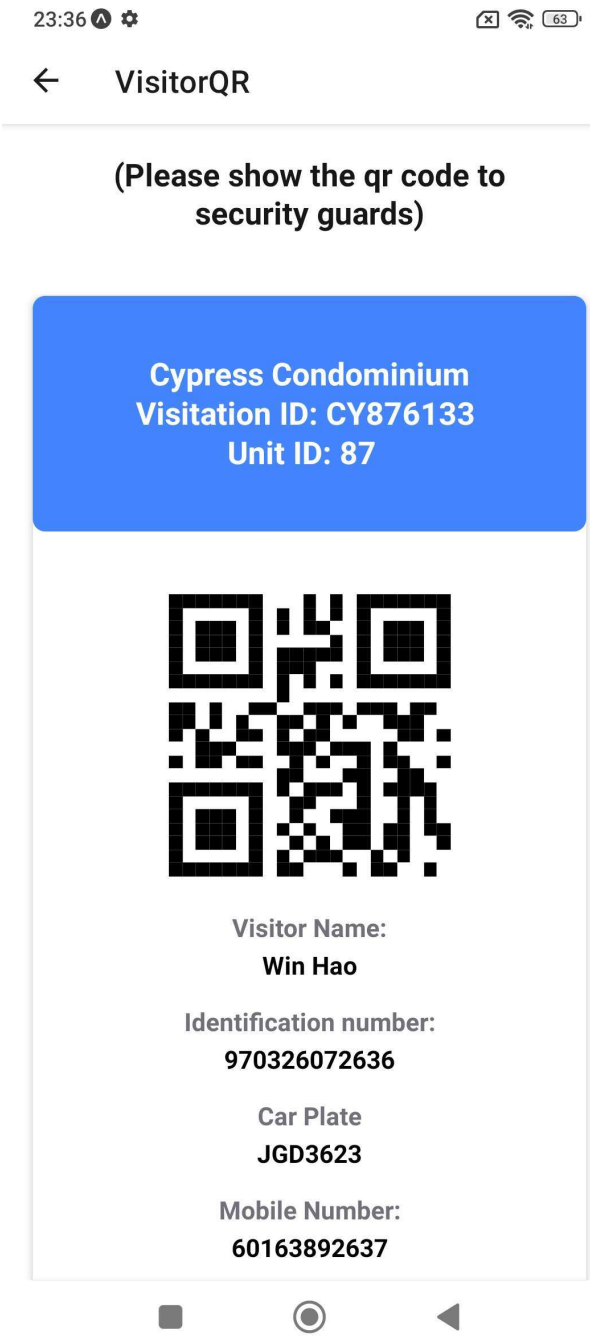


Figure 5.46 Check in Visitation Page

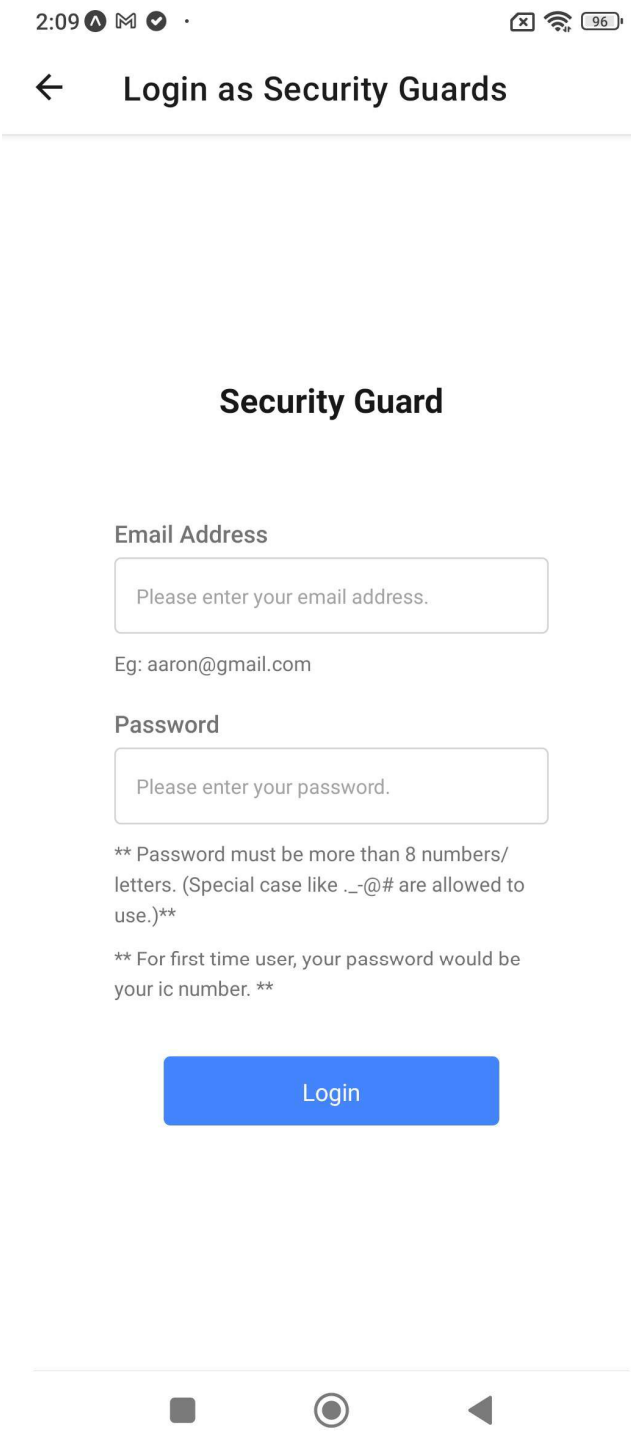


Figure 5.47 Login as Security Guard

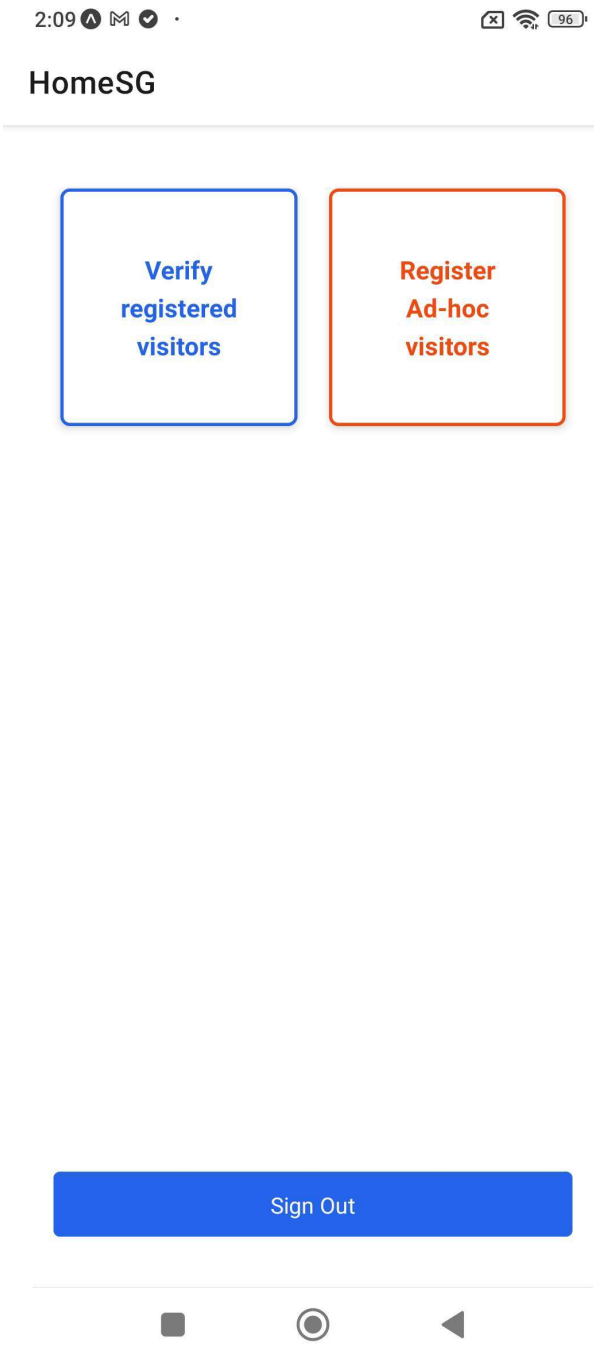


Figure 5.48 Home Page

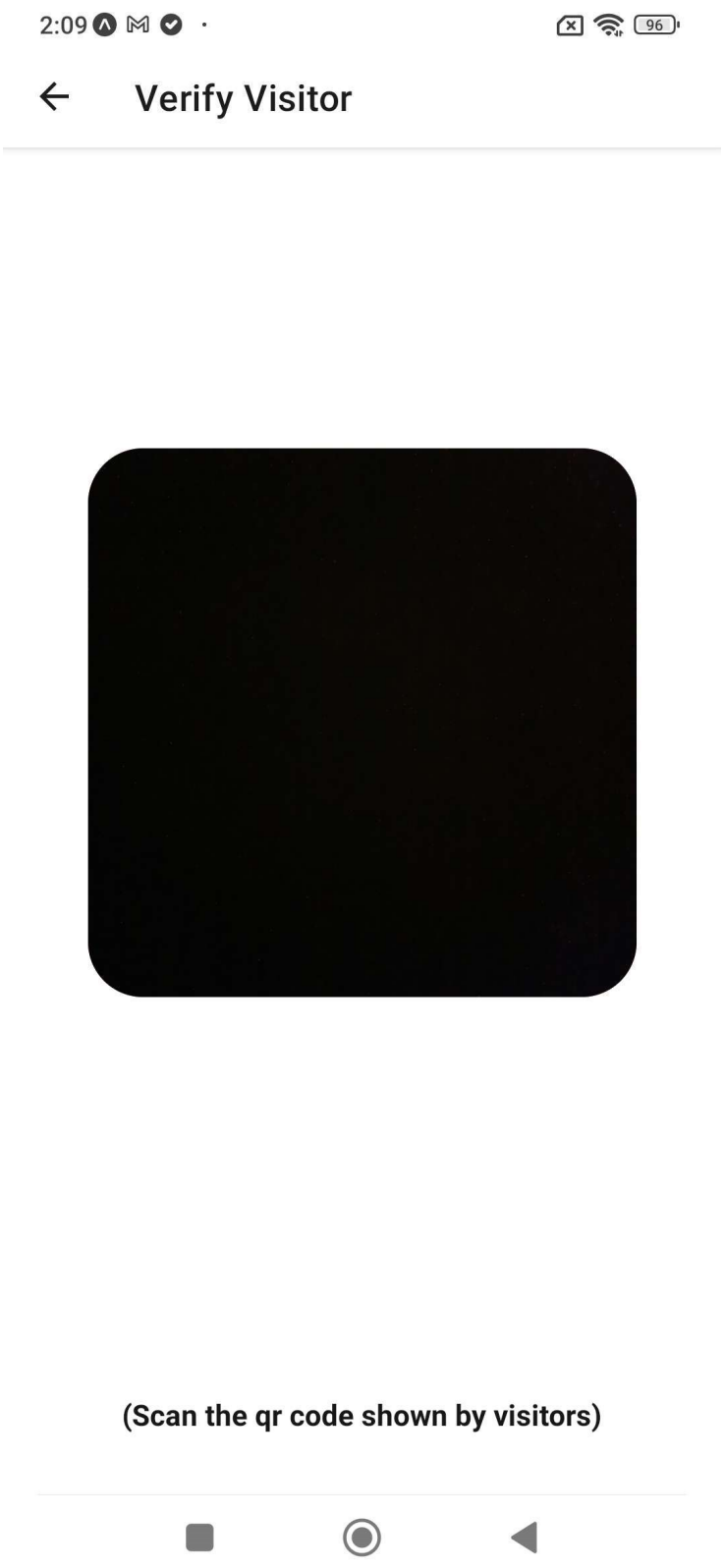


Figure 5.49 Verify Check in Page

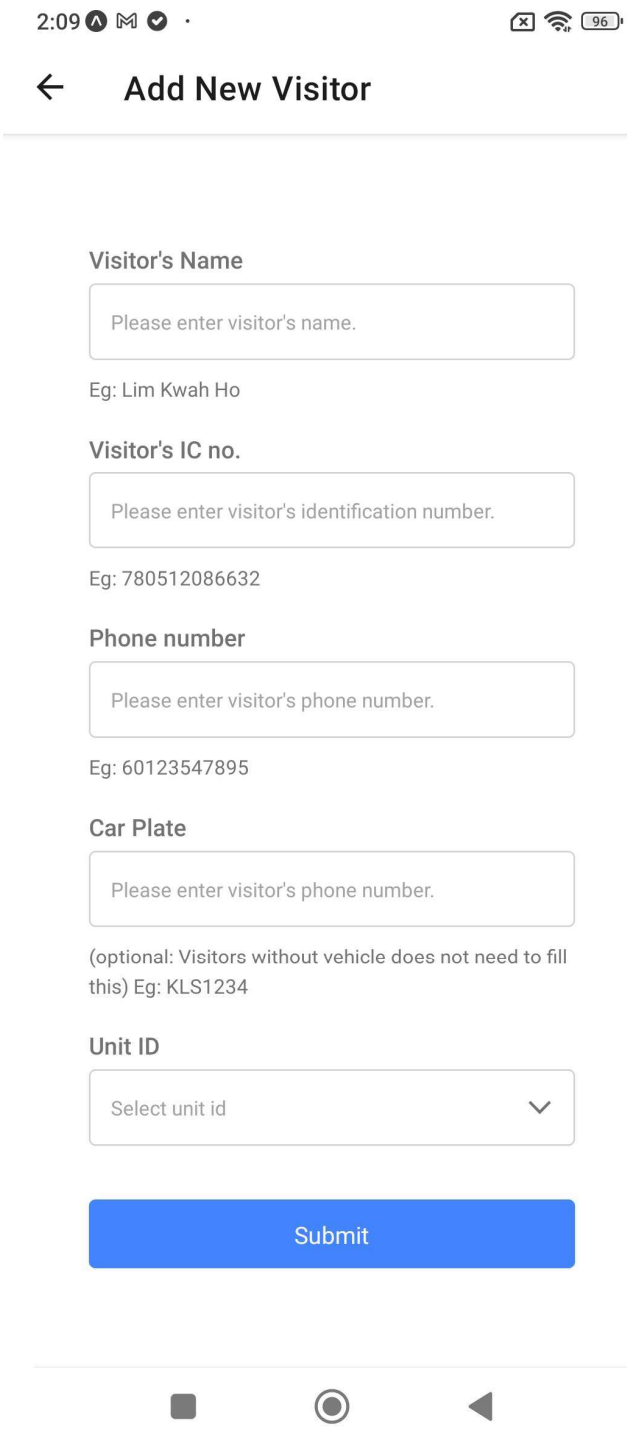


Figure 5.50 Add New Visitor Page (Security Guard View)

CHAPTER 6
SYSYEM IMPLEMENTATION

6.1 Introduction

This chapter focuses on the project's system implementation. It covers the backend implementation of supabase, authentication and authorization, frontend libraries used and deployment process.

6.2 Implementation of Supabase

Supabase offers a JavaScript client library named supabase-js for consuming applications. The client library approach is modular in nature. Supabase maintains each sub-library as a separate implementation for a single external system. Supabase-js leverages internal client libraries that are mapped to the respective Supabase technical building blocks as an entry point. These building blocks are showed in the Figure 6.1 and further discussed in this Chapter.

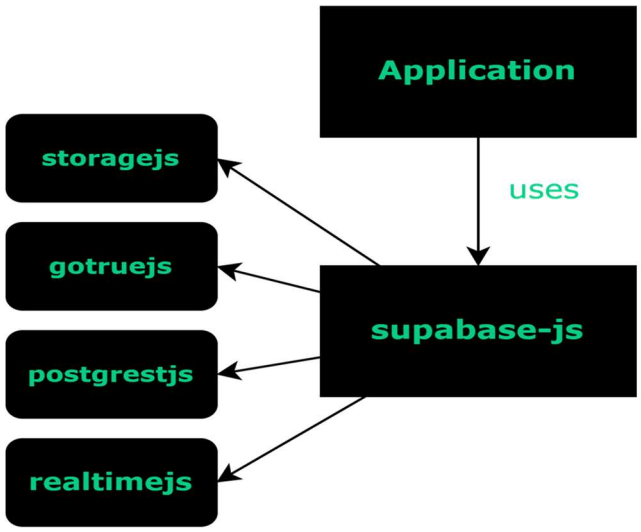


Figure 6.1 Building Blocks of the Supabase JavaScript Client Libraries

6.2.1 Postgrest-js

A JavaScript client for PostgREST is called Postgrest-js. Making a restful interface that is similar to an Object-Relational Mapping (ORM) is the goal of this module. It is responsible for the /rest endpoint. It helps to perform CRUD operation with the function and all the related function used and operation is listed in the Table 6.1.

Table 6.1 Postgrest-js

Function	**Operation**
select()	Read
insert()	Create
update()	Update
delete()	Delete

6.2.2 Gotrue-js

Gotrue-js is responsible for the /auth endpoint. It offers the ability to log in, log out, and other functions. It assists with user authentication and registration in the project. It will also handle user signup, authentication, and customised user data and is based on OAuth2 and JSON Web Tokens(JWT). The related function and its description are listed in Table 6.2 below.

Table 6.2 Gotrue-js

Function	**Description**
signIn()	To log in the application
signOut()	To log out the application
signUp()	To create a new user and add the data to the authentication database.
session()	To get the session data, if there is an active session

user()	To get the user data, if there is a logged in user. Example: get user's session id
update()	To update the user data, if there is a logged in user. Example: update password

6.2.3 Realtime-js

The /realtime endpoint is accessible using Realtime-js. It allows the client to subscribe to changes like UPDATE, CREATE, DELETE in PostgreSQL database via logical replication and then broadcast those changes via WebSockets. The function of realtime.js and its description is listed in the Table 6.3.

Table 6.3 Realtime-js

Function	**Description**
on().subscribe()	To subcribe to realtime changes in database.
removeSubscription()	To remove an active subscription and return the number of open connections.

6.2.4 Storage.js

The /storage endpoint is accessible through Storage-js. It enables the client to interact with Supabase Storage. Supabase Storage is intrgrated with Postgres Database and used to store and serve large files. The files can be any sort of media file which includes image, GIFs, and videos. It is best practice to store files outside of database because of their sizes. Buckets are distinct containers for files. Generally, each buckets is created with different Security and Access Rules. For example, files in bucket can be set as public which accessible for

everyone or set as restricted which require logged-in access. The function of storage used in this project and its description is listed in the table below.

Table 6.4 Storage-js

Function	**Description**
from.upload()	To upload the file to existing bucket
from.download()	To download files
from.remove()	To delete files

6.3 Authentication and authorization

6.3.1 Authentication

The authentication of this project used is gotrue-js as mentioned as above.

6.3.2 Authorization

The authorization of this project used is postgres's row level security. It is a granular authorization rules where the level of details used to put on authorization rules for deciding to deny or grant the access. By default, each tables do not have any policy. Each policy is attached to a table, and the policy is executed every time a table is accessed. Each policy has a name, and a table can have numerous policies set for it. Each policy for a table must have a distinct name because policies are table-specific. Policies having the same name across several tables are possible. If row security is implemented on the table any normal access to a table for choosing rows or editing rows must be authorised by a row security policy if row security is implemented on the table. A default-deny policy is used if there is no policy for the table, which prevents any rows from being displayed or modifiable. An expression that yields a Boolean result is necessary in order to determine which rows are viewable or modified in accordance with a policy. Prior to any conditions or functions derived from the user's query, this expression will be evaluated for each row. The expression will not process any rows for which it does not return true. The CREATE POLICY

command is used to create policies, the ALTER POLICY command is used to change them, and the DROP POLICY command is used to remove them. Use the ALTER TABLE command to activate and disable row security for a specific table. As seen in Figure 6.2 of this project's Supabase dashboard, the CREATE POLICY command is used to create the table residents' policies.

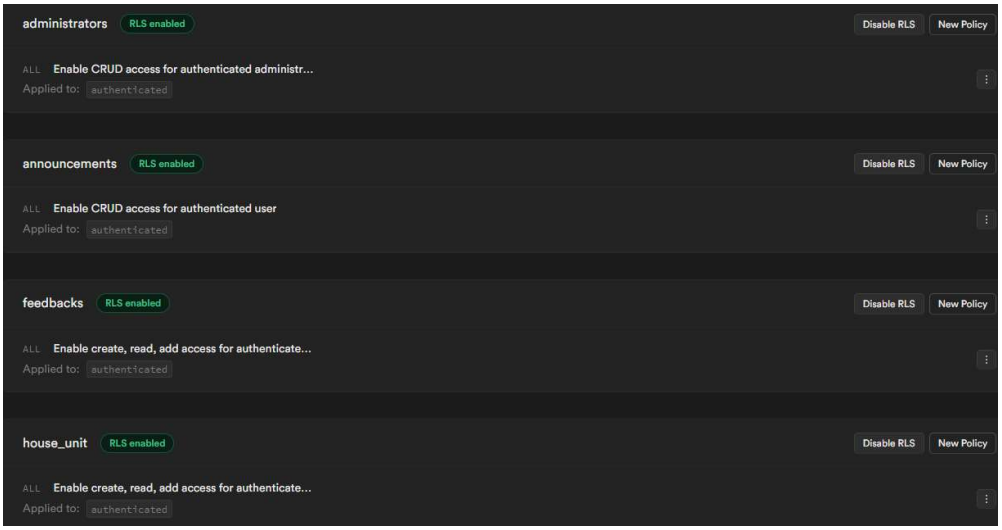


Figure 6.2 Screenshot of Supabase screenshot (RLS enabled and policy added)

6.4 Front end libraries used

ChakraUI and Native Base are used to create attractive and practical components for both the front end of mobile and web applications. Chakra UI is a React components library with built-in accessibility. It includes a sleek design system that is simple to extend and configure. Developers may quickly prototype their ideas and achieve the required style thanks to the straightforward styling API's significant reduction in development time. ChakraUI could also support SSR that function well in Next.js and also offers a huge amount of high-quality React components that are incredibly customizable. This makes the reason of ChakraUI is used in this project. For example, core principles like style props can easily be overridden and extended to reduce the use of cases or styles. An open-source UI library called NativeBase makes it simple to create universal design systems for both Android and iOS platform. NativeBase is supported in Expo, Web, and React Native CLI-initiated projects and was created for React Native. NativeBase has UI components like Image, Button,

Alert, and others built into it. By using those UI components of NativeBase and ChakraUI, the UI of the mobile application was built successfully according to the prototype.

6.5 Deployment

The web application was hosted through Vercel as it is the creator of Next.js. Vercel automatically aliases the preferred domain name to the latest deployment and ensures fresh certificates are installed. A brief setup of the deployment process is shown in Table 6.5.

Table 6.5 Deployment process

No	**Description**
1	Initiate Git in the local project directory
2	Create a Github repository
3	Link remote Github repository with local Git
4	Login to Vercel with same Github account
5	Import the Git Repository and Configure Supabase access
6	Configure the project
7	Pull environment variables from Vercel to local by login Vercel using their CLI tool. Link the Vercel project with the 'npx vercal link'. Then copy the environment variables from Vercel project through 'npx vercel env pull'. It will automatically create .env file containing our Supabase environment variables. Rename this file to .env.local to automatically ignore it from git.
8	Commit the code and push to default branch.
9	Vercel automatically trigger the builds and launches them to the cloud.

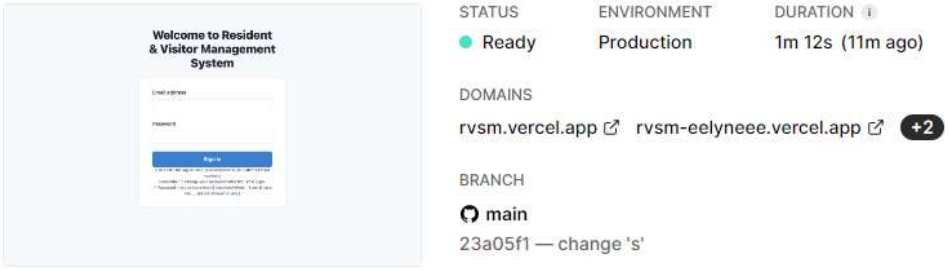


Figure 6.3 Show Vercel Dashboard

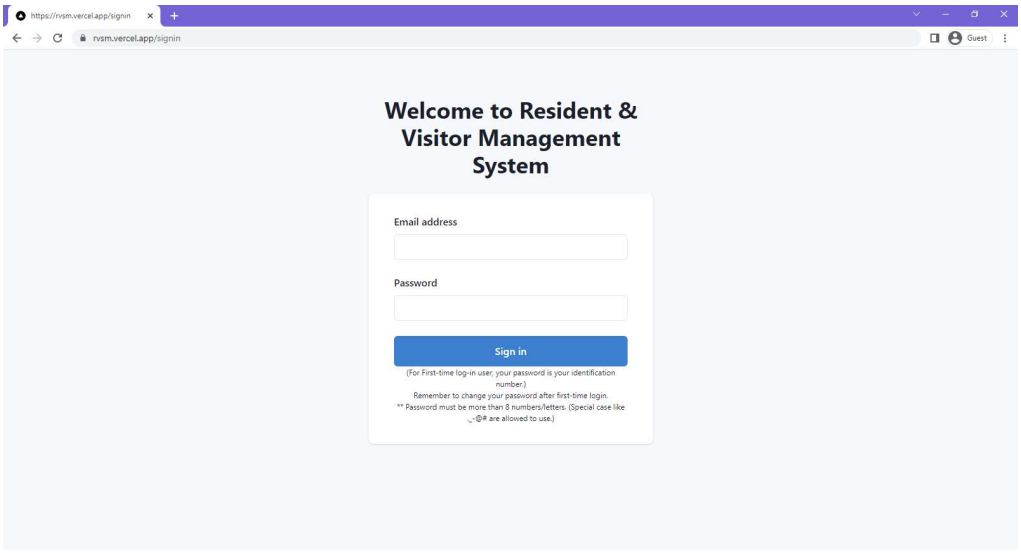


Figure 6.4 Show The Successfully Deployment of The Web Application

6.6 Summary

This chapter provides an overall concept of the front end and back end implementation of this project. These implementation includes implementation of supabase, authentication and authorization of the project, front end library used, and the deployment process.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

The test results for the project's testing are gathered in this chapter. Unit testing, integration testing, and user acceptance testing are all tested. Additionally, this chapter displays a traceability matrix connecting use cases, functional requirements, and test cases.

7.2 Traceability between Use Cases, Functional Requirements and Test Cases

Software testing is an essential stage in the development of software which assists in determining if the actual outcomes correspond to the expected result. Additionally, it guarantees that the functional requirements have complied with the stated objectives and permits the stakeholders to assess the application's quality. Traceability matrices are built as a connection to manage the relationship between functional requirements, test cases, and use cases in order to have a better sight to keep track of each function and test.

7.2.1 Use Case Table

Table 7.1 Use Cases

Use Case ID	Use Case Name	System
UC001	Login Account	Web application
UC002	Manage Resident's Registration	Web application
UC003	Track Resident's Information	Web application
UC004	Manage Administrators	Web application
UC005	Manage Security guards	Web application

UC006	Modify User Profile	Web application
UC007	Track Visitor's Records	Web application
UC008	Manage Announcements	Web application
UC009	Manage Feedback	Web application
UC010	Submit Registration form	Mobile application
UC011	Login Account	Mobile application
UC012	Modify User Account	Mobile application
UC013	Register Visitor	Mobile application
UC014	Manage Registration Visitation	Mobile application
UC015	Check-in Visitations	Mobile application
UC016	Verify Check-in Visitaitons	Mobile application
UC017	View Announcement	Mobile application
UC018	Manage Feedback	Mobile application

7.2.2 Functional Requirement Table

The functional requirement ID and their corresponding requirement specification statements are presented in Table 7.2.

Table 7.2 Functional requirements

Functional Requirement ID	**Functional Requirements**
W-1	The web application shall allow the management team to login their account.
W-2	The web application shall allow the management team to manage resident's registration.
W-3	The web application shall allow the management team to track resident's information.
W-4	The web application shall allow the management team to manage administrators.

W-5	The web application shall allow the management team to manage security guard.
W-6	The web application shall allow the management team to modify their user profile.
W-7	The web application shall allow the management team to track the visitor's records.
W-8	The web application shall allow the management team to manage announcements to all residents.
W-9	The web application shall allow the management team to manage feedback from residents.
M-1	The mobile application shall allow residents to submit registration form to the management team.
M-2	The mobile application shall allow residents and security guards to login their accounts.
M-3	The mobile application shall allow residents to modify their user profiles.
M-4	The mobile application shall allow residents and security guards to register for visitors.
M-5	The mobile application shall allow residents to manage their registered visitations.
M-6	The mobile application shall allow visitors to check in their visitations.
M-7	The mobile application shall allow security guards to verify check in visitations.
M-8	The mobile application shall allow residents to view announcements published by management teams.
M-9	The mobile application shall allow residents to manage feedback.

7.2.3 Test Cases Table of Unit Testing

One of the tests carried out for the project was unit testing. The components are all tested manually to make sure they all function properly.

7.2.3.1 Web application

Table 7.3 Unit testing test cases (web application)

Test Case ID	Test Case Name	Test Case Description	Related FR ID	Status
UTC001	Test login with correct credential	Examine whether the system will show login success	W-1	Pass
UTC002	Test login with incorrect credential	Examine whether the system will show error message		Pass
UTC003	Test retrieve all registration request	Examine whether all the registration request able to access from the client side	W-2	Pass
UTC004	Test update selected registration request	Examine whether the update approval button return an updated status message, or a reject status message		Pass
UTC005	Test retrieve all registered resident's information	Examine whether all the registered resident's information is accessible from the client side	W-3	Pass
UTC006	Test remove selected resident	Examine whether the remove message is shown when remove button is clicked		Pass
UTC007	Test retrieve all administrators	Examine whether all the administrator's information is accessible	W-4	Pass
UTC008	Test add administrators with valid input	Examine whether the successful add message will show after add button is pressed		Pass
UTC009	Test add administrators	Examine whether the error message will show		Pass

	with invalid inputs	after add button is pressed		
UTC010	Test remove administrators	Examine whether the successful remove message will show after the remove button pressed		Pass
UTC011	Test retrieve all security guards	Examine whether all the security guard's information is accessible	W-5	Pass
UTC012	Test add security guards with valid input	Examine whether the successful add message will show after add button is pressed		Pass
UTC013	Test add security guards with invalid inputs	Examine whether the error message will show after add button is pressed		Pass
UTC014	Test remove security guards	Examine whether the successful remove message will show after the remove button pressed		Pass
UTC015	Test modify user password with valid input	Examine whether the successful update message will show after update button is pressed	W-6	Pass
UTC016	Test modify user password with invalid input	Examine whether the error message will show after update button is pressed		Pass
UTC017	Test retrieve all visitation records	Examine whether all the visitation records' information is accessible	W-7	Pass
UTC018	Test retrieve all announcement	Examine whether all the announcement's information is accessible	W-8	Pass
UTC019	Test add new announcement with valid input	Examine whether the successful add message will show after add button is pressed		Pass
UTC020	Test add new announcement with invalid inputs	Examine whether the error message will show after add button is pressed		Pass

UTC021	Test update selected announcement with valid input	Examine whether the successful update message will show after update button is pressed		Pass
UTC022	Test update selected announcement with invalid input	Examine whether the error message will show after update button is pressed		Pass
UTC023	Test remove selected announcement	Examine whether the successful remove message will show after the remove button pressed		Pass
UTC024	Test retrieve all feedback	Examine whether all the feedback's information is accessible	W-9	Pass
UTC025	Test update selected feedback's reply with valid input	Examine whether the successful update message will show after the update button pressed		Pass
UTC026	Test update selected feedback's reply with invalid input	Examine whether the error message will show after the update button pressed		Pass

7.2.3.2 Mobile application

Table 7.4 Unit testing test cases (mobile application)

Test Case ID	Test Case Name	Test Case Description	Related FR ID	Status
UTC027	Test submit registration form with valid input	Examine whether the successful submit message will show after the submit button pressed	M-1	Pass
UTC028	Test submit registration form with invalid input	Examine whether the error message will show after the submit button pressed		Pass

UTC029	Test login with correct credential	Examine whether the system will show login success	M-2	Pass
UTC030	Test login with incorrect credential	Examine whether the system will show error message		Pass
UTC031	Test modify user password with valid input	Examine whether the successful update message will show after update button is pressed	M-3	Pass
UTC032	Test modify user password with invalid input	Examine whether the error message will show after update button is pressed		Pass
UTC033	Test add visitor with valid input	Examine whether the successful add message will show after add button is pressed	M-4	Pass
UTC034	Test add visitor with invalid input	Examine whether the error message will show after add button is pressed		Pass
UTC035	Test read all visitation	Examine whether all the visitation is accessible	M-5	Pass
UTC036	Test remove selected visitation	Examine whether the removed message will show after remove button is pressed		Pass
UTC037	Test check in visitation with valid input	Examine whether the system will show successful check in	M-6	Pass
UTC038	Test check in visitation with invalid input	Examine whether the system will show error message		Pass
UTC039	Test verify check-in visitation	Examine whether the system show success verify	M-7	Pass
UTC040	Test retrieve all announcements	Examine whether all the announcement is accessible	M-8	Pass
UTC041	Test add feedback with valid input	Examine whether the successful add message will show after add button is pressed	M-9	Pass
UTC042	Test add feedback with invalid input	Examine whether the error message will		Pass

		show after add button is pressed		
UTC043	Test retrieved all feedback	Examine whether all the feedback is accessible		Pass

7.2.4 Test Cases Table for Integration Testing

To ensure that modules can interact and communicate with one another correctly, integration tests examine interoperability between components. To make certain that all the data displayed are accurate and suitable, manual integration testing is done.

7.2.4.1 Web application

Table 7.5 Integration testing test cases (web application)

Test Case ID	Test Case Name	Test Case Description	Status
ITC001	Test login page	Examine whether the user can navigate to login page	Pass
ITC002	Test home page if user is authenticated	Examine whether the user can navigate to home page if user is authenticated	Pass
ITC003	Test home page if user is not authenticated	Examine whether the user is redirect to login page if the user is not authenticated.	Pass
ITC004	Test log out	Examine whether the user is removed from session and redirect to login page	Pass
ITC005	Test resident page	Examine whether the user can navigate to resident page after the user is authenticated	Pass
ITC006	Test update residents' registration's request	Examine whether the status of registration request is updated to database after validated	Pass
ITC007	Test remove resident's registration's request	Examine whether the resident's registration information is removed from database	Pass

ITC008	Test remove resident's information	Examine whether the resident's information is removed from database	Pass
ITC009	Test administrator page	Examine whether the user can navigate to add administrator page after the user is authenticated	Pass
ITC010	Test add administrator page	Examine whether the user can navigate to add administrator page after the user is authenticated	Pass
ITC011	Test add administrators page	Examine whether the administrator's information added to database after validated	Pass
ITC012	Test remove administrators	Examine whether the administrator's information is removed from database	Pass
ITC013	Test security guard page	Examine whether the user can navigate to security guard page after the user is authenticated	Pass
ITC014	Test add security guards page	Examine whether the user can navigate to add security guard page after the user is authenticated	Pass
ITC015	Test add security guards	Examine whether the security guard's information added to database after validated	Pass
ITC016	Test remove security guards	Examine whether the security guard's information is removed from database	Pass
ITC017	Test user profile page	Examine whether the user can navigate to user profile page after the user is authenticated	Pass
ITC018	Test update password	Examine whether the new password is updated to the backend	Pass
ITC019	Test visitation page	Examine whether the user can navigate to visitation page after the user is authenticated	Pass
ITC020	Test Announcement page	Examine whether the user can navigate to announcement page after the user is authenticated	Pass
ITC021	Test add announcement page	Examine whether the user can navigate to add announcement page after the user is authenticated	Pass
ITC022	Test add announcement	Examine whether the announcement's information added to database after validated	Pass
ITC023	Test update announcement	Examine whether the announcement's information updated to database after validated	Pass
ITC024	Test remove announcement	Examine whether the announcement's information is removed from database	Pass

ITC025	Test feedback page	Examine whether the user can navigate to feedback page after the user is authenticated	Pass
ITC026	Test update feedback reply	Examine whether the feedback's reply is updated to database after validated	Pass

7.2.4.2 Mobile application

Table 7.6 Integration testing test cases (mobile application)

Test Case ID	Test Case Name	Test Case Description	Status
ITC027	Test registration page	Examine whether the user can navigate to registration page	Pass
ITC028	Test submit registration	Examine whether the submission of registration request is added to database	Pass
ITC029	Test login resident page	Examine whether the user can navigate to resident's login page	Pass
ITC030	Test login security guard page	Examine whether the user can navigate to security guard's login page	Pass
ITC031	Test home page if user is authenticated	Examine whether the user can navigate to home page if user is authenticated	Pass
ITC032	Test log out	Examine whether the user is removed from session and redirect to login page	Pass
ITC033	Test user profile page	Examine whether the user can navigate to user profile page after the user is authenticated	Pass
ITC034	Test update password	Examine whether the new password is updated to the backend	Pass
ITC035	Test add visitors page	Examine whether the user can navigate to add visitors page after the user is authenticated	Pass
ITC036	Test add visitors	Examine whether the visitation's information is added to database after validated	Pass
ITC037	Test visitation page	Examine whether the user can navigate to visitation page after the user is authenticated	Pass
ITC038	Test remove visitation	Examine whether the visitation's information is removed from database	Pass
ITC039	Test check in visitation page	Examine whether the user can navigate to check in visitation page	Pass

ITC040	Test verify check in visitation page	Examine whether the user can navigate to verify check in visitation page	Pass
ITC041	Test announcements page	Examine whether the user can navigate to announcement page after the user is authenticated	Pass
ITC042	Test feedback page	Examine whether the user can navigate to feedback page after the user is authenticated	Pass
ITC043	Test add feedback page	Examine whether the user can navigate to feedback page after the user is authenticated	Pass
ITC044	Test add feedback	Examine whether the feedback is added to database after validated	Pass

7.2.5 Tracebility Matrix

For readers to comprehend the relationship between the testing done, the functional requirements, and the use cases, traceability matrices were provided. The traceability matrices for all the tests that were performed in accordance with the functional requirements and use case described in the chapter are shown in Table 7.7.

Table 7.7 Tracibility matrices

Test Cases ID	**Functional Requirement ID**	**Use Case ID**
UTC001, UTC002	W-1	UC001
UTC003, UTC004	W-2	UC002
UTC005, UTC006	W-3	UC003
UTC007, UTC008, UTC009, UTC010	W-4	UC004
UTC011, UTC012, UTC013, UTC014	W-5	UC005
UTC015, UTC016	W-6	UC006
UTC017	W-7	UC007

UTC018, UTC019, UTC020, UTC021, UTC022, UTC023	W-8	UC008
UTC024, UTC025	W-9	UC009
UTC027, UTC028	M-1	UC010
UTC029, UTC030	M-2	UC011
UTC031, UTC032	M-3	UC012
UTC033, UTC034	M-4	UC013
UTC035, UTC036	M-5	UC014
UTC037, UTC038	M-6	UC015
UTC039	M-7	UC016
UTC040	M-8	UC017
UTC041, UTC042, UTC043	M-9	UC018

7.3 User Acceptance Test

7.3.1 User Acceptance Test Plan

The management teams, residents, visitors, and security guards were the four user groups that participated in the user acceptance test. Residents, visitors, and security guards would be more concerned with the acceptance of the mobile application, whereas the management teams would be with the web application. As the information required for the evaluation required two different sets of questionnaires, feedback from several user groups was sought after.

It was initially intended to use think-aloud testing to determine acceptance of the web application and mobile application among various user groups. During the test, testers would be required to describe their thought and their action and reaction toward the system were observed and documented. . By adopting this technique, the user's emotions and acceptance is easier to capture when using the application. A total of 5 management team members of residential areas and a 10 residents of Cypress Condominium, 5 security guards and 10 visitors were reached out and invited to participate in the user acceptance testing. The testing was conducted physically in order to have a better observation of the testers' reactions. On the testing day, a laptop and a mobile phone are prepared for the testers to test the web application and mobile application. The testers were given a user acceptance test form consisted of all the test scenarios described in Appendix B. The test forms are also included in Appendix C. The testers were required to complete all the tests listed by following the test step given. When testers had completed all the test cases, they would be required to share their feedback on the Resident and Visitor Management System by answering the last section in the user acceptance form.

7.3.2 User Acceptance Test Cases

7.3.2.1 Web application

Table 7.8 UAT test cases (web application)

Test Case ID	Test Case Name	Test Description	Pass / Fail	Tested by (user group)	Related Functional ID
UATC001	Login account	To verify the account is logged in	Pass	Management Team	W-1
UATC002	Approve resident's registration	To verify the resident's registration is approved	Pass	Management Team	W-2
UATC003	Reject Resident's registration	To verify the resident's registration is rejected	Pass	Management Team	
UATC004	View all resident's information	To verify all the resident information is showed	Pass	Management Team	W-3
UATC005	View all Administrators	To verify all the administrators are showed	Pass	Management Team	W-4
UATC006	Add new administrator	To verify new administrator is added	Pass	Management Team	
UATC007	Remove administrators	To verify administrator is removed	Pass	Management Team	

UATC008	View all security guards	To verify all the security guards are showed	Pass	Management Team	W-5
UATC009	Add new security guard	To verify new security guard is added	Pass	Management Team	
UATC010	Remove security guard	To verify security guard is removed	Pass	Management Team	
UATC011	Modify new password	To verify new password can successfully login	Pass	Management Team	W-6
UATC012	View all visitation information	To verify all the visitation information is showed	Pass	Management Team	W-7
UATC013	Search visitation using unit id	To verify the related visitation is showed	Pass	Management Team	
UATC014	View all announcement	To verify all announcement is showed	Pass	Management Team	W-8
UATC015	Add and publish new announcement	To verify announcement is added and published	Pass	Management Team	
UATC016	Add draft announcement	To verify announcement is added and save as drafted	Pass	Management Team	
UATC017	Update and publish draft announcement	To verify draft announcement	Pass	Management Team	

		is updated and published			
UATC018	View announcement	To verify announcement detail is showed	Pass	Management Team	
UATC019	Remove announcement	To verify announcement is removed	Pass	Management Team	
UATC020	View all feedback	To verify all feedback is showed	Pass	Management Team	W-9
UATC021	Reply new feedback	To verify new feedback is replied	Pass	Management Team	
UATC022	View replied feedback	To verify replied feedback is showed	Pass	Management Team	

7.3.2.2 Mobile application

Table 7.9 UAT test cases (mobile application)

Test Case ID	**Test Case Name**	**Test Description**	**Pass / Fail**	**Tested by (user group)**	**Related Functional ID**
UATC023	Submit registration form	To verify the registration form is added	Pass	Resident	M-1
UATC024	Login account	To verify the account is logged in	Pass	Resident	M-2
UATC025	Login account	To verify the account is logged in	Pass	security guard	

UATC026	Logout	To verify the user is logged out	Pass	Resident	
UATC027	Logout	To verify the user is logged out	Pass	security guard	
UATC028	Modify user profile	To verify phone number, carplate, and password is updated and can successfully login after change the password	Pass	Resident	M-3
UATC029	Add visitor (residents)	To verify new visitors is added	Pass	Resident	M-4
UATC030	Add visitor (security guards)	To verify new visitors is added	Pass	Security guard	
UATC031	View all visitation	To verify all visitation is showed	Pass	Resident	M-5
UATC032	View visitation	To verify the visitation detail is showed	Pass	Resident	
UATC033	Remove upcoming visitation	To verify the visitation is removed	Pass	Resident	
UATC034	Check in visitation	To verify the visitation is successfully checked in	Pass	Visitor	M-6
UATC035	Verify check in visitation	To verify the check in	Pass	Security guard	M-7

		visitation is verified			
UATC036	View all announcement	To verify all announcement is showed	Pass	Resident	M-8
UATC037	View announcement	To verify announcement detail is showed	Pass	Resident	
UATC038	View all feedback	To verify all feedback is showed	Pass	Resident	M-9
UATC039	View feedback	To verify feedback detail is showed	Pass	Resident	
UATC040	Add new feedback	To verify feedback is added	Pass	Resident	

7.3.3 User Acceptance Test Result

Both sets of user acceptance test results and feedback were included in the appendix for reference. (Appendix D: User Acceptance Test Feedback Result)

7.3.3.1 Web Application

The user acceptance test for the web application was conducted within the management team user group. A total of 5 testers participated in this user acceptance test. The time to reach out to testers was limited, therefore, not enough testers are found or willing to participate in the user acceptance test. There were 2 testers aged from 20 to 30, 2 testers aged from 31 to 40 and 1 tester aged from 41 to 50. 10 questions were prepared in the form to collect the testers' feedback after the user acceptance test. The user acceptance result summary was listed in Table 7.10.

*1 = Strongly Disagree, 5 = Strongly Agree

Table 7.10 UAT feedback result summary (web application)

No	Question	Average Rating (1 -5)
Q1	This application will help you to facilitate communication with residents.	3.6
Q2	This application will help you easier to manage resident's information.	4.2
Q3	This application will help you easier to manage visitor's information	4.2
Q4	This application will help you easier to publish announcement.	4.8
Q5	This application will help you easier to manage resident's feedback.	3.8
Q6	This application will help you easier to manage security teams information.	4.4
Q7	This application will help you to simply administrative work.	4.2
Q8	Rate the overall user interface design	4.2
Q9	Rate the accuracy of data in the application.	4.6
Q10	Rate the satisfaction level in the application.	4.4

Table 7.11 UAT feedback result summary (web application)

No	Question	Sign off
Q11	As a user of the web application, would you sign off the user acceptance test?	5 – yes 0 - no

According to the table above, most of the questions in the user acceptance test scored around 4 to 5. However, question 1 which asked the testers if the application will help them to facilitate communication with residents only scored 3.6 out of 5. The reason being is one of the testers responded that more replies should be allowed for 1 feedback when conducting the user acceptance test with test case ID = UATC019 (reply new feedback). Generally, the overall feedback received from the testers was positive. This demonstrated that the system could successfully assist the testers in resolving their issues. With the score of 4.4 out of 5 on the question of rating satisfaction level in the application and all the testers are willing to sign off the test, it is considered that the web application would be accepted by the majority of management teams.

7.3.3.2 Mobile Application

The user acceptance test for the mobile application was conducted with 3 different types of target users which are residents. security guards, and visitors. The number of participants and age group of each target user group are listed in Table 7.12 and Table 7.13.

Table 7.12 UAT participants user type

Target user type	**Number of participants**
Residents	10
Visitors	10
Security guards	5

Table 7.13 UAT participants age group

Age group	**Number of participants**
20-30	10
31-40	7
41-50	5
>50	3

A total of 25 testers have participated in the user acceptance test for the mobile application of the Resident and Visitor Management System. They were 10 residents, 10 visitors, and 5 security guards. The number of testers in each targeted user type is inconsistent as the time to reach out to testers was limited, therefore, not enough testers in the target users group would happen. In this user acceptance test, 10 of 25 testers were aged from 20 to 30, 7 out of 25 respondents, 5 out of 25 respondents, and 3 out of 25 respondents are found according to the Table 7.13. As the Resident and Visitor Management System's main objective is to facilitate the communication between the management teams and residents and also the project focused 50% on management teams, 30% on residents, 10% on visitors, and 10% on security guards, therefore, the user acceptance test for visitors and security guards were conducted only with the functional testing without the feedback collection. The user acceptance test summary for mobile applications were listed in Table 7.14.

*1 = Strongly Disagree, 5 = Strongly Agree

Table 7.14 UAT Feedback Result Summary (mobile application)

No	**Question**	**Average Rating (1 -5)**
Q1	This application will help you easier to register your visitor's information.	4.2
Q2	This application will help you to manage your visitor's information.	4.2
Q3	This application will help you easier to get the latest information from management team.	4.1
Q4	This application will help you easier to report your issues to the management team.	4.3
Q5	This application will help you to facilitate communication with management teams.	4.0
Q6	Rate the overall user interface design	4.0

Q7	Rate the accuracy of data in the application.	4.2
Q8	Rate the satisfaction level in the application.	4.1

Table 7.15 UAT Feedback Result Summary (mobile application)

No	**Question**	**Sign off**
Q11	As a user of the web application, would you sign off the user acceptance test?	25 – yes 0 - no

According to the table above, most of the questions in the user acceptance test scored above 4.0 out of 5.0. This demonstrated that the system could successfully assist the testers in resolving their issues. With a score of 4.1 out of 5 in the question of rating satisfaction level in the application and all testers are willing to sign off the test, it is considered that the mobile application would be accepted by the majority of the residents.

7.4 Summary

This chapter provides the results of different types of tests. The positive result of the unit test and integration test shows the application is complete, functional and operable. For the user acceptance test, the average results were positive and both the web application and mobile application of the resident and visitor management system are accepted by the users.

CHAPTER 8

CONCLUSION AND RECOMMEDATIONS

8.1 Introduction

The project was begun in January 2022 and took about seven months to complete. The planning phase of this project, which lasted around 2 months, began with research into the problems related to the project, the definition of objectives, and the gathering of user requirements. Next, analysing the collected information and create solution based on the problem discovered happened in the analysis and design phase. All of the findings are documented in the first four chapters. The development phase of the system has started in May 2022 after approval of the proposal. The whole development phase was completed around three months and all tests were finished within two weeks' time. The system design, system implementation and system testing were also discussed in Chapters 5, Chapter 6, and Chapter 7. The project was closed in September 2022. In this chapter, the fulfilment of the objectives, limitations of the project, and also recommendations for future improvement were discussed.

8.2 Objective Examination

All three objectives defined in the project as shown in the list below were accomplished successfully with the implemented application.

1. To create a web application to simplify administrative work in management team
2. To develop a mobile application to facilitate the communication between the management teams and residents
3. To evaluate the mobile application and web application by conducting user acceptance test

The first objective was achieved as the web application has been integrated successfully by helping the management teams digitalise the whole

administration process. Instead of dealing with all the messy documents, the web application shows all the residents, visitors, announcements, and feedback information in the basic user interface which is easily understood by the users. It also could successfully help management teams to handle all the registration requests and manage the resident, visitors, management team members (administrators), security guards, announcements, and feedback information. The second objective was achieved with the implementation of the announcement broadcasting module and feedback module. The announcement broadcasting module created a channel that allows management teams to publish a digital announcement to all residents with just one click. Besides, the feedback module implemented also allows residents to enter their feedback and categories. Those feedbacks will be categorized and shown on the management side which also decreases the response time to urgent feedback or complaint. Thus, it facilitates two-way communication between management teams and residents. Lastly, the third objective was accomplished with the help of target users by conducting a user acceptance test after the development phase ended. The user acceptance of all target users received is almost positive. The average rate of acceptance level is around 4 out of 5 which showed both the web application and the mobile application are accepted by the target user group.

8.3 Limitations

The project's main objectives were met, but despite this, a number of system and project limitations were discovered. The first limitation was the limited notification of updates feature. The web and mobile applications are not able to instantly notify users of the latest updates through email, SMS or push notifications. The updates can only be viewed by the users when they are using the web or mobile application. Therefore, this causes the users to have to constantly access the web and mobile applications to check for any new updates notification.

The second limitation was found in the visitation module. The current version of the mobile application only allows visitors to check in using visitation

id and unit id and generate a QR code for the visitors. However, memorising the visitation id and unit id could be tedious for visitors although the main purpose of this setup is to ensure the visitor is the person who is authorised by the residents. Besides, the current version of the application only records the check-in date and check-in time of the visitors but not the check-out date and time. This may affect the completeness of the visitation data and extends the time the management teams spend sifting through the useful visitation data.

The third limitation was the limited reply to the feedback. The current version of both application only supports one reply for each feedback. However, as commented by the user acceptance test tester, sometimes, one reply is not enough for one feedback as the management teams would need to update the follow-up action to the residents for the feedback.

The fourth limitation was the system was only available in English. However, the users of the system may come from different backgrounds and may only know specific languages such as Malay, Mandarin or Cantonese. Therefore, the language used in this system may not be fully understood by the users and causing the users cannot fully utilise the system.

8.4 Recommendation for Future Work

The Resident and Visitor Management System still has a lot of space for improvement in light of the limitations highlighted above. Therefore, there are some recommendations for future improvements listed in Table 8.1. These recommendations may be valuable for future developers, but the enhancements are not limited to the item listed, they may find out more about this application and suggest more enhancements to be made.

Table 8.1 Recommendations

No	Recommendation	Description
1	Live Notification Update	In order to offer users an immediate notification as soon as possible, the present system might integrate live notification updates. There are numerous channels that can be used, including email, SMS, and mobile push notifications.
2	Record Visitation Check Out Date and Time	The current system only records the check in date and time. To increase the accuracy of data, check out date and time should be recorded by security guards.
3	Scan IC to confirm the visitation	The current mobile application requires visitors to enter visitation id and unit id in order to generate QR code for visitation. The better approach is to implement a feature for security guard to scan the identification card of the visitors, retrieve and analyse the identification card automatically. By this feature, the mobile application could ensure the person who use the QR is the person who invited by the resident.
4	Implement AI Chatbot and realtime chat	As the current system only support one reply per feedback, sometimes the residents feedback could not be answered by only one reply. Therefore, an AI chatbot could be implemented to answer more frequent asked question. Additionally, a realtime chat could also be implemented to allow management team to answer residents feedback instantly.

5	More Language Support	English are the only languages used in the current system. The application could be translated to different language that mostly used by the users such as Malay, Mandarin or Hindi.
6	Implement payment feature	As a system that used to assist the management operation, payment feature is the feature that most wanted by the respondents when collecting the requirements. Therefore, the application could implement payment features to allow residents pay their management fees and allow management team to check the the bill.

REFERENCES

Al-Hurmuzi, S., Al-Khanjari, Z., AI-Kindi, I., 2018, *Proposed Feasible PEF framework for User Acceptance Testing*. In: Faculty of Information Technology Applied Science Private University, 8th International Conference on Computer Science and Information Technology (CSIT). Amman, Jordan, 11-12 July 2018, New York: IEEE.

Biorn-Hansen, A., Gronli, T., Ghinea, G., Alouneh, S., 2018, An Empirical Study of Cross-Platform Mobile Development in Industry, *Wireless Communications and Mobile Computing*, [online] Available at: <https://www.hindawi.com/journals/wcmc/2019/5743892/ > [Accessed 19 March 2022].

Bryan A. Weber, Hossein Yarandi, Meredeth A. Rowe, Justus P. Weber,, 2005. A comparison study: paper-based versus web-based data collection and management. *Applied Nursing Research*, [e-journal] 18(3), pp. 182-185. https://doi.org/10.1016/j.apnr.2004.11.003.

Chra, J., 2021. Comparison of Technologies for Multiplatform Mobile Applications Development. Master. MASARYK UNIVERSITY. Available at: < https://theses.cz/id/3urd00/> [Accessed 12 March 2022].

Dennis, A., Wixom, B.H. & Tegarden, D.P., 2020. *Systems Analysis & Design: An object-oriented approach with UML*, Hoboken, NJ: Wiley.

Eaglesflight, 2022. *Teamwork in the Workplace: Frequent and Effective Communication*. [online] Eagle's Flight. Available at: <https://www.eaglesflight.com/resource/teamwork-in-the-workplace-frequent-and-effective-communication/> [Accessed 11 February 2022].

Ericsson, K. and Simon, H., 2022. *APA PsycNet*. [online] Psycnet.apa.org. Available at: <https://psycnet.apa.org/record/1985-97337-000> [Accessed 25 March 2022].

Government of Malaysia, 2019. Shared Prosperity Vision 2030. [online] Government of Malaysia. Available at: <https://www.pmo.gov.my/wp-content/uploads/2019/10/SPV2030-summary-en.pdf> [Accessed 17 March 2022].

Ichsani, Y., 2018, Usability Performance Evaluation of Information System with Concurrent Think-Aloud Method as User Acceptance Testing: A Literature Review, *Advances in Intelligent Systems Research (AISR)*, [online] Available at: < https://www.researchgate.net/publication/325563123_Usability_Performance_Evaluation_of_Information_System_with_Concurrent_Think-Aloud_Method_as_User_Acceptance_Testing_A_Literature_Review > [Accessed 18 March 2022].

Jung, M., Youn, A., Bae, J., Choi, Y., 2015, *A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment*. In: Yanchun Zhang and Alfredo Cuzzocrea, 8th International Conference on Computer Science and Information Technology (CSIT). Jeju, South Korea, 25-28 November 2015, New York: IEEE.

Kumparak, G., 2022. *TechCrunch is part of the Yahoo family of brands*. [online] Techcrunch.com. Available at: <https://techcrunch.com/2021/09/09/supabase-raises-30m-for-its-open-source-insta-backend/> [Accessed 25 March 2022].

Lindberg, T., 2018. *A/B-testing for web design: A comparative study of response times between MySQL and PostgreSQL*. Bachelor. University of Skövde. Available at: < https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1215120&dswid=3922> [Accessed 17 March 2022].

Marijan, B., 2022. *12 Best Cloud Databases and Why Use Them | phoenixNAP KB*. [online] Knowledge Base by phoenixNAP. Available at: <https://phoenixnap.com/kb/cloud-database> [Accessed 26 March 2022].

Ohyver, M., Moniaga, J.V., Sungkawa, I., Subagyo, B.E., Chandra, I.A., 2019, The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test, *Procedia Computer Science*, [online] Available at: <https://www.sciencedirect.com/science/article/pii/S1877050919311500> [Accessed 19 March 2022].

Olubisi, I., 2022. *How to Build a Full Stack Application With Supabase, React, and Tailwind CSS in Nextjs*. [online] freeCodeCamp.org. Available at: <https://www.freecodecamp.org/news/how-to-build-a-full-stack-application-with-tailwind-css-and-supabase-in-nextjs/> [Accessed 27 March 2022].

QuestionPro. 2022. *Quantitative Research: Definition, Methods, Types and Examples | QuestionPro*. [online] Available at: <https://www.questionpro.com/blog/quantitative-research/> [Accessed 18 February 2022].

Rawat, P., Mahajan A.N., 2020, ReactJS: A Modern Web Development Framework, International Journal of Innovative Science and Research Technology, [online] Available at: < https://ijisrt.com/assets/upload/files/IJISRT20NOV485.pdf > [Accessed 20 March 2022].

Ross, P., 2022, Under the hood: Architecture and Technology Stack of Supabase. [online] workingsoftware.dev. Available at: < https://www.workingsoftware.dev/tech-stack-and-architecture-of-supabase/ > [Accessed 20 August 2022].

Salim, F., 2022. *Using Supabase as an Alternative to Firebase in Flutter*. [online] Engineering Education (EngEd) Program | Section. Available at: <https://www.section.io/engineering-education/using-supabase-as-an-alternative-to-firebase-in-flutter/> [Accessed 29 March 2022].

Shareef, T.H., Shareef, K.H., Rashid, B.N., 2022, A Survey of Comparing Different Cloud Database Performance: SQL and NoSQL, Passer Journal, [online] Available at: < https://passer.garmian.edu.krd/article_144858.html > [Accessed 20 March 2022].

Sheldon, R., 2022. *How to choose between SQL and NoSQL databases - Simple Talk*. [online] Simple Talk. Available at: <https://www.red-gate.com/simple-talk/databases/nosql/how-to-choose-between-sql-and-nosql-databases/> [Accessed 27 March 2022].

Stack Overflow, 2022. *Stack Overflow Developer Survey 2021*. [online] Stack Overflow. Available at: <https://insights.stackoverflow.com/survey/2021> [Accessed 26 March 2022].

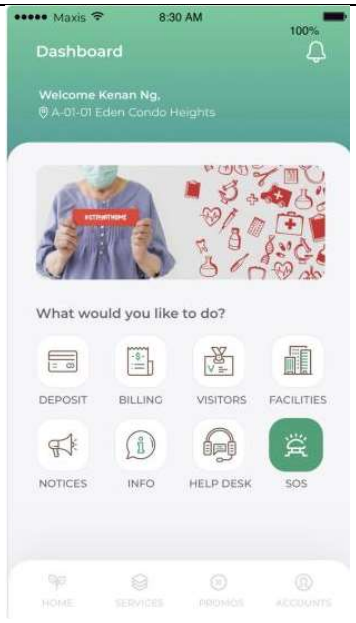
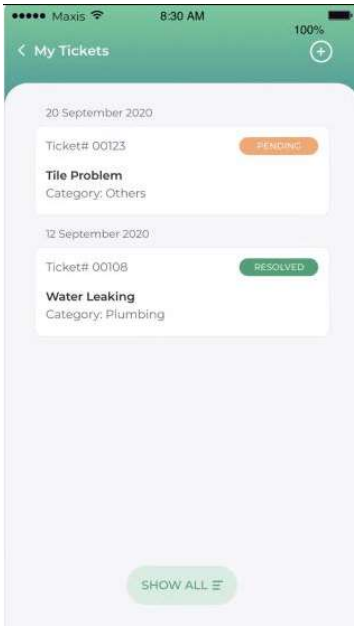
Sualim, S.A., Yassin, M.N., Mohamad, R., 2016, Comparative Evaluation of Automated User Acceptance Testing Tool for Web Based Application, *International Journal of Software Engineering and Technology*, [online] Available at: < https://www.researchgate.net/profile/Radziah-Mohamad/publication/313208880_Comparative_Evaluation_of_Automated_User_Acceptance_Testing_Tool_for_Web_Based_Application/links/58928bbba6fdcc1b4146c5f1/Comparative-Evaluation-of-Automated-User-Acceptance-Testing-Tool-for-Web-Based-Application.pdf> [Accessed 16 March 2022].

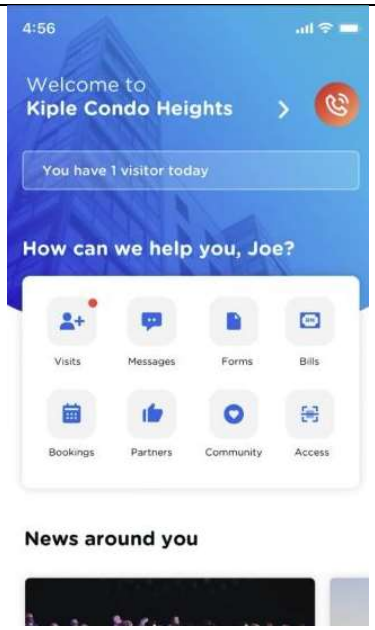
Sukamolson, S., 2007. Fundamentals of quantitative research. *Language Institute Chulalongkorn University*, *1*(3), pp.1-20., [online] Available at: < https://www.researchgate.net/publication/242772176_Fundamentals_of_quantitative_research> [Accessed 19 February 2022].

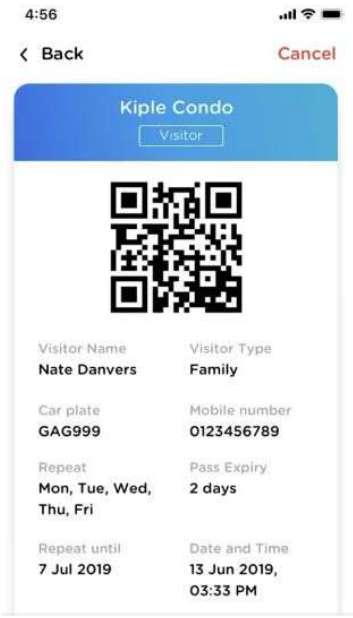
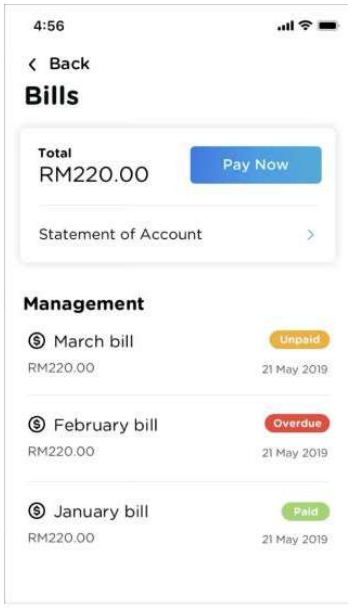
Wurmser, Y., 2022. *Apps Far Outpace Browsers in US Adults' Mobile Time Spent*. [online] Insider Intelligence. Available at: <https://www.emarketer.com/content/the-majority-of-americans-mobile-time-spent-takes-place-in-apps> [Accessed 14 February 2022]

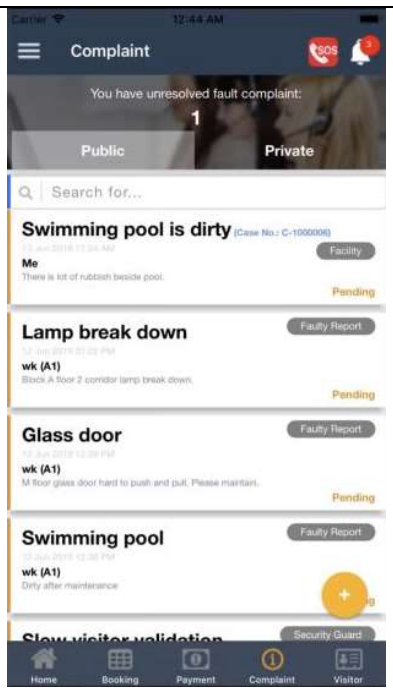
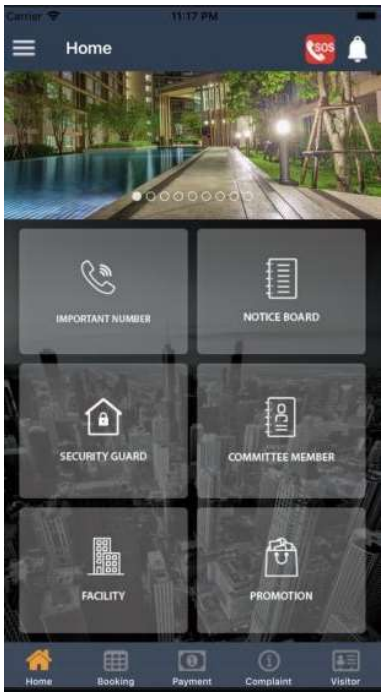
APPENDICES

APPENDIX A: User Review of Similar System Retrieved from Application Store

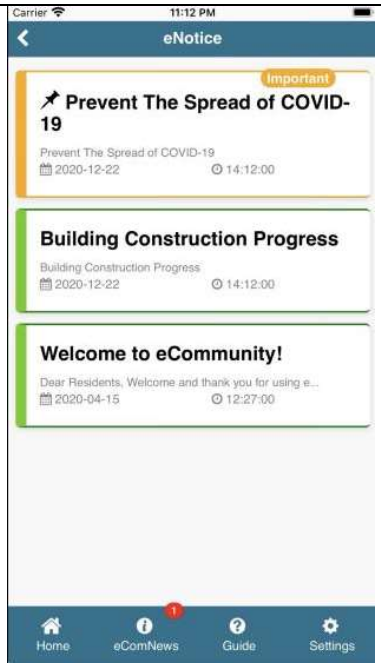
Application	Screenshots	User Review
Eden Community App	 	Review 1: More user guide should be provided Review 2: The login feature need to be improved. Review 3: Bad UI and UX experience. The UI elements are all over the place and overlaps on buttons.

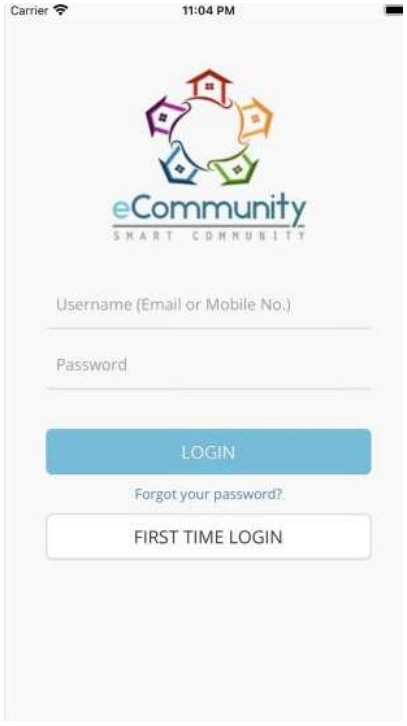
		
kipleLive		Review 1: Cannot lodge report through message tab mentioned in the guide Review 2: The new updates unable to support the book facitlites functions. Review 3: The application always keep user log out for no reason. Review 4: The applications collects too much information for visitors, user suggest to

	 	only collect visitor's phone number.

M4U Home System	 	Review 1: Advice on registration should be provided Review 2: The text is not visible in dark mode. Review 3: The application is hardly to login and proper instruction is not provided.

		
eCommunity		Review 1: The application is very convenience to use, straight froward and have simple user interface. Review 2: Most user friendly apps ever use compare to other apps. Review 3: Provide many useful functions to help in manage unit affairs


Carrier
11:12 PM
eNotice
Important
Prevent The Spread of COVID-19
Prevent The Spread of COVID-19
2020-12-22
14:12:00
Building Construction Progress
Building Construction Progress
2020-12-22
14:12:00
Welcome to eCommunity!
Dear Residents, Welcome and thank you for using e...
2020-04-15
12:27:00
Home
eComNews
Guide
Settings


Carrier
11:04 PM
eCommunity
SMART COMMUNITY
Username (Email or Mobile No.)
Password
LOGIN
Forgot your password?
FIRST TIME LOGIN

APPENDIX B: Test Cases

Unit Test (web application)

Test Case ID	UTC001	Test Name	Test login with correct credential	Pass/Fail	Pass
Test Case Description		Examine whether the system will show login success			
Test Case Scenario				Test Data	
1. User enters the valid and correct email and password to. 2. User presses the login button.				Email: admin@gmail.com Password: 123456789	
Expected Result				Actual Result	
The system redirects to home page.				The system redirects to home page.	

Test Case ID	UTC002	Test Name	Test login with incorrect credential	Pass/Fail	Pass
Test Case Description		Examine whether the system will show error message when invalid input is entered.			
Test Case Scenario				Test Data	
1. User enters the invalid email and password. 2. User presses the login button.				Email: abc@g Password: 123456	
Expected Result				Actual Result	
The error message prompts the user entered the invalid input.				The error message prompts the user entered the invalid input.	

Test Case ID	UTC003	Test Name	Test retrieve all registration request	Pass/Fail	Pass
Test Case Description		Examine whether all the registration request able to access from the client side			
Test Case Scenario				Test Data	
1. User enters the residents page and clicks on the new registration tab.				-	
Expected Result				Actual Result	
All the new registration request are displayed.				All the new registration results are displayed.	

Test Case ID	UTC004	Test Name	Test update selected registration request	Pass/Fail	Pass
Test Case Description		Examine whether the update approval button return an updated status message, or a reject status message			
Test Case Scenario				Test Data	
1. User clicks on the approve button on the selected registration's request. OR 1. User clicks on the reject button on the selected registration's request.				-	
Expected Result				Actual Result	
The status message will be prompt to tell the user the status is updated.				The status message will be prompt to tell the user the status is updated.	

Test Case ID	UTC005	Test Name	Test retrieve all registered resident's information	Pass/Fail	Pass
Test Case Description		Examine whether all the registered resident's information is accessible from the client side			
Test Case Scenario				Test Data	
1. User enters the residents page and clicks on the existing resident tab.				-	
Expected Result				Actual Result	
All the existing resident's information are displayed.				All the existing resident's information are displayed.	

Test Case ID	UTC006	Test Name	Test remove selected resident	Pass/Fail	Pass
Test Case Description		Examine whether the remove message is shown when remove button is clicked.			
Test Case Scenario				Test Data	
1. User clicks on the remove button on the selected residents.				-	
Expected Result				Actual Result	
The status message will be prompt to tell the user the residents is removed.				The status message will be prompt to tell the user the residents is removed.	

Test Case ID	UTC007	Test Name	Test retrieve all administrators	Pass/Fail	Pass
Test Case Description		Examine whether all the administrator's information is accessible			
Test Case Scenario				Test Data	
1. User enters the administrator's page.				-	
Expected Result				Actual Result	
All the administrator's information is displayed.				All the administrator's information is displayed.	

Test Case ID	UTC008	Test Name	Test add administrators with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful add message will show after add button is pressed			
Test Case Scenario				Test Data	
1. User enters the valid name, identification card number, phone number and email. 2. User clicks on the add button				Name: Yann IC: 880215016231 Phone number: 60123568745 Email: yann@gmail.com	
Expected Result				Actual Result	
The success added message is prompted.				The success added message is prompted.	

Test Case ID	UTC009	Test Name	Test add administrators with invalid inputs	Pass/Fail	Pass
Test Case Description	Examine whether the error message will show after add button is pressed				
Test Case Scenario				Test Data	
1. User enters the invalid name, identification card number, phone number and email. 2. User clicks on the add button				Name: Yann23 IC: 880215 Phone number: 012356874 Email: yann.csof	
Expected Result				Actual Result	
The error message is prompted.				The error message is prompted.	

Test Case ID	UTC010	Test Name	Test remove administrators	Pass/Fail	Pass
Test Case Description	Examine whether the successful remove message will show after the remove button pressed				
Test Case Scenario				Test Data	
1. User clicks on the remove button on the selected administrator.				-	
Expected Result				Actual Result	
The removed message will be prompt to tell the user the administrator is removed.				The removed message will be prompt to tell the user the administrator is removed.	

Test Case ID	UTC011	Test Name	Test retrieve all security guards	Pass/Fail	Pass
Test Case Description		Examine whether all the security guard's information is accessible			
Test Case Scenario				Test Data	
1. User enters the security guard's page.				-	
Expected Result				Actual Result	
All the security guard's information is displayed.				All the security guard's information is displayed.	

Test Case ID	UTC012	Test Name	Test add security guards with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful add message will show after add button is pressed			
Test Case Scenario				Test Data	
1. Users enters valid name, identification card number, phone number, and email. 2. Users clicks on the add security guard button.				Name: Timothy IC: 850512013265 Phone number: 60123485742 Email: timothy@gmail.com	
Expected Result				Actual Result	
The status message will be prompt to tell the user the security guard is added.				The status message will be prompt to tell the user the security guard is added.	

Test Case ID	UTC013	Test Name	Test add security guards with invalid inputs	Pass/Fail	Pass
Test Case Description		Examine whether the error message will show after add button is pressed			
Test Case Scenario				Test Data	
1. Users enters invalid name, identification card number, phone number, and email. 2. Users clicks on the add security guard button.				Name: Timothy452 IC: 850512013 Phone number: 012-348574 Email: timothy@gma	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt to tell the user the security guard is added.	

Test Case ID	UTC014	Test Name	Test remove security guards	Pass/Fail	Pass
Test Case Description		Examine whether the successful remove message will show after the remove button pressed			
Test Case Scenario				Test Data	
1. Users clicks on the remove button on the selected security guard.				-	
Expected Result				Actual Result	
The removed message will be prompt to tell the user the security guard is removed.				The removed message will be prompt to tell the user the security guard is removed.	

Test Case ID	UTC015	Test Name	Test modify user password with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful update message will show after update button is pressed			
Test Case Scenario				Test Data	
1. User enters the new password in valid format(more than 8 numbers or letters). 2. User clicks on the update button.				Password: 0123456789	
Expected Result				Actual Result	
The successful update message will be prompt to tell the user the password is updated.				The successful update message will be prompt to tell the user the password is updated.	

Test Case ID	UTC016	Test Name	Test modify user password with invalid input	Pass/Fail	Pass
Test Case Description		Examine whether the error message will show after update button is pressed			
Test Case Scenario				Test Data	
1. User enters the new password in invalid format. 2. User clicks on the update button.				Password: sdfds	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt to tell the user the password is updated.	

Test Case ID	UTC017	Test Name	Test retrieve all visitation records	Pass/Fail	Pass
Test Case Description		Examine whether all the visitation records' information is accessible			
Test Case Scenario				Test Data	
1. User enters the visitation page and clicks on the total visitation tab.				-	
Expected Result				Actual Result	
All the visitation information are displayed.				All the visitation information are displayed.	

Test Case ID	UTC018	Test Name	Test retrieve all announcement	Pass/Fail	Pass
Test Case Description		Examine whether all the announcement's information is accessible			
Test Case Scenario				Test Data	
1. User enters the announcement page.				-	
Expected Result				Actual Result	
All the announcement information are displayed.				All the announcement information are displayed.	

Test Case ID	UTC019	Test Name	Test add new announcement with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful add message will show after add button is pressed			
Test Case Scenario				Test Data	
1. Users enters valid title, description, and upload image. 2. Users clicks on the save and publish button.				Title: Updated Lift Maintenance Time Description: The lift maintenance time is changed to 19 Aug 2022. Please be noticed that the lift is not functionable during the period. Image: lift.png	
Expected Result				Actual Result	
The successful add message will be prompt to tell the user the announcement is published.				The successful add message will be prompt to tell the user the announcement is published.	

Test Case ID	UTC020	Test Name	Test add new announcement with invalid inputs	Pass/Fail	Pass
Test Case Description		Examine whether the successful add message will show after add button is pressed			
Test Case Scenario				Test Data	
1. Users enters invalid title, description, and upload image. 2. Users clicks on the save and publish button.				Title: “” Description: “” Image: null	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC021	Test Name	Test update selected announcement with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful update message will show after update button is pressed			
Test Case Scenario				Test Data	
1. Users enters valid title, description, and upload image. 2. Users clicks on the update and publish button.				Title: Yoga class Description: Feel free to join the yoga class. Contact us at: xxx Image: yoga.png	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC022	Test Name	Test update selected announcement with invalid input	Pass/Fail	Pass
Test Case Description		Examine whether the error message will show after update button is pressed			
Test Case Scenario				Test Data	
1. Users enters invalid title, description, and upload image. 2. Users clicks on the save and publish button.				Title: “” Description: “” Image: null	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC023	Test Name	Test remove selected announcement	Pass/Fail	Pass
Test Case Description		Examine whether the successful remove message will show after the remove button pressed			
Test Case Scenario				Test Data	
1. Users clicks on the remove button on the selected announcement				-	
Expected Result				Actual Result	
The removed message will be prompt to tell the user the announcement is removed.				The removed message will be prompt to tell the user the announcement is removed.	

Test Case ID	UTC024	Test Name	Test retrieve all feedback	Pass/Fail	Pass
Test Case Description		Examine whether all the feedback's information is accessible			
Test Case Scenario				Test Data	
1. User enters the feedback page.				-	
Expected Result				Actual Result	
All the feedback information are displayed.				The removed message will be prompt to tell the user the announcement is removed.	

Test Case ID	UTC025	Test Name	Test update selected feedback's reply with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful update message will show after the update button pressed			
Test Case Scenario				Test Data	
1. User enters valid reply message. 2. User clicks on send reply button.				Content: Thank you for raising out the issue. We will take action as soon as possible.	
Expected Result				Actual Result	
The successful update message will be prompt to tell the user the message is send.				The successful update message will be prompt to tell the user the message is send.	

Test Case ID	UTC026	Test Name	Test update selected feedback's reply with invalid input	Pass/Fail	Pass
Test Case Description		Examine whether the error message will show after the update button pressed			
Test Case Scenario				Test Data	
1. User enters invalid reply message. 2. User clicks on send reply button.				Content: ""	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Unit Test (mobile application)

Test Case ID	UTC027	Test Name	Test submit registration form with valid input	Pass/Fail	Pass
Test Case Description	Examine whether the successful submit message will show after the submit button pressed				
Test Case Scenario				Test Data	
1. User enters name, identification card number, phone number, email address, password, car plate number, unit id, address and upload supporting documents in valid format. 2. User clicks on the sign-up button.				Name: Julian Bob Identification number: 780214054652 Phone number: 60142541485 Email address: julianbob@gmail.com Password: 123456789 Car plate number: KSL1245 Unit ID: 1452 Address: 12-2-1, Cypress Condo, Jalan Sg Long, Kajang, Selangor. Supporting documents: waterbill.png	
Expected Result				Actual Result	
The successful submit message will be prompt to tell the registration is sent to the management team.				The successful submit message will be prompt to tell the registration is sent to the management team.	

Test Case ID	UTC028	Test Name	Test submit registration form with invalid input	Pass/Fail	Pass
Test Case Description		Examine whether the error message will show after the submit button pressed			
Test Case Scenario				Test Data	
1. User enters name, identification card number, phone number, email address, password, car plate number, unit id, address and upload supporting documents in invalid format. 2. User clicks on the sign-up button.				Name: dfd55 Identification number: 751554 Phone number: 012-5454521 Email: sdd@sd Password: 123456 Car plate number: 1254 Unit ID: “” Address: “” Supporting document: “”	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC029	Test Name	Test login with correct credential	Pass/Fail	Pass
Test Case Description		Examine whether the system will show login success			
Test Case Scenario				Test Data	
1. User enters valid and correct email address and password. 2. User clicks on enter login button.				Email: f@gmail.com Password: 123456789	
Expected Result				Actual Result	
The user will be redirect to home page.				The user will be redirect to home page.	

Test Case ID	UTC030	Test Name	Test login with incorrect credential	Pass/Fail	Pass
Test Case Description		Examine whether the system will show error message			
Test Case Scenario				Test Data	
1. User enters invalid email address and password. 2. User clicks on enter login button.				Email: ss@gmail. Password: 123456	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC031	Test Name	Test modify user password with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful update message will show after update button is pressed			
Test Case Scenario				Test Data	
1. User enters new password in valid format. 2. User clicks on the edit password button.				Password: felicia1234	
Expected Result				Actual Result	
The successful update message will be prompt.				The successful update message will be prompt.	

Test Case ID	UTC032	Test Name	Test modify user password with invalid input	Pass/Fail	Pass
Test Case Description		Examine whether the error message will show after update button is pressed			
Test Case Scenario				Test Data	
1. User enters new password in invalid format. 2. User clicks on the edit password button.				Password: sd#@$@	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC033	Test Name	Test add visitor with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the successful add message will show after add button is pressed			
Test Case Scenario				Test Data	
1. User enters visitor's name, visitor's identification number, phone number, date visit, time visit, car plate number in valid format. 2. Users clicks on the submit button.				Visitor's name: Liew Visitor's IC: 880430098754 Phone number: 60158756985 Date visit: 31-Aug-2022 Time visit: 12:05:12 Car plate number: JSD2556	
Expected Result				Actual Result	
The successful added message will be prompt.				The successful added message will be prompt.	

Test Case ID	UTC034	Test Name	Test add visitor with invalid input	Pass/Fail	Pass
Test Case Description	Examine whether the error message will show after add button is pressed				
Test Case Scenario				Test Data	
1. User enters visitor's name, visitor's identification number, phone number, date visit, time visit, car plate number in invalid format. 2. Users clicks on the submit button.				Visitor's name: 5343r Visitor's IC: 880430-09-8754 Phone number: 015 8756985 Date visit: 31-Aug-2022 Time visit: 12:05:12 Car plate number: 3343S	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC035	Test Name	Test read all visitation	Pass/Fail	Pass
Test Case Description	Examine whether all the visitation is accessible				
Test Case Scenario				Test Data	
1. User enters the visitation page.				-	
Expected Result				Actual Result	
All the visitation information are displayed.				All the visitation information are displayed.	

Test Case ID	UTC036	Test Name	Test remove selected visitation	Pass/Fail	Pass
Test Case Description		Examine whether the removed message will show after remove button is pressed			
Test Case Scenario				Test Data	
1. User clicks on the remove button on the selected visitation.				-	
Expected Result				Actual Result	
The removed message will be prompt.				The removed message will be prompt.	

Test Case ID	UTC037	Test Name	Test check in visitation with valid input	Pass/Fail	Pass
Test Case Description		Examine whether the system will show successful check in			
Test Case Scenario				Test Data	
1. User enters valid visitation id and unit id. 2. User clicks on the submit button.				Visitation id: CY8754333 Unit id: 875	
Expected Result				Actual Result	
The successful check in message will be shown.				The successful check in message will be shown.	

Test Case ID	UTC038	Test Name	Test check in visitation with invalid input	Pass/Fail	Pass
Test Case Description		Examine whether the system will show error message			
Test Case Scenario				Test Data	
1. User enters invalid visitation id and unit id. 2. User clicks on the submit button.				Visitation id: 8754333 Unit id: cy875	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC039	Test Name	Test verify check-in visitation	Pass/Fail	Pass
Test Case Description		Examine whether the system show success verify			
Test Case Scenario				Test Data	
1. User scan the qr code provided by visitors.				-	
Expected Result				Actual Result	
The successful update message is displayed				The error message will be prompt.	

Test Case ID	UTC040	Test Name	Test retrieve all announcements	Pass/Fail	Pass
Test Case Description	Examine whether all the announcement is accessible				
Test Case Scenario				Test Data	
1. User enters the announcement page.				-	
Expected Result				Actual Result	
All the announcement information is displayed.				All the announcement information is displayed.	

Test Case ID	UTC041	Test Name	Test add feedback with valid input	Pass/Fail	Pass
Test Case Description	Examine whether the successful add message will show after add button is pressed				
Test Case Scenario				Test Data	
1. User enters title, description and categories in valid format. 2. User clicks on the add button.				Title: Rooftop leaking Description: I found the rooftop leaking, Category: defeat of common area	
Expected Result				Actual Result	
The successful added message will be prompt.				The successful added message will be prompt.	

Test Case ID	UTC042	Test Name	Test add feedback with invalid input	Pass/Fail	Pass
Test Case Description		Examine whether the error message will show after add button is pressed			
Test Case Scenario				Test Data	
1. User enters title, description and categories in invalid format. 2. User clicks on the add button.				Title: “” Description: “” Category: “”	
Expected Result				Actual Result	
The error message will be prompt.				The error message will be prompt.	

Test Case ID	UTC043	Test Name	Test retrieved all feedback	Pass/Fail	Pass
Test Case Description		Examine whether all the feedback is accessible			
Test Case Scenario				Test Data	
1. User enters the feedback page.				-	
Expected Result				Actual Result	
All the feedback information is displayed.				All the feedback information is displayed.	

Integration Test (web application)

Test Case ID	ITC001	Test Name	Test login page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to login page			
Test Case Scenario				Test Data	
1. User enters the url or link provided to enter the login page.				-	
Expected Result				Actual Result	
User is able to access the login page.				User is able to access the login page.	

Test Case ID	ITC002	Test Name	Test home page if user is authenticated	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to home page if user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the home page in the side navigation bar.				-	
Expected Result				Actual Result	
User will be redirected to homepage of the application.				User will be redirected to homepage of the application.	

Test Case ID	ITC003	Test Name	Test home page if user is not authenticated	Pass/Fail	Pass
Test Case Description	Examine whether the user is redirect to login page if the user is not authenticated.				
Test Case Scenario				Test Data	
1. User clicks on the home page in the side navigation bar.				-	
Expected Result				Actual Result	
User will not be redirected to homepage of the application.				User will not be redirected to homepage of the application.	

Test Case ID	ITC004	Test Name	Test log out	Pass/Fail	Pass
Test Case Description	Examine whether the user is removed from session and redirect to login page				
Test Case Scenario				Test Data	
1. User clicks on the log out button.				-	
Expected Result				Actual Result	
User is redirected to login page.				User is redirected to login page.	

Test Case ID	ITC005	Test Name	Test resident page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to resident page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the resident's button on the side navigation bar.				-	
Expected Result				Actual Result	
User is navigated to the resident page after the user is authenticated.				User is navigated to the resident page after the user is authenticated.	

Test Case ID	ITC006	Test Name	Test update residents' registration's request	Pass/Fail	Pass
Test Case Description		Examine whether the status of registration request is updated to database after validated			
Test Case Scenario				Test Data	
1. User enters the resident page. 2. User clicks on approve or reject button on the selected resident's registration request.				-	
Expected Result				Actual Result	
The status of the resident's registration request updated in database. The residents information will remove from new registration tab and added to existing residents tab.				The status of the resident's registration request updated in database.	

Test Case ID	ITC007	Test Name	Test remove resident's registration's request	Pass/Fail	Pass
Test Case Description		Examine whether the resident's registration information is removed from database			
Test Case Scenario				Test Data	
1. User enters the resident page. 2. User clicks on reject button on the selected resident's registration request.				-	
Expected Result				Actual Result	
The registration request is removed from the database and will no longer show in resident's page.				The registration request is removed from the database and will no longer show in resident's page.	

Test Case ID	ITC008	Test Name	Test remove resident's information	Pass/Fail	Pass
Test Case Description		Examine whether the resident's information is removed from database			
Test Case Scenario				Test Data	
1. User enters the resident page. 2. User clicks on remove button on the selected resident's information				-	
Expected Result				Actual Result	
The resident's information is removed from database and no longer show in resident's page.				The resident's information is removed from database and no longer show in resident's page.	

Test Case ID	ITC009	Test Name	Test administrator page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to administrator page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the administrator's button in the side navigation bar.				-	
Expected Result				Actual Result	
User is navigated to the administrator page after the user is authenticated.				User is navigated to the administrator page after the user is authenticated.	

Test Case ID	ITC010	Test Name	Test add administrator page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to add administrator page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the administrator's button in the side navigation bar. 2. User is redirected to administrator's page. 3. User clicks on add administrator button				-	
Expected Result				Actual Result	
User is navigated to the add administrator page after the user is authenticated.				User is navigated to the add administrator page after the user is authenticated.	

Test Case ID	ITC011	Test Name	Test add administrators	Pass/Fail	Pass
Test Case Description		Examine whether the administrator's information added to database after validated			
Test Case Scenario				Test Data	
1. User enters the administrator's page. 2. User clicks on the add administrator button and redirected to add administrator page. 3. User enters the valid administrator's information. 4. User clicks on the add button				-	
Expected Result				Actual Result	
New administrator information added to the database and shown in the administrator's page.				New administrator information added to the database and shown in the administrator's page.	

Test Case ID	ITC012	Test Name	Test remove administrators	Pass/Fail	Pass
Test Case Description		Examine whether the administrator's information is removed from database			
Test Case Scenario				Test Data	
1. User enters the administrator's page. 2. User clicks on the remove button on the selected administrator.				-	
Expected Result				Actual Result	
The administrators will be removed from database and no longer show in the administrator page.				The administrators will be removed from database and no longer show in the administrator page.	

Test Case ID	ITC013	Test Name	Test security guard page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to security guard page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the administrator's button in the side navigation bar.				-	
Expected Result				Actual Result	
User is navigated to security guard page after the user is authenticated.				User is navigated to the security guard page after the user is authenticated.	

Test Case ID	ITC014	Test Name	Test add security guard page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to add security guard page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the security guard's button in the side navigation bar. 2. User is redirected to security guard's page. 3. User clicks on add security guard button				-	
Expected Result				Actual Result	
User is navigated to the add security guard page after the user is authenticated.				User is navigated to the add security guard page after the user is authenticated.	

Test Case ID	ITC015	Test Name	Test add security guard	Pass/Fail	Pass
Test Case Description		Examine whether the security guard's information added to database after validated			
Test Case Scenario				Test Data	
1. User enters the security guard's page. 2. User clicks on the add security guard button and redirected to add security guard page. 3. User enters the valid security guard 's information. 4. User clicks on the add button.				-	
Expected Result				Actual Result	
New security guard information added to the database and shown in the security guard's page.				New security guard information added to the database and shown in the security guard's page..	

Test Case ID	ITC016	Test Name	Test remove security guards	Pass/Fail	Pass
Test Case Description		Examine whether the security guard's information is removed from database			
Test Case Scenario				Test Data	
1. User enters the security guard's page. 2. User clicks on the remove button on the selected security guards.				-	
Expected Result				Actual Result	
The security guard will be removed from database and no longer show in the security guard page.				The security guard will be removed from database and no longer show in the security guard page.	

Test Case ID	ITC017	Test Name	Test user profile page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to user profile page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the avatar in the side navigation bar.				-	
Expected Result				Actual Result	
User is navigated to user profile page after the user is authenticated.				User is navigated to the user profile page after the user is authenticated.	

Test Case ID	ITC018	Test Name	Test update password	Pass/Fail	Pass
Test Case Description		Examine whether the new password is updated to the backend			
Test Case Scenario				Test Data	
1. User clicks on the avatar in the side navigation bar and redirected to the user profile page. 2. User enters new valid password. 3. User clicks on update profile button.				-	
Expected Result				Actual Result	
The password is updated to the backend of the application. User needs to use the new password to login their account on next login.				The password is updated to the backend of the application. User needs to use the new password to login their account on next login.	

Test Case ID	ITC019	Test Name	Test visitation page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to visitation page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the visitation page in the side navigation bar.				-	
Expected Result				Actual Result	
User is navigated to visitation page after the user is authenticated.				User is navigated to visitation page after the user is authenticated.	

Test Case ID	ITC020	Test Name	Test Announcement page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to announcement page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the announcement page in the side navigation bar.				-	
Expected Result				Actual Result	
User is navigated to announcement page after the user is authenticated.				User is navigated to announcement page after the user is authenticated.	

Test Case ID	ITC021	Test Name	Test add announcement page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to add announcement page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the announcement page in the side navigation bar and redirected to the announcement page. 2. User clicks on the add announcement button.				-	
Expected Result				Actual Result	
User is navigated to add announcement page after the user is authenticated.				User is navigated to add announcement page after the use is authenticated.	

Test Case ID	ITC022	Test Name	Test add announcement	Pass/Fail	Pass
Test Case Description		Examine whether the announcement's information added to database after validated			
Test Case Scenario				Test Data	
1. User enters the add announcement page. 2. User enters the announcement information in valid format. 3. User clicks on the save and publish button				-	
Expected Result				Actual Result	
The newly published announcement is added to database and will be shown in the announcement page.				The newly published announcement is added to database and will be shown in the announcement page.	

Test Case ID	ITC023	Test Name	Test update announcement	Pass/Fail	Pass
Test Case Description		Examine whether the announcement's information updated to database after validated			
Test Case Scenario				Test Data	
1. User enters the announcement page. 2. User clicks on update button on the specific announcement. 3. User enters the updated announcement information in valid format. 4. User clicks on the save and publish button.				-	
Expected Result				Actual Result	
The draft announcement is updated to database and will be shown in the announcement page.				The draft announcement is updated to database and will be shown in the announcement page.	

Test Case ID	ITC024	Test Name	Test remove announcement	Pass/Fail	Pass
Test Case Description		Examine whether the announcement's information is removed from database			
Test Case Scenario				Test Data	
1. User enters the announcement page. 2. User clicks on remove button on the specific announcement.				-	
Expected Result				Actual Result	
The announcement is removed to database and will no longer be shown in the announcement page.				The announcement is removed to database and will no longer be shown in the announcement page.	

Test Case ID	ITC025	Test Name	Test feedback page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to feedback page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the feedback button in the side navigation bar.				-	
Expected Result				Actual Result	
User is navigated to feedback page after the user is authenticated.				User is navigated to feedback page after the user is authenticated.	

Test Case ID	ITC026	Test Name	Test update feedback reply	Pass/Fail	Pass
Test Case Description		Examine whether the feedback's reply is updated to database after validated			
Test Case Scenario				Test Data	
1. User enters the feedback page and clicks on the new feedback tab. 2. User clicks on the reply button on the selected feedback. 3. User enters the reply message in valid format. 4. User clicks on send reply button.				-	
Expected Result				Actual Result	
The reply message will be updated to database and the selected feedback will no longer show in the new feedback tab but in the replied feedback tab.				The reply message will be updated to database and the selected feedback will no longer show in the new feedback tab but in the replied feedback tab.	

Integration Test (mobile application)

Test Case ID	ITC027	Test Name	Test registration page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to registration page			
Test Case Scenario				Test Data	
1. User clicks on register button in login page.				-	
Expected Result				Actual Result	
User is navigated to register page.				User is navigated to register page.	

Test Case ID	ITC028	Test Name	Test submit registration	Pass/Fail	Pass
Test Case Description		Examine whether the submission of registration request is added to database			
Test Case Scenario				Test Data	
1. User enters the registration page. 2. User enters the registration information in valid format. 3. User clicks on submit button				-	
Expected Result				Actual Result	
The registration information is added to database and the user will be redirect to welcome page. (The registration information will be shown in resident page in the web application.)				The registration information is added to database and the user will be redirect to welcome page.	

Test Case ID	ITC029	Test Name	Test login resident page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to login resident page			
Test Case Scenario				Test Data	
1. User clicks on login as resident button in welcome page.				-	
Expected Result				Actual Result	
User is navigated to log in as resident page.				User is navigated to log in as resident page.	

Test Case ID	ITC030	Test Name	Test login security guard page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to security guard's login page			
Test Case Scenario				Test Data	
1. User clicks on login as security guard button in welcome page.				-	
Expected Result				Actual Result	
User is navigated to log in as security guard page.				User is navigated to log in as security guard page.	

Test Case ID	ITC031	Test Name	Test home page if user is authenticated	Pass/Fail	Pass
Test Case Description	Examine whether the user can navigate to home page if user is authenticated				
Test Case Scenario				Test Data	
1. User enters the valid login credentials. 2. User clicks on the login button.				-	
Expected Result				Actual Result	
User is navigated to home page after successfully login.				User is navigated to home page after successfully login.	

Test Case ID	ITC032	Test Name	Test log out	Pass/Fail	Pass
Test Case Description	Examine whether the user is removed from session and redirect to login page				
Test Case Scenario				Test Data	
1. User enters the profile page. 2. User clicks on the log out button.				-	
Expected Result				Actual Result	
User is removed from session and redirect to login page.				User is removed from session and redirect to login page.	

Test Case ID	ITC033	Test Name	Test user profile page	Pass/Fail	Pass
Test Case Description	Examine whether the user can navigate to user profile page after the user is authenticated				
Test Case Scenario				Test Data	
1. User clicks on the profile photo in the home page.				-	
Expected Result				Actual Result	
User is navigated to the profile page.				User is navigated to the profile page.	

Test Case ID	ITC034	Test Name	Test update password	Pass/Fail	Pass
Test Case Description		Examine whether the new password is updated to the backend			
Test Case Scenario				Test Data	
1. User enters profile page. 2. User enters new valid password. 3. User clicks on update password button.				-	
Expected Result				Actual Result	
The new password is updated in the backend. The application will prompt the user to re-login the application with new password.				The new password is updated in the backend. The application will prompt the user to re-login the application with new password.	

Test Case ID	ITC035	Test Name	Test add visitors page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to add visitors page after the user is authenticated			
Test Case Scenario				Test Data	
1. User enters the visitation page. 2. User clicks on the add button.				-	
Expected Result				Actual Result	
User is navigated to add visitor page.				User is navigated to add visitor page.	

Test Case ID	ITC036	Test Name	Test add visitors	Pass/Fail	Pass
Test Case Description		Examine whether the visitation's information is added to database after validated			
Test Case Scenario				Test Data	
1. User enters the add visitor's page. 2. User enters the visitation information in valid format. 3. User clicks on the add button.				-	
Expected Result				Actual Result	
The visitation information is added to database and will be shown in the visitation page.				The visitation information is added to database and will be shown in the visitation page.	

Test Case ID	ITC037	Test Name	Test visitation page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to visitation page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the visitation button on the bottom tab navigation bar. OR 2. User clicks on the visitation button on the home screen.				-	
Expected Result				Actual Result	
User is navigated to the visitation page.				User is navigated to the visitation page.	

Test Case ID	ITC038	Test Name	Test remove visitation	Pass/Fail	Pass
Test Case Description		Examine whether the visitation's information is removed from database			
Test Case Scenario				Test Data	
1. User enters visitation page. 2. User clicks on the remove button on the selected visitation.				-	
Expected Result				Actual Result	
The selected visitation is removed from database and will no longer show in visitation page.				The selected visitation is removed from database and will no longer show in visitation page.	

Test Case ID	ITC039	Test Name	Test check in visitation page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to check in visitation page			
Test Case Scenario				Test Data	
1. User clicks on use as visitors in the welcome page.				-	
Expected Result				Actual Result	
User is navigated to the check in visitation page.				User is navigated to the check in visitation page.	

Test Case ID	ITC040	Test Name	Test verify check in visitation page	Pass/Fail	Pass
Test Case Description	Examine whether the user can navigate to verify check in visitation page				
Test Case Scenario				Test Data	
1. User login as security guards with valid login credentials and is redirected to home page. 2. User clicks on verify registered visitors in home page.				-	
Expected Result				Actual Result	
User is navigated to the verify check in visitation page.				User is navigated to the verify check in visitation page.	

Test Case ID	ITC041	Test Name	Test announcement page	Pass/Fail	Pass
Test Case Description	Examine whether the user can navigate to announcement page after the user is authenticated				
Test Case Scenario				Test Data	
1. User clicks on the announcement button on the bottom tab navigation bar. OR 2. User clicks on the announcement button on the home screen.				-	
Expected Result				Actual Result	
User is navigated to the announcement page.				User is navigated to the announcement page.	

Test Case ID	ITC042	Test Name	Test feedback page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to feedback page after the user is authenticated			
Test Case Scenario				Test Data	
1. User clicks on the feedback button on the bottom tab navigation bar. OR 2. User clicks on the feedback button on the home screen.				-	
Expected Result				Actual Result	
User is navigated to the feedback page.				User is navigated to the feedback page.	

Test Case ID	ITC043	Test Name	Test add feedback page	Pass/Fail	Pass
Test Case Description		Examine whether the user can navigate to add feedback page after the user is authenticated			
Test Case Scenario				Test Data	
1. User enters feedback page. 2. Users clicks on add feedback button.				-	
Expected Result				Actual Result	
User is navigated to the add feedback page.				User is navigated to the add feedback page.	

Test Case ID	ITC044	Test Name	Test add feedback	Pass/Fail	Pass
Test Case Description		Examine whether the feedback is added to database after validated			
Test Case Scenario				Test Data	
1. User enters add feedback page. 2. User enters feedback information in valid format. 3. User clicks on add feedback button.				-	
Expected Result				Actual Result	
The feedback information is added into database and will be shown in feedback page.				The feedback information is added into database and will be shown in feedback page.	

User Acceptance Test (web application)

Test Case ID	UATC001	Test Name	Login account	Tested by	Management Team
Test Case Description		To verify the account is logged in			
Test Case Instruction				Test Data	
1. Enter the application in web browser with the url (). 2. Enter the valid email and password. 3. Clicks on login button.				Email: admin@gmail.com Password: 123456789	
Expected Result				Actual Result	
The user is authenticated and redirected to home page.				The user is authenticated and redirected to home page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass				1.	

Test Case ID	UATC002	Test Name	Approve resident's registration	Tested by	Management Team
Test Case Description		To verify the resident's registration is approved			
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on residents' button on the side navigation bar. 3. The application will display all the resident's registration. 4. Looks for the resident named "Ken Tan" 5. Clicks on the approve button on the resident row.				-	
Expected Result				Actual Result	
A successful update status message is showed and added to existing residents tab.				A successful update status message is showed and added to existing residents tab.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC003	Test Name	Reject resident's registration	Tested by	Management Team
Test Case Description	To verify the resident's registration is rejected				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on residents' button on the side navigation bar. 3. Clicks on the new application tab. 4. The application displays all the resident's registration. 5. Looks for the resident named "Julian Lim" 6. Clicks on the reject button on the resident row. 7. The application prompt the user to confirm reject the resident's registration. 8. Clicks on confirm reject button.				-	
Expected Result				Actual Result	
A successful reject message is showed and removed from the resident's registration list.				A successful update status message is showed and added to existing residents tab.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC004	Test Name	View all resident's information	Tested by	Management Team
Test Case Description	To verify all the resident information is showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on residents' button on the side navigation bar. 3. Clicks on the existing residents tab.				-	
Expected Result				Actual Result	
The application shows all the existing resident information.				The application shows all the existing resident information.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC005	Test Name	View All Administrators	Tested by	Management Team
Test Case Description	To verify all the adminstrators are showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on administrator's button on the side navigation bar.				-	
Expected Result				Actual Result	
The application shows all the administrator information.				The application shows all the administrator information.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC008	Test Name	View All Security guards	Tested by	Management Team
Test Case Description	To verify all the security guards are showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on security guard's button on the side navigation bar.				-	
Expected Result				Actual Result	
The application shows all the administrator information.				The application shows all the administrator information.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC006	Test Name	Add new administrator	Tested by	Management Team
Test Case Description	To verify new administrator is added				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on administrator's button on the side navigation bar. 3. The application displays all administrator's information. 4. Clicks on the add administrator button. 5. The application redirect to add administrator page and displays a form. 6. Enters name, identification no, email address, phone number and role. 7. Clicks on add button.				Name: Jason Liew Identification card no: 880725042154 Email address: jasonliew@gmail.com Phone number: 60152459854 Role: staff	
Expected Result				Actual Result	
The application prompt the successfully added message.				The application prompt the successfully added message.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC007	Test Name	Remove administrators	Tested by	Management Team
Test Case Description		To verify administrator is removed			
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on administrator's button on the side navigation bar. 3. The application displays all administrator's information. 4. Looks for the administrator named "Phil lam". 5. Clicks on the remove button on the administrator row. 6. The application prompt the user to confirm delete administrators. 7. Clicks on confirm delete button.				-	
Expected Result				Actual Result	
The application prompt the successfully deleted message.				The application prompt the successfully deleted message.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC009	Test Name	Add new security guard	Tested by	Management Team
Test Case Description	To verify new security guard is added				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on security guard's button on the side navigation bar. 3. The application displays all security guard's information. 4. Clicks on the add security guard button. 5. The application redirect to add security guard page and displays a form. 6. Enters name, identification no, email address, and phone number. 7. Clicks on add button.				Name: Keith Sia Yuan Identification card no: 880423054524 Email address: keithyuan@gmail.com Phone number: 60182548658	
Expected Result				Actual Result	
The application prompt the successfully added message.				The application prompt the successfully added message.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC010	Test Name	Remove security guard	Tested by	Management Team
Test Case Description		To verify security guard is removed			
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on security guard's button on the side navigation bar. 3. The application displays all security guard's information. 4. Looks for the security guard named "Kok Kim Jun". 5. Clicks on the remove button on the security guard row. 6. The application prompt the user to confirm delete security guard. 7. Clicks on the confirm delete button.				-	
Expected Result				Actual Result	
The application prompt the successfully deleted message.				The application prompt the successfully deleted message.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC011	Test Name	Modify new password	Tested by	Management Team
Test Case Description	To verify new password can successfully login				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the profile photo on the side navigation bar. 3. The application will redirect to the profile page. 4. Enters new valid password. 5. Clicks on update button.				New Password: 987654321	
Expected Result				Actual Result	
The application prompt the successfully update new password message.				The application prompt the successfully update new password message.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC012	Test Name	View all visitation information	Tested by	Management Team
Test Case Description	To verify all the visitation information is showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the visitors button on the side navigation bar. 3. The application will redirect to the visitation page. 4. Clicks on the total visitors' tabs.				-	
Expected Result				Actual Result	
All the checked in visitation will be displayed.				All the checked in visitation will be displayed.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC013	Test Name	Search visitation using unit id	Tested by	Management Team
Test Case Description	To verify the related visitation is showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the visitors button on the side navigation bar. 3. The application will redirect to the visitation page. 4. Clicks on the total visitors' tabs. 5. Enters unit id in the search bar.				Unit id: 87	
Expected Result				Actual Result	

All the searched visitation with related unit id are showed.	All the searched visitation with related unit id are showed.
Pass / Fail	Comments
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass	

Test Case ID	UATC014	Test Name	View all announcement	Tested by	Management Team
Test Case Description	To verify all announcement is showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the announcement button on the side navigation bar. 3. The application displays all the announcement.				-	
Expected Result				Actual Result	
All announcements are displayed on the announcement.				All announcements are displayed on the announcement.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC015	Test Name	Add and publish new announcement	Tested by	Management Team
Test Case Description	To verify announcement is added and published				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the announcement button on the side navigation bar. 3. The application displays all the announcement. 4. Clicks on add new announcement button. 5. Enters title, description, and upload image. 6. Clicks on the save and publish button.				Title: Update Swimming Pool Opening Time Description: The operation hours is updated to 8:00pm. Image: pool.png	
Expected Result				Actual Result	
The successfully added and published message is prompted and redirect to the announcement page.				The successfully added and published message is prompted and redirect to the announcement page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC016	Test Name	Add draft announcement	Tested by	Management Team
Test Case Description		To verify announcement is added and save as drafted			
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the announcement button on the side navigation bar. 3. The application displays all the announcement. 4. Clicks on add new announcement button. 5. Enters title, description, and upload image. 6. Clicks on the save and publish button.				Title: Update Gym Room Opening Time Description: The operation hours is updated to 6:00pm. Image: gym.png	
Expected Result				Actual Result	
The successfully added and save as draft message is prompted and redirect to the announcement page.				The successfully added and save as draft message is prompted and redirect to the announcement page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC017	Test Name	Update and publish draft announcement	Tested by	Management Team
Test Case Description	To verify announcement is added and save as drafted				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the announcement button on the side navigation bar. 3. The application displays all the announcement. 4. Clicks on edit button on the announcement with title “Management Fee Adjustment” 5. The application redirects to the edit announcement page. 6. Edit the description with the test data. 7. Clicks on save and publish button.				Description: The management fee will be adjusted to RM800 from 1 Jan 2023 onwards.	
Expected Result				Actual Result	
The successfully added and publish message is prompted and redirect to the announcement page.				The successfully added and publish message is prompted and redirect to the announcement page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC018	Test Name	View announcement	Tested by	Management Team
Test Case Description	To verify announcement detail is showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the announcement button on the side navigation bar. 3. The application displays all the announcement. 4. Clicks on view button on the announcement with title “Management Fee Adjustment”				-	
Expected Result				Actual Result	
The application redirects to the view announcement page and the title, updated description in test case “update and publish draft announcement” (UATC015), image are showed in the announcement details page.				The application redirects to the view announcement page and the title, updated description in test case “update and publish draft announcement” (UATC015), image are showed in the announcement details page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC019	Test Name	Remove announcement	Tested by	Management Team
Test Case Description		To verify announcement is removed			
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the announcement button on the side navigation bar. 3. The application displays all the announcement. 4. Clicks on remove button the announcement with the title "Management Fee Adjustment 2022" 5. The application prompt the user to confirm remove announcement. 6. Clicks on confirm delete button				-	
Expected Result				Actual Result	
The application prompt the successfully deleted message.				The application prompt the successfully deleted message.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC020	Test Name	View all feedback	Tested by	Management Team
Test Case Description		To verify all feedback is showed			
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the feedback button on the side navigation bar. 3. The application displays all the feedback in 2 different tab. 4. Clicks on replied feedback tab.				-	
Expected Result				Actual Result	
All the replied feedback is showed in a list.				All the replied feedback is showed in a list.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC021	Test Name	Reply new feedback	Tested by	Management Team
Test Case Description		To verify new feedback is replied			
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the feedback button on the side navigation bar. 3. Clicks on the new feedback tab. 4. Clicks on the reply button on the new feedback named “Roof Leaking” 5. The application redirects to the reply feedback page. 6. Clicks on reply button. 7. Enters the reply message. 8. Clicks on send reply button.				-	
Expected Result				Actual Result	
The successful updated message is prompted and the application redirects to feedback page.				The successful updated message is prompted and the application redirects to feedback page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass				Tester 2 responded should allows more reply to 1 feedback.	

Test Case ID	UATC022	Test Name	View replied feedback	Tested by	Management Team
Test Case Description	To verify all feedback is showed				
Test Case Instruction				Test Data	
1. Log in the account. 2. Clicks on the feedback button on the side navigation bar. 3. The application displays all the feedback in 2 different tab. 4. Clicks on replied feedback tab. 5. Clicks on view button with the feedback name “Complaint on swimming pool operating hours” 6. The application redirects to replied feedback page.				-	
Expected Result				Actual Result	
The feedback details and replied message by administrator is showed.				The feedback details and replied message by administrator is showed. .	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

User Acceptance Test (Mobile application)

Test Case ID	UATC023	Test Name	Submit registration form	Tested by	Resident
Test Case Description		To verify the registration form is added			
Test Case Instruction				Test Data	
1. Enter the application on a mobile device. 2. Click on log in as resident button. 3. The mobile application redirects to the login as resident page. 4. Clicks on sign up button. 5. The application redirects to the register as residents page. 6. Enters name, identification card number, email address, password, car plate, unit id, address, and uploads supporting documents. 7. Clicks on signup button.				Name: Lily Tan IC no: 980401080456 Email: lilytan@gmail.com Password: 123456789 Carplate: KDD8752 Unit id: 812 Address: 8-1-2, Jalan Sungai Long, 43000, Kajang. Supporting document: water_bill.png	
Expected Result				Actual Result	
A successfully registered message is prompted, and the application will redirect the user to welcome page.				A successfully registered message is prompted, and the application will redirect the user to welcome page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC022	Test Name	Login account	Tested by	Resident
Test Case Description		To verify the account is logged in			
Test Case Instruction				Test Data	
1. Enter the application in a mobile device. 2. Click on log in as resident button. 3. Enter valid email and password. 4. Click on log in button.				Email: resident@gmail.com Password: 123456789	
Expected Result				Actual Result	
The user is authenticated and redirected to home page.				The user is authenticated and redirected to home page.	
Entry Criteria				Exit Criteria	
Pass / Fail				Comments	
7. Pass 8. Pass 9. Pass 10. Pass 11. Pass 12. Pass					

Test Case ID	UATC025	Test Name	Login account	Tested by	Security guards
Test Case Description	To verify the account is logged in				
Test Case Instruction				Test Data	
1. Enter the application in a mobile device. 2. Click on log in as resident button. 3. Enter valid email and password. 4. Click on log in button.				Email: s@gmail.com Password: 123456789	
Expected Result				Actual Result	
The user is authenticated and redirected to home page.				The user is authenticated and redirected to home page.	
Pass / Fail				Comments	
13. Pass 14. Pass 15. Pass 16. Pass 17. Pass 18. Pass					

Test Case ID	UATC026	Test Name	Logout	Tested by	Security guards
Test Case Description	To verify the user is logged out				
Test Case Instruction				Test Data	
1. Clicks on the home page button. 2. Clicks on the button of profile picture. 3. The application redirects to profile page. 4. Clicks on log out button.				-	
Expected Result				Actual Result	
The user is redirected to login page.				The user is redirected to login page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC027	Test Name	Logout	Tested by	Residents
Test Case Description	To verify the user is logged out				
Test Case Instruction				Test Data	
1. Clicks on log out button on home page.				-	
Expected Result				Actual Result	
The user is redirected to login page.				The user is redirected to login page.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC028	Test Name	Modify user profile	Tested by	Resident
Test Case Description	To verify phone number, car plate, and password is updated and can successfully login after change the password				
Test Case Instruction				Test Data	
1. Login the account 2. Clicks on the home page button. 3. Clicks on the button of profile picture. 4. The application redirects to profile page. 5. Enter new phone number, carplate number, and password. 6. Clicks on edit profile button.				Phone number: 60154235789 Carplate number: FS242 Password: 987654321	
Expected Result				Actual Result	
The new phone number, carplate number are updated and added message is prompt. The user is				The new phone number, carplate number are updated. The user is	

automatically logged out and able to log in with new password.	automatically logged out and able to log in with new password.
Pass / Fail	Comments
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass	

Test Case ID	UATC029	Test Name	Add visitor	Tested by	Resident
Test Case Description		To verify new visitors is added			
Test Case Instruction				Test Data	
1. Login the account 2. Clicks on the visitors button on the bottom tab navigation bar. 3. Clicks on add visitors button. 4. Enter visitor's name, identification card no, phone number, date visited, time visited, car plate. 5. Clicks on submit button.				Name: Joshua Identification card no: 980201054654 Phone number: 60127518975 Date visited: 13 September 2022 Time visited: 12:00pm	
Expected Result				Actual Result	
The successfully added message will be prompt.				The successfully added message are prompt.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC030	Test Name	Add visitor	Tested by	Security guard
Test Case Description		To verify new visitors is added			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on register ad-hoc visitors button on the home page. 3. The application redirects to add visitor page. 4. Enters visitor's name, visitor's IC no, phone number, car plate, unit id. 5. Clicks on submit button.				Name: Onn Sha Sha IC no: 780415048756 Phone number: 0168746545 Car plate: JSD5456 Unit id: 846	
Expected Result				Actual Result	
The successfully added message will be prompt.				The successfully added message are prompt.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC031	Test Name	View all visitation	Tested by	Resident
Test Case Description		To verify all visitation is showed			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the visitors button on the bottom tab navigation bar.				-	
Expected Result				Actual Result	
All the visitation information are showed.				All the visitation information are showed.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC032	Test Name	View visitation	Tested by	Resident
Test Case Description		To verify the visitation detail is showed			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the visitors button on the bottom tab navigation bar. 3. Clicks on the view button on visitation card with the visitor's named "Oaklynn". 4. The application redirects to view visitors page.				-	
Expected Result				Actual Result	
The application will be navigated to visitation detail page and the visitor’s details are showed.				The application will be navigated to visitation detail page and the visitor’s details are showed.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC033	Test Name	Remove upcoming visitation	Tested by	Resident
Test Case Description		To verify the visitation is removed			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the visitors button on the bottom tab navigation bar. 3. Clicks on the remove button on visitation card with the visitor's named "Apollo". 4. The application prompt a alert message to confirm user remove action. 5. Clicks on ok button.				-	
Expected Result				Actual Result	
The successfully deleted message will be prompt and the visitation will be deleted from database.				The successfully deleted message are prompt and the visitation are deleted from database.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC034	Test Name	Check in visitation	Tested by	Visitor
Test Case Description		To verify the visitation is successfully checked in			
Test Case Instruction				Test Data	
1. Enter the application on a mobile device. 2. Click on log in as visitors. 3. Enters visitation id and unit id. 4. The application show the generated qr code for this visitation. 5. Show the qr code to security guard.				Visitation id: CY875645 Unit ID:87	
Expected Result				Actual Result	
The application will show the qr code with the visitation detail and able to scan by the security guard.				The application showed the qr code with the visitation detail and able to be scanned by the security guard.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC035	Test Name	Verify check in visitation	Tested by	Security guard
Test Case Description		To verify the check in visitation is verified			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the verify registered visitors button on the home page. 3. Scan the qr code shown by visitors. 4. The application will show the details of the visitation. 5. Clicks on the verify button.				QR shown by visitors	
Expected Result				Actual Result	
The application will be navigated to the verified visitors page. The visitation updated with the checkin date and time and the security guard id.				The application will be navigated to the verified visitors page. The visitation updated with the checkin date and time and the security guard id.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC036	Test Name	View all announcement	Tested by	Resident
Test Case Description		To verify all announcement is showed			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the announcement button on the bottom tab navigation bar.				-	
Expected Result				Actual Result	
All announcement is showed.				All announcement is showed.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC037	Test Name	View announcement	Tested by	Resident
Test Case Description		To verify announcement detail is showed			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the announcement button on the bottom tab navigation bar. 3. The application redirects to announcement page. 4. Clicks on the view more button on announcement with the title "Updated Management Fee Notice."				-	
Expected Result				Actual Result	
The application will navigate to view announcement page and the details of the announcement are showed.				The application navigates to view announcement page and the details of the announcement are showed.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC038	Test Name	View all feedback	Tested by	Resident
Test Case Description		To verify all feedback is showed			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the feedback button on the bottom tab navigation bar.				-	
Expected Result				Actual Result	
All the feedbacks are showed.				All the feedbacks are showed.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC039	Test Name	View feedback	Tested by	Resident
Test Case Description		To verify feedback detail is showed			
Test Case Instruction				Test Data	
1. Login the account. 2. Clicks on the feedback button on the bottom tab navigation bar. 3. The application redirects to feedback page. 4. Clicks on the view more button on the feedback with the title "Rooftop leaking issue".				-	
Expected Result				Actual Result	
The application will navigate to view feedback page and all the feedback details and replies will show.				The application navigates to view feedback page and all the feedback details and replies are showed.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

Test Case ID	UATC040	Test Name	Add new feedback	Tested by	Resident
Test Case Description		To verify feedback is added			
Test Case Instruction				Test Data	
1. Login the account. 2. Click on the feedback button on the bottom tab navigation bar. 3. The application redirects to feedback page. 4. Click on the add button on the feedback page. 5. The application redirects to add feedback page. 6. Enter title, description, categories of feedback. 7. Clicks on submit button.				Title: Complaint on Level 7 Neighbour Description: Always hear the noise coming from my upstairs after 12am. Categories: others	
Expected Result				Actual Result	
The successfully added message will be prompt.				The successfully added message is prompt.	
Pass / Fail				Comments	
1. Pass 2. Pass 3. Pass 4. Pass 5. Pass 6. Pass					

APPENDIX C: User Acceptance Test Questionnaire

User Acceptance and Satisfactory Form (Web application)

User Acceptance and Satisfactory Form

Hi, I am Low Ee Lyne, an undergraduate student of Universiti Tunku Abdul Rahman (UTAR) pursuing on Bachelor of Science (Honours) Software Engineering. This survey is conducted as part of the data collection of the course of Final Year Project. The main objective of this survey is to collect user acceptance and feedback towards the Resident and Visitor Management System.

Resident and Visitor Management System consists of web application and mobile application. The web application is to help the management team simplify their administrative work while the mobile application is to facilitate the communication between the management teams and residents.

This survey requires you as a management team members to explore on the Residents and Visitor Management System through this link (https://rvsm-rho.vercel.app) before answering it. This web application is responsive, you are able to use it on mobile, tablet or computer device. This survey contains 3 sections.

Your details and responses will be kept entirely confidential, the questionnaire and the information collected will be kept anonymously and the feedback received are only used for academic practice and research purposes.

If you have any questions about this survey questionnaire, kindly contact me through the email address (eelyne1099@1utar.my). I am willing to answer or clarify any questions that require further explanation. I appreciate your cooperation in completing this survey and providing precious time, accurate information and thoughtful suggestions to complete this questionnaire.

Yours faithfully,
Low Ee Lyne
Lee Kong Chian Faculty of Engineering and Science
Bachelor of Science (Honours) Software Engineering
Universiti Tunku Abdul Rahman (UTAR)

Sign in to Google to save your progress. Learn more

*Required

Which age group are you in? *

- ◯ 20-30
- ◯ 31-40
- ◯ 41-50
- ◯ >50

Next Clear form

User Acceptance and Satisfactory Form

Sign in to Google to save your progress. Learn more

*Required

Kindly open the application by through the link (https://rvsm-rho.vercel.app) and perform each test accordingly.

Able to login account? *

1. Enter the application in web browser.
2. Enter the valid email and password. (Email: admin@gmail.com, Password: 123456789)
3. Clicks on login button.

- ◯ Yes
- ◯ No

Able to approve resident's registration *

1. Log in the account.
2. Clicks on residents' button on the side navigation bar.
3. The application will display all the resident's registration.
4. Looks for the resident named "Ken Tan"
5. Clicks on the approve button on the resident row.

- ◯ Yes
- ◯ No

Able to reject resident's registration *

1. Log in the account.
2. Clicks on residents' button on the side navigation bar.
3. Clicks on the new application tab.
4. The application displays all the resident's registration.
5. Looks for the resident named "Julian Lim"
6. Clicks on the reject button on the resident row.
7. The application prompt the user to confirm reject the resident's registration.
8. Clicks on confirm reject button.

- ◯ Yes
- ◯ No

Able to view all resident's information *

1. Log in the account.
2. Clicks on residents' button on the side navigation bar.
3. Clicks on the existing residents tab.

- ◯ Yes
- ◯ No

Able to view all administrators *

1. Log in the account.
2. Clicks on administrator's button on the side navigation bar.

- ◯ Yes
- ◯ No

Able to add new administrator *

1. Log in the account.
2. Clicks on administrator's button on the side navigation bar.
3. The application displays all administrator's information.
4. Clicks on the add administrator button.
5. The application redirect to add administrator page and displays a form.
6. Enters name, identification no, email address, phone number and role.
(Name: Jason Liew, Identification card no: 880725042154, Email address: jasonliew@gmail.com, Phone number: 60152459854, Role: staff)
7. Clicks on add button.

- ◯ Yes
- ◯ No

Able to remove administrators *

1. Log in the account.
2. Clicks on administrator's button on the side navigation bar.
3. The application displays all administrator's information.
4. Looks for the administrator named "Phil lam".
5. Clicks on the remove button on the administrator row.
6. The application prompt the user to confirm delete administrators.
7. Clicks on confirm delete button.

- ◯ Yes
- ◯ No

Able to view all security guard *

1. Log in the account.
2. Clicks on security guard's button on the side navigation bar.

- ◯ Yes
- ◯ No

Able to add new security guard *

1. Log in the account.
2. Clicks on security guard's button on the side navigation bar.
3. The application displays all security guard's information.
4. Clicks on the add security guard button.
5. The application redirect to add security guard page and displays a form.
6. Enters name, identification no, email address, and phone number.
(Name: Keith Sia Yuan, Identification card no: 880423054524, Email address: keithyuan@gmail.com, Phone number: 60182548658)
7. Clicks on add button.

- ◯ Yes
- ◯ No

Able to remove security guard *

1. Log in the account.
2. Clicks on security guard's button on the side navigation bar.
3. The application displays all security guard's information.
4. Looks for the security guard named "Kok Kim Jun".
5. Clicks on the remove button on the security guard row.
6. The application prompt the user to confirm delete security guard.
7. Clicks on the confirm delete button.

- ◯ Yes
- ◯ No

Able to modify new password *

1. Log in the account.
2. Clicks on the profile photo on the side navigation bar.
3. The application will redirect to the profile page.
4. Enters new valid password. (New Password: 987654321)
5. Clicks on update button.

- ◯ Yes
- ◯ No

Able to view all visitation information *

1. Log in the account.
2. Clicks on the visitors button on the side navigation bar.
3. The application will redirect to the visitation page.
4. Clicks on the total visitors' tabs.

◯ Yes

◯ No

Able to search visitation using unit id *

1. Log in the account.
2. Clicks on the visitors button on the side navigation bar.
3. The application will redirect to the visitation page.
4. Clicks on the total visitors' tabs.
5. Enters unit id in the search bar. (Unit id: 87)

◯ Yes

◯ No

Able to view all announcement *

1. Log in the account.
2. Clicks on the announcement button on the side navigation bar.
3. The application displays all the announcement.

◯ Yes

◯ No

Able to add and publish announcement *

1. Log in the account.
2. Clicks on the announcement button on the side navigation bar.
3. The application displays all the announcement.
4. Clicks on add new announcement button.
5. Enters title, description, and upload image.
*(*Title: Update Swimming Pool Opening Time, Description: The operation hours is updated to 8:00pm. , Image: pool.png*)*
6. Clicks on the save and publish button.

◯ Yes

◯ No

Able to add draft announcement *

1. Log in the account.
2. Clicks on the announcement button on the side navigation bar.
3. The application displays all the announcement.
4. Clicks on add new announcement button.
5. Enters title, description, and upload image.
(Title: Update Gym Room Opening Time, Description: The operation hours is updated to 6:00pm., Image: gym.png)
6. Clicks on the save and publish button.

- ◯ Yes
- ◯ No

Able to update and publish draft announcement *

1. Log in the account.
2. Clicks on the announcement button on the side navigation bar.
3. The application displays all the announcement.
4. Clicks on edit button on the announcement with title "Management Fee Adjustment"
5. The application redirects to the edit announcement page.
6. Edit the description. (Description: The management fee will be adjusted to RM800 from 1 Jan 2023 onwards.)
7. Clicks on save and publish button.

- ◯ Yes
- ◯ No

Able to view announcement *

1. Log in the account.
2. Clicks on the announcement button on the side navigation bar.
3. The application displays all the announcement.
4. Clicks on view button on the announcement with title "Management Fee Adjustment"

- ◯ Yes
- ◯ No

Able to remove announcement *

1. Log in the account.
2. Clicks on the announcement button on the side navigation bar.
3. The application displays all the announcement.
4. Clicks on remove button the announcement with the title "Management Fee Adjustment 2022"
5. The application prompt the user to confirm remove announcement.
6. Clicks on confirm delete button

- ◯ Yes
- ◯ No

Able to view all feedback *

1. Log in the account.
2. Clicks on the feedback button on the side navigation bar.
3. The application displays all the feedback in 2 different tab.
4. Clicks on replied feedback tab.

- ◯ Yes
- ◯ No

Able to reply new feedback *

1. Log in the account.
2. Clicks on the feedback button on the side navigation bar.
3. Clicks on the new feedback tab.
4. Clicks on the reply button on the new feedback named "Roof Leaking"
5. The application redirects to the reply feedback page.
6. Clicks on reply button.
7. Enters the reply message.
8. Clicks on send reply button.

- ◯ Yes
- ◯ No

Able to view replied feedback *

1. Log in the account.
2. Clicks on the feedback button on the side navigation bar.
3. Clicks on the new feedback tab.
4. Clicks on the reply button on the new feedback named "Roof Leaking"
5. The application redirects to the reply feedback page.
6. Clicks on reply button.
7. Enters the reply message.
8. Clicks on send reply button.

- ◯ Yes
- ◯ No

Back Next Clear form

User Acceptance and Satisfactory Form

Sign in to Google to save your progress. Learn more

*Required

Share your feedback after performing the test.

This application will help you to facilitate communication with residents. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you easier to manage resident's information. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you easier to manage visitor's information *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you easier to publish announcement. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you easier to manage resident's feedback. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you easier to manage security teams information. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you to simply administrative work. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Rate the overall user interface design *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Rate the accuracy of data in the application. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Rate the satisfaction level in the application. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

As a user of the web application, would you sign off the user acceptance test? *

(Sign off - you agree that they have thoroughly tested the solution, and agree that it is bug-free and ready for deployment)

- ○ Yes
- ○ No

Submit Clear form

User Acceptance and Satisfactory Form (Mobile application)

User Acceptance and Satisfactory Form

Hi, I am Low Ee Lyne, an undergraduate student of Universiti Tunku Abdul Rahman (UTAR) pursuing on Bachelor of Science (Honours) Software Engineering. This survey is conducted as part of the data collection of the course of Final Year Project. The main objective of this survey is to collect user acceptance and feedback towards the Resident and Visitor Management System. Resident and Visitor Management System consists of web application and mobile application. The web application is to help the management team simplify their administrative work while the mobile application is to facilitate the communication between the management teams and residents.

This survey requires you to explore on the mobile application of Residents and Visitor Management System through this link (For android device -) before answering it. This survey contains 3 sections.

Your details and responses will be kept entirely confidential, the questionnaire and the information collected will be kept anonymously and the feedback received are only used for academic practice and research purposes.

If you have any questions about this survey questionnaire, kindly contact me through the email address (eelyne1099@1utar.my). I am willing to answer or clarify any questions that require further explanation. I appreciate your cooperation in completing this survey and providing precious time, accurate information and thoughtful suggestions to complete this questionnaire.

Yours faithfully,
Low Ee Lyne
Lee Kong Chian Faculty of Engineering and Science
Bachelor of Science (Honours) Software Engineering
Universiti Tunku Abdul Rahman (UTAR)

Sign in to Google to save your progress. Learn more

*Required

Who are you tested as? *

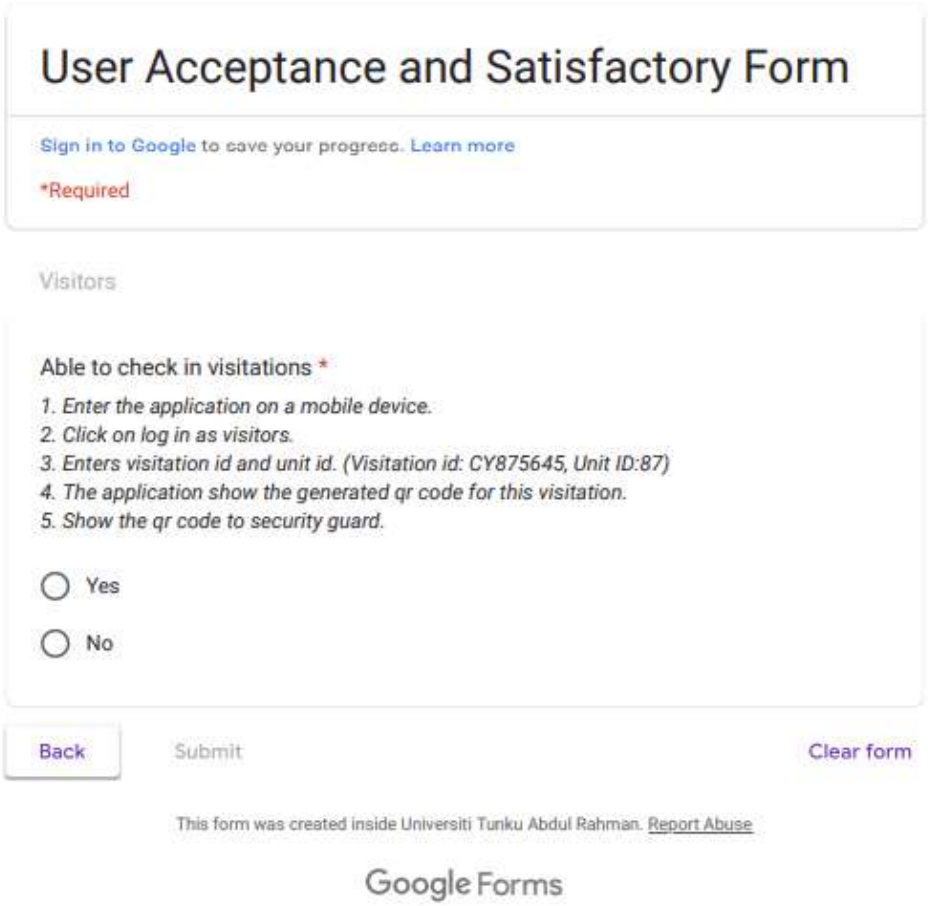
- ◯ Resident
- ◯ Visitor
- ◯ Security guard

Which age group are you in? *

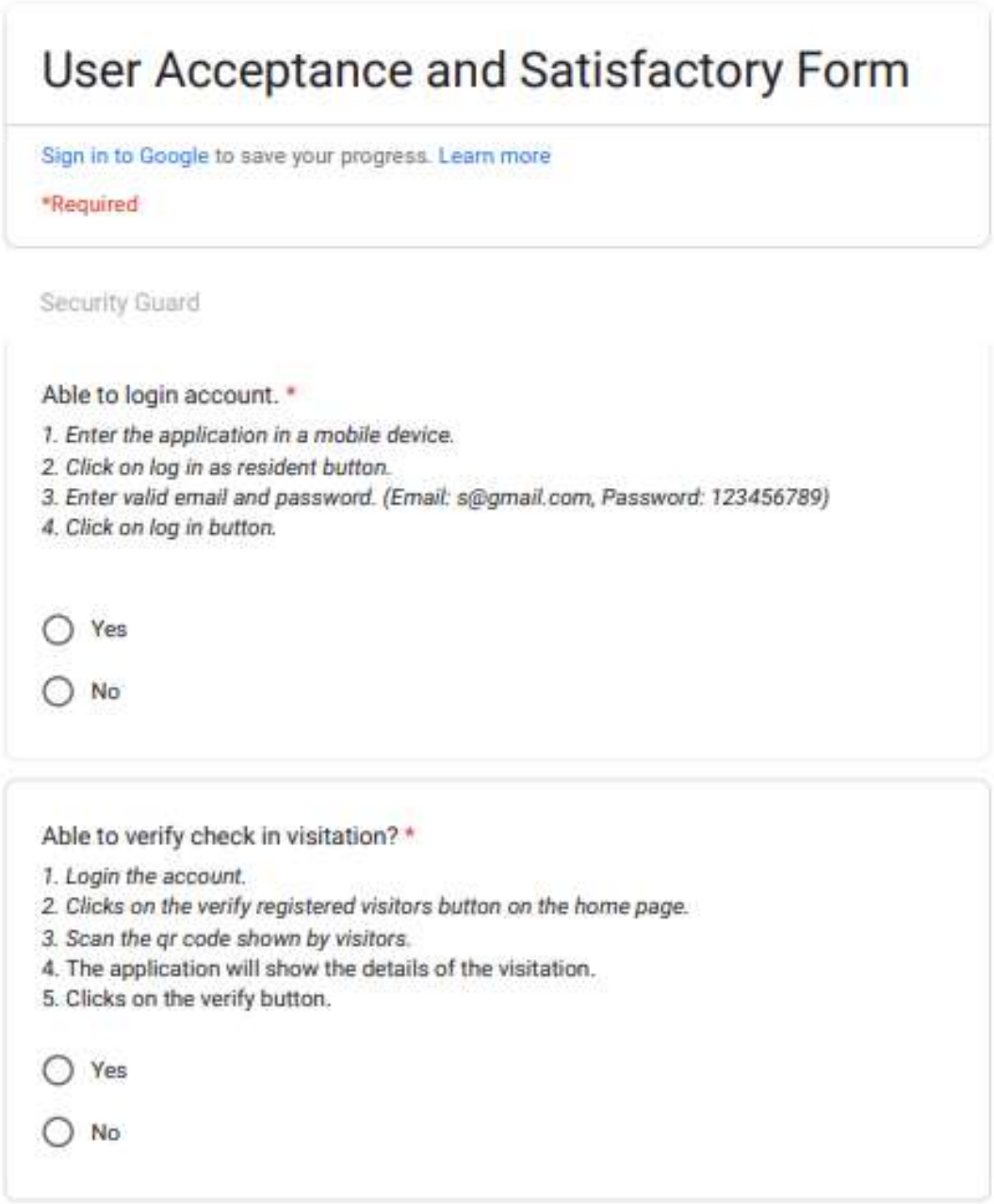
- ◯ 20-30
- ◯ 31-40
- ◯ 41-50
- ◯ >50

Next Clear form

For Visitor


User Acceptance and Satisfactory Form
Sign in to Google to save your progress. Learn more
*Required
Visitors
Able to check in visitations *
1. Enter the application on a mobile device.
2. Click on log in as visitors.
3. Enters visitation id and unit id. (Visitation id: CY875645, Unit ID:87)
4. The application show the generated qr code for this visitation.
5. Show the qr code to security guard.
Yes
No
Back
Submit
Clear form
This form was created inside Universiti Tunku Abdul Rahman. Report Abuse
Google Forms

For security guard



User Acceptance and Satisfactory Form

Sign in to Google to save your progress. Learn more

*Required

Security Guard

Able to login account. *

1. Enter the application in a mobile device.
2. Click on log in as resident button.
3. Enter valid email and password. (Email: s@gmail.com, Password: 123456789)
4. Click on log in button.

- ◯ Yes
- ◯ No

Able to verify check in visitation? *

1. Login the account.
2. Clicks on the verify registered visitors button on the home page.
3. Scan the qr code shown by visitors.
4. The application will show the details of the visitation.
5. Clicks on the verify button.

- ◯ Yes
- ◯ No

Able to add ad-hoc visitors *

1. Login the account.
2. Clicks on register ad-hoc visitors button on the home page.
3. The application redirects to add visitor page.
4. Enters visitor's name, visitor's IC no, phone number, car plate, unit id.
(Name: Onn Sha Sha, IC no: 780415048756, phone number: 0168746545, carplate: JSD5456, unit id: 846)
5. Clicks on submit button.

- ◯ Yes
- ◯ No

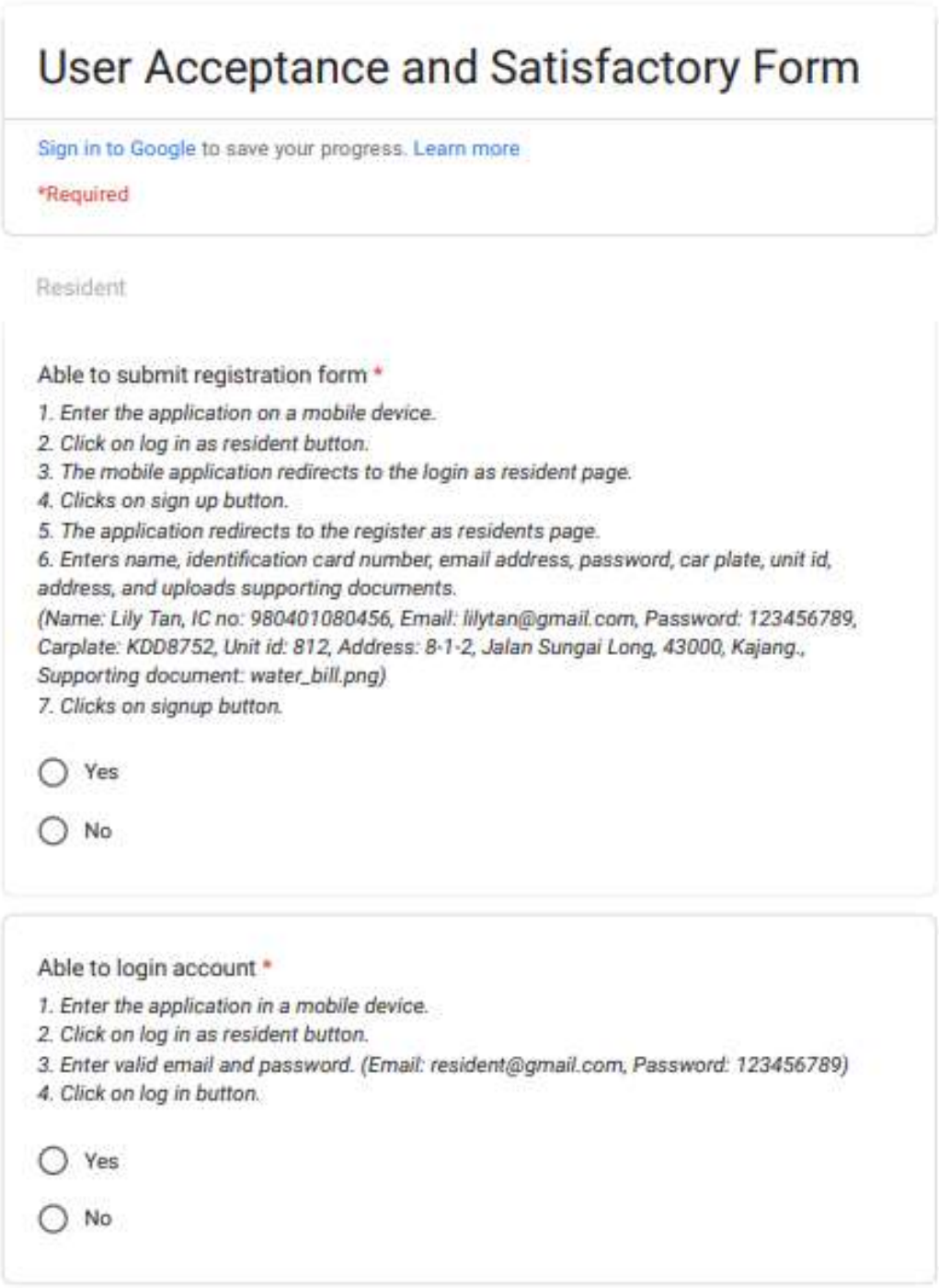
Able to logout account *

1. Clicks on log out button on home page.

- ◯ Yes
- ◯ No

Back Submit Clear form

For residents



User Acceptance and Satisfactory Form

Sign in to Google to save your progress. Learn more

*Required

Resident

Able to submit registration form *

1. Enter the application on a mobile device.
2. Click on log in as resident button.
3. The mobile application redirects to the login as resident page.
4. Clicks on sign up button.
5. The application redirects to the register as residents page.
6. Enters name, identification card number, email address, password, car plate, unit id, address, and uploads supporting documents.
(Name: Lily Tan, IC no: 980401080456, Email: lilytan@gmail.com, Password: 123456789, Carplate: KDD8752, Unit id: 812, Address: 8-1-2, Jalan Sungai Long, 43000, Kajang., Supporting document: water_bill.png)
7. Clicks on signup button.

◯ Yes

◯ No

Able to login account *

1. Enter the application in a mobile device.
2. Click on log in as resident button.
3. Enter valid email and password. (Email: resident@gmail.com, Password: 123456789)
4. Click on log in button.

◯ Yes

◯ No

Able to logout account *

1. Clicks on the home page button.
2. Clicks on the button of profile picture.
3. The application redirects to profile page.
4. Clicks on log out button.

- ◯ Yes
- ◯ No

Able to modify user profile *

1. Login the account
2. Clicks on the home page button.
3. Clicks on the button of profile picture.
4. The application redirects to profile page.
5. Enter new phone number, carplate number, and password. (Phone number: 60154235789, Carplate number: FS242, Password: 987654321)
6. Clicks on edit profile button.

- ◯ Yes
- ◯ No

Able to add visitor *

1. Login the account
2. Clicks on the visitors button on the bottom tab navigation bar.
3. Clicks on add visitors button.
4. Enter visitor's name, identification card no, phone number, date visited, time visited, car plate. (Name: Joshua, Identification card no: 980201054654, Phone number: 60127518975, Date visited: 13 September 2022, Time visited: 12:00pm)
5. Clicks on submit button.

- ◯ Yes
- ◯ No

Able to view all visitation *

1. Login the account.
2. Clicks on the visitors button on the bottom tab navigation bar.

- ◯ Yes
- ◯ No

Able to view visitation *

1. Login the account.
2. Clicks on the visitors button on the bottom tab navigation bar.
3. Clicks on the view button on visitation card with the visitor's named "Oaklynn".
4. The application redirects to view visitors page.

- ◯ Yes
- ◯ No

Able to remove upcoming visitation *

1. Login the account.
2. Clicks on the visitors button on the bottom tab navigation bar.
3. Clicks on the remove button on visitation card with the visitor's named "Apollo".
4. The application prompt a alert message to confirm user remove action.
5. Clicks on ok button.

- ◯ Yes
- ◯ No

Able to view all announcement *

1. Login the account.
2. Clicks on the announcement button on the bottom tab navigation bar.

- ◯ Yes
- ◯ No

Able to view announcement *

1. Login the account.
2. Clicks on the announcement button on the bottom tab navigation bar.
3. The application redirects to announcement page.
4. Clicks on the view more button on announcement with the title "Updated Management Fee Notice."

- ◯ Yes
- ◯ No

Able to view all feedback *

1. Login the account.
2. Clicks on the feedback button on the bottom tab navigation bar.

- ◯ Yes
- ◯ No

Able to view feedback *

1. Login the account.
2. Clicks on the feedback button on the bottom tab navigation bar.
3. The application redirects to feedback page.
4. Clicks on the view more button on the feedback with the title "Rooftop leaking issue".

- ◯ Yes
- ◯ No

Able to add new feedback *

1. Login the account.
2. Click on the feedback button on the bottom tab navigation bar.
3. The application redirects to feedback page.
4. Click on the add button on the feedback page.
5. The application redirects to add feedback page.
6. Enter title, description, categories of feedback. (Title: Complaint on Level 7 Neighbor, Description: Always hear the noise coming from my upstairs after 12am. Categories: others)
7. Clicks on submit button.

◯ Yes

◯ No

Back Next Clear form

User Acceptance and Satisfactory Form

Sign in to Google to save your progress. Learn more

*Required

Share your feedback after performing the test.

This application will help you easier to register your visitor's information. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you to manage your visitor's information. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you easier to get the latest information from management team. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you easier to report your issues to the management team. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

This application will help you to facilitate communication with management teams. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Rate the overall user interface design *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Rate the accuracy of data in the application. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

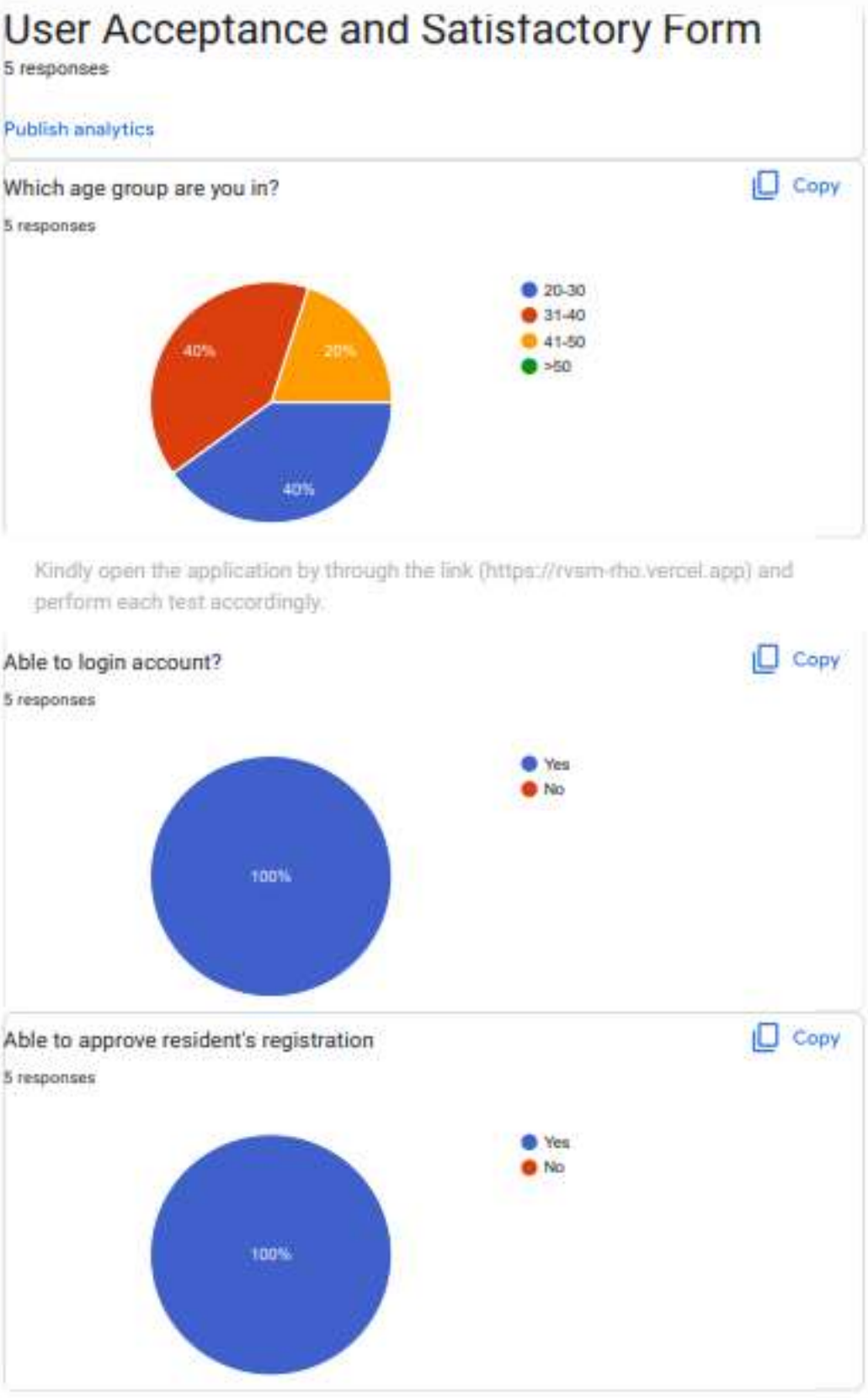
Rate the satisfaction level in the application. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Back Submit Clear form

APPENDIX D: User Acceptance Test Feedback Result

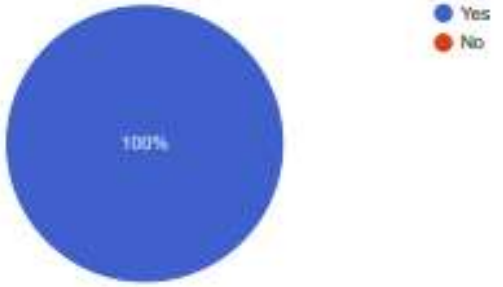
User Acceptance and Satisfactory Form Result (Web application)



Able to reject resident's registration

Copy

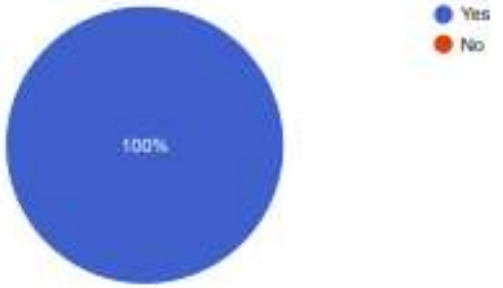
5 responses



Able to view all resident's information

Copy

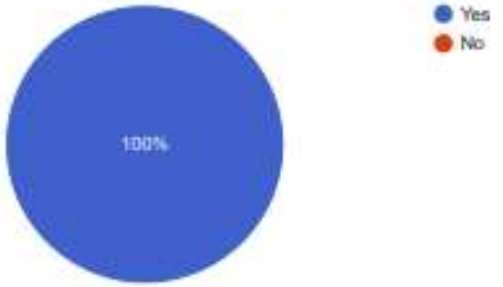
5 responses



Able to view all administrators

Copy

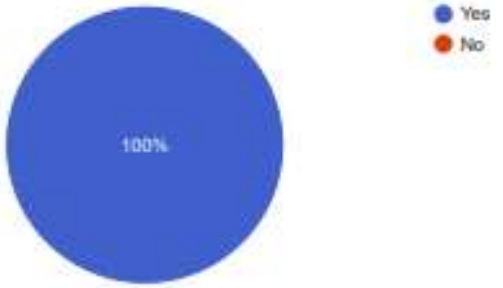
5 responses



Able to add new administrator

Copy

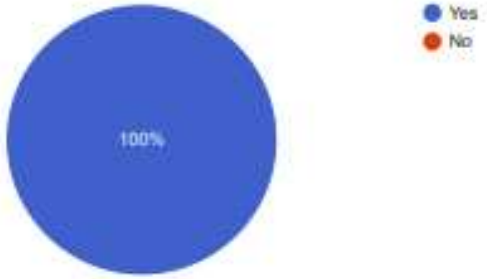
5 responses



Able to remove administrators

Copy

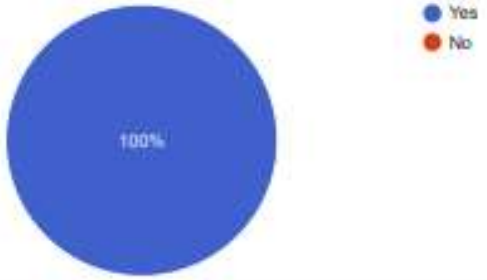
5 responses



Able to view all security guard

Copy

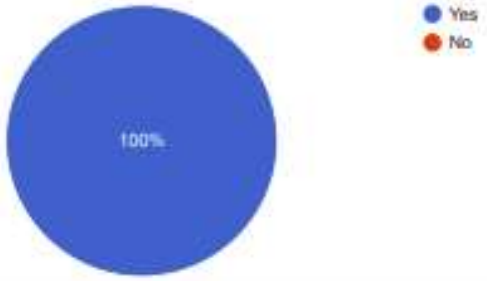
5 responses



Able to add new security guard

Copy

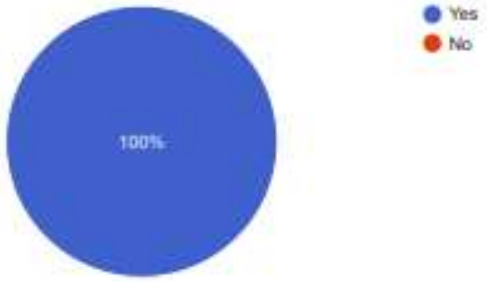
5 responses



Able to remove security guard

Copy

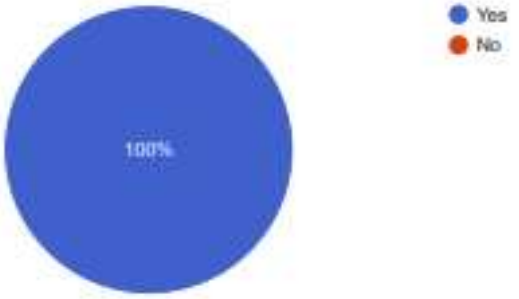
5 responses



Able to modify new password

Copy

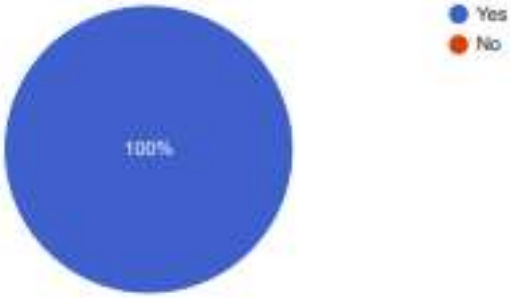
5 responses



Able to view all visitation information

Copy

5 responses



Able to search visitation using unit id

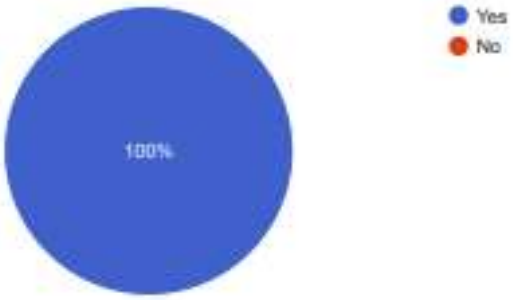
Copy

5 responses

Able to view all announcement

Copy

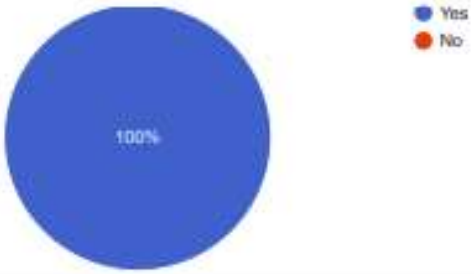
5 responses



Able to add and publish announcement

Copy

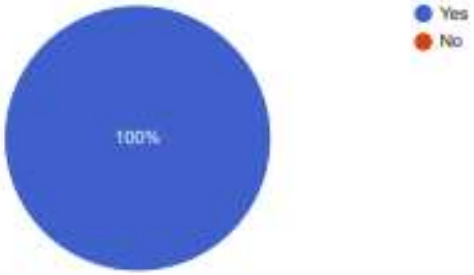
5 responses



Able to add draft announcement

Copy

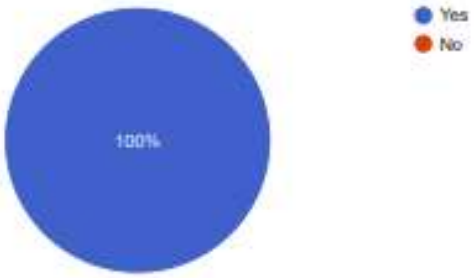
5 responses



Able to update and publish draft announcement

Copy

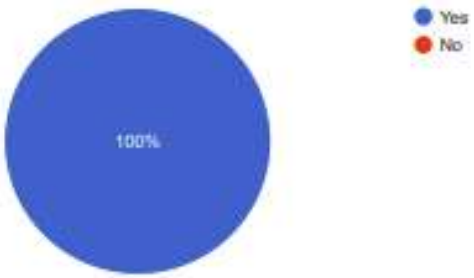
5 responses



Able to view announcement

Copy

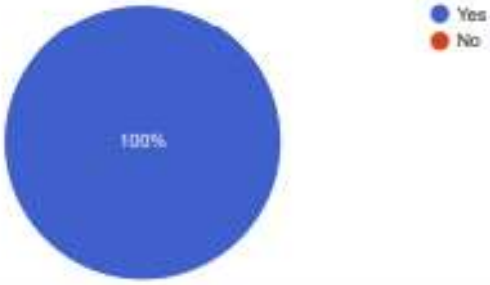
5 responses



Able to remove announcement

Copy

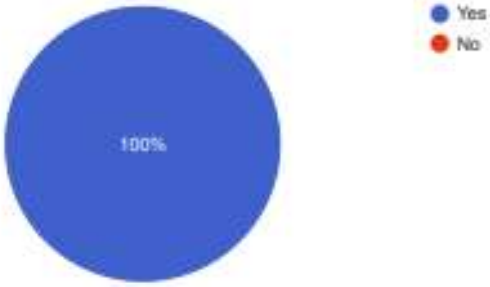
5 responses



Able to view all feedback

Copy

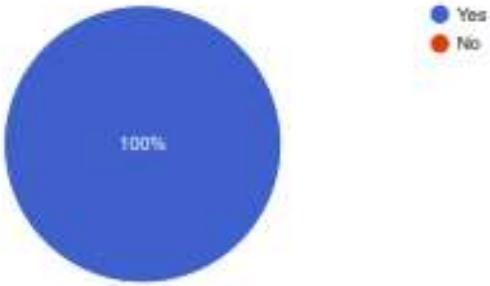
5 responses



Able to reply new feedback

Copy

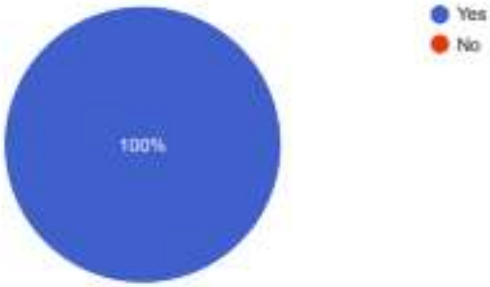
5 responses



Able to view replied feedback

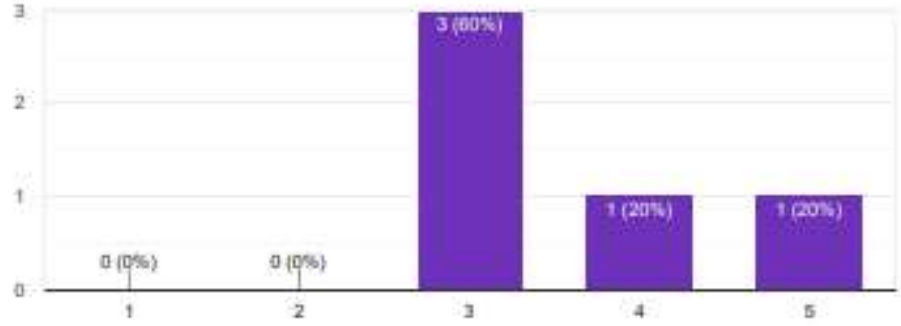
Copy

5 responses



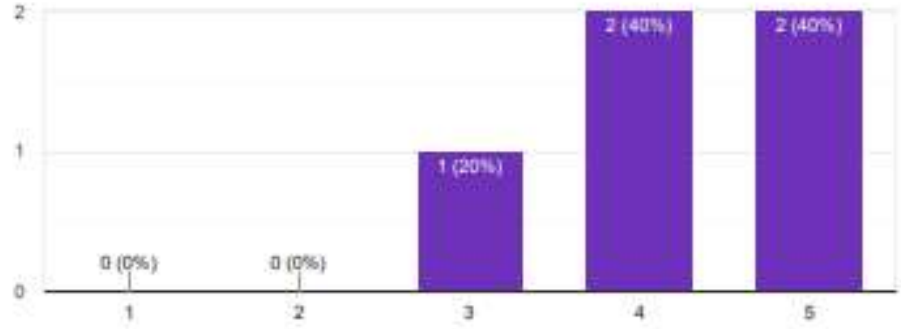
This application will help you to facilitate communication with residents. Copy

5 responses



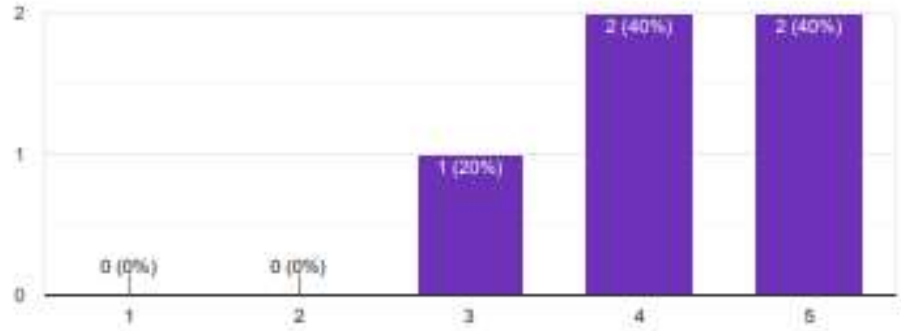
This application will help you easier to manage resident's information. Copy

5 responses



This application will help you easier to manage visitor's information Copy

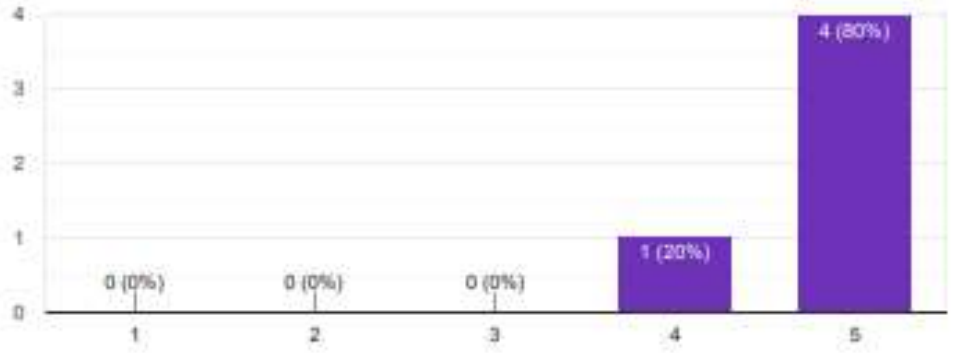
5 responses



This application will help you easier to publish announcement.

Copy

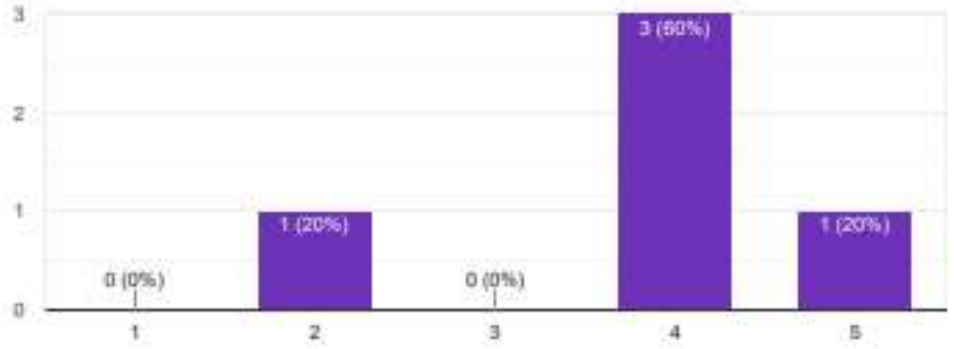
5 responses



This application will help you easier to manage resident's feedback.

Copy

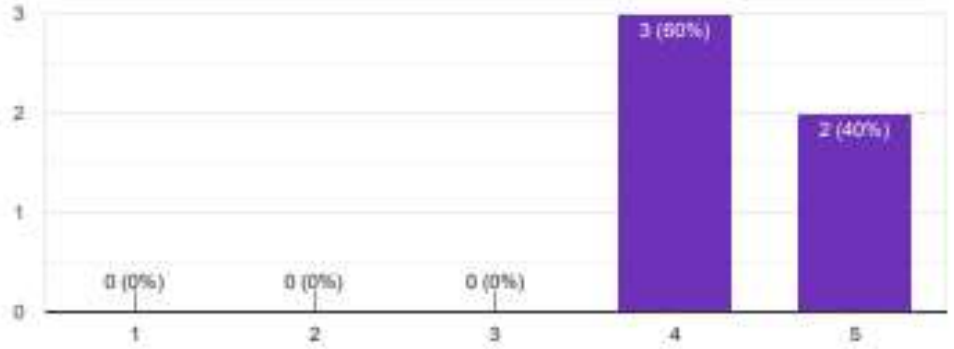
5 responses



This application will help you easier to manage security teams information.

Copy

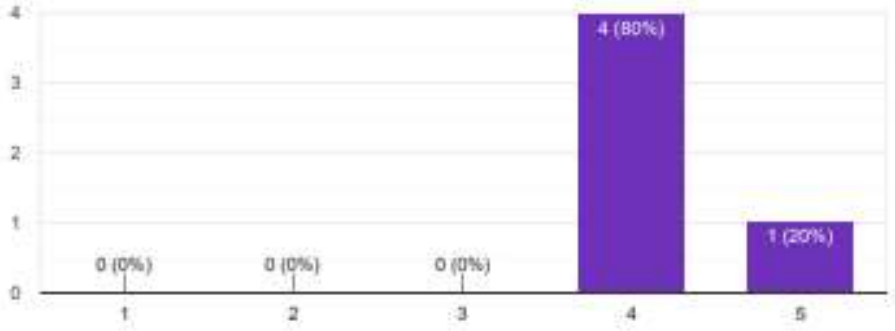
5 responses



This application will help you to simply administrative work.

Copy

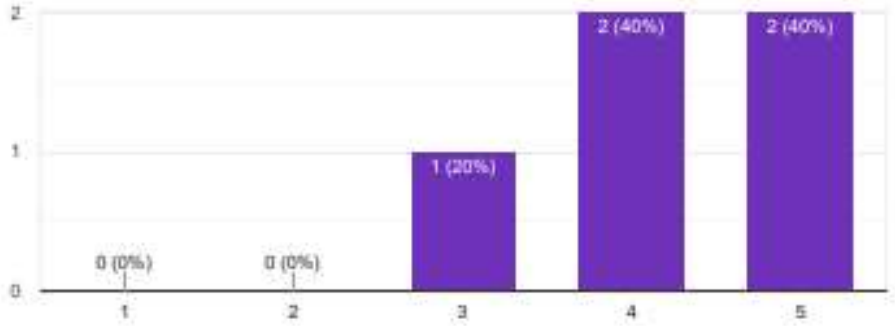
5 responses



Rate the overall user interface design

Copy

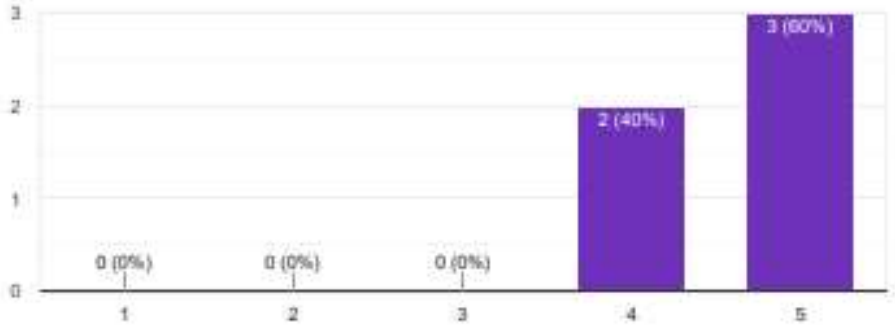
5 responses



Rate the accuracy of data in the application.

Copy

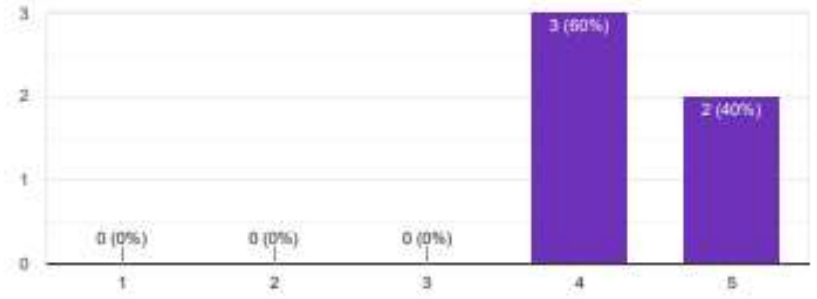
5 responses



Rate the satisfaction level in the application.

Copy

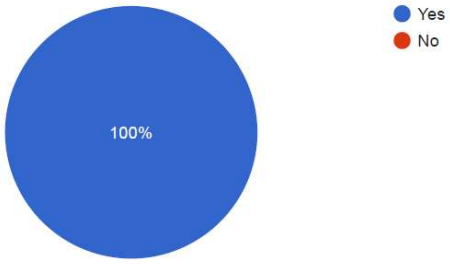
5 responses



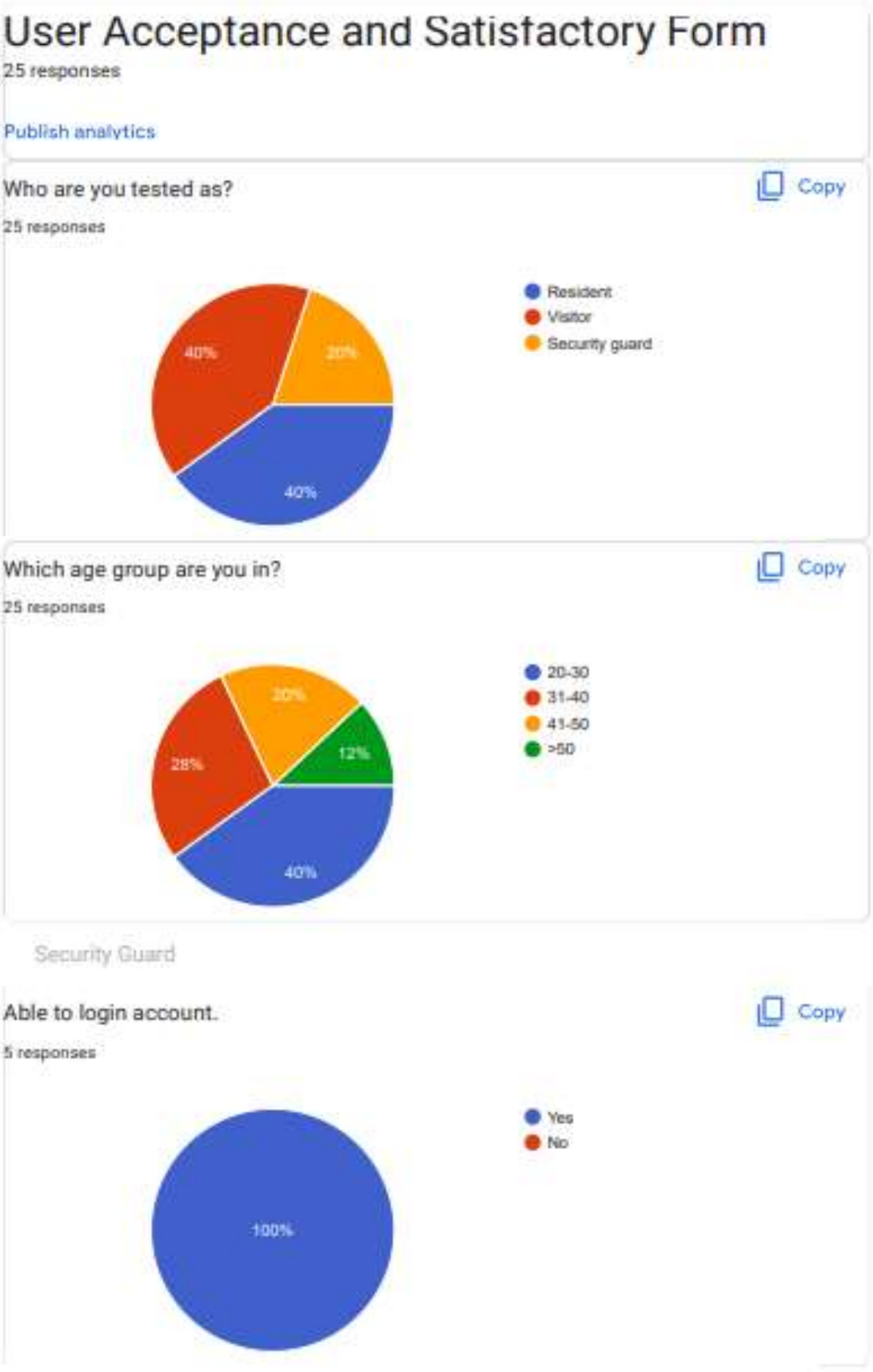
As a user of the web application, would you sign off the user acceptance test?

Copy

5 responses



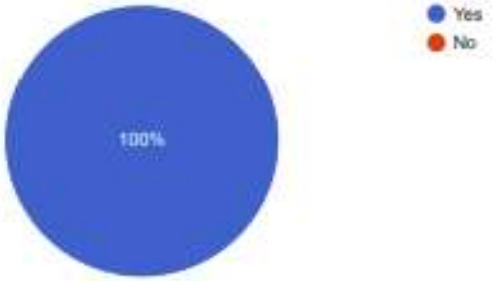
User Acceptance and Satisfactory Form Result (Mobile application)



Able to verify check in visitation?

Copy

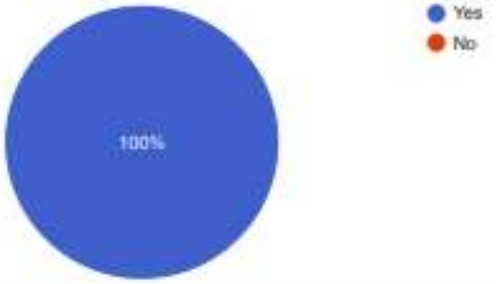
5 responses



Able to add ad-hoc visitors

Copy

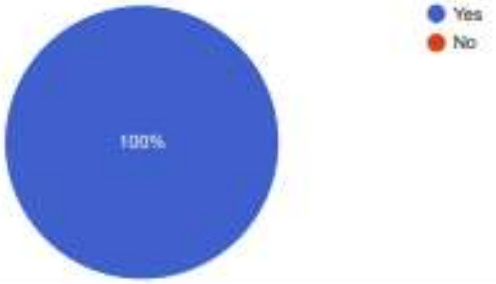
5 responses



Able to logout account

Copy

5 responses



Visitors

Able to check in visitations

Copy

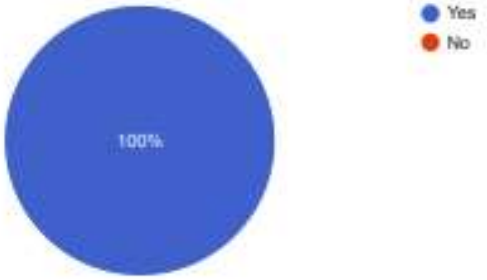
10 responses

Resident

Able to submit registration form

Copy

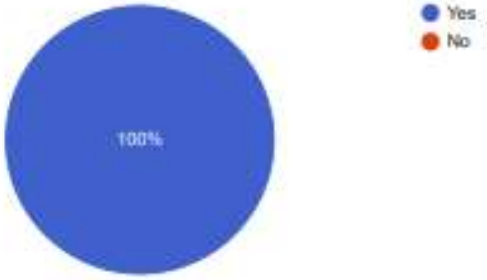
10 responses



Able to login account

Copy

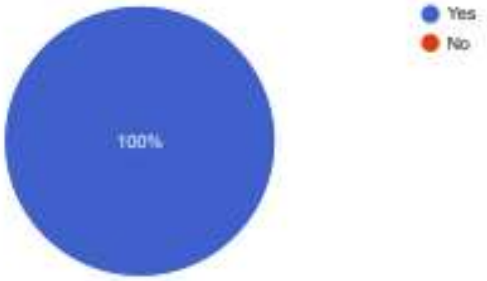
10 responses



Able to logout account

Copy

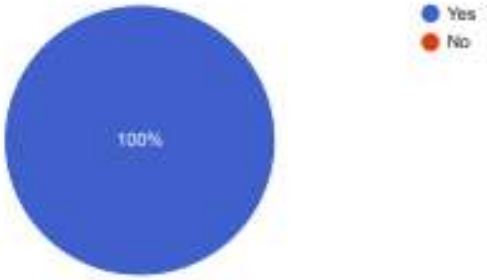
10 responses



Able to modify user profile

Copy

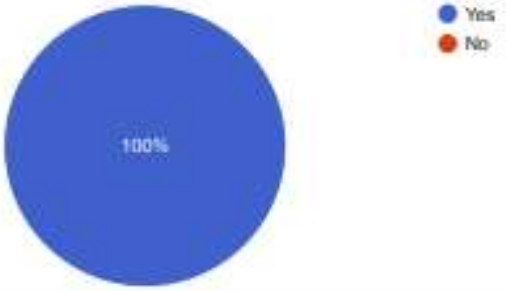
10 responses



Able to add visitor

Copy

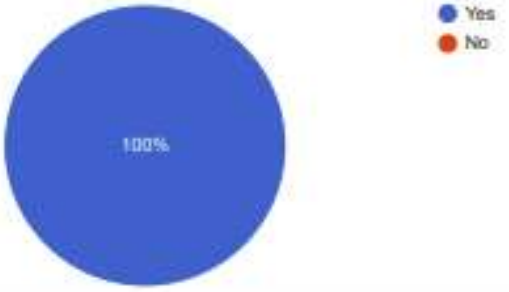
10 responses



Able to view all visitation

Copy

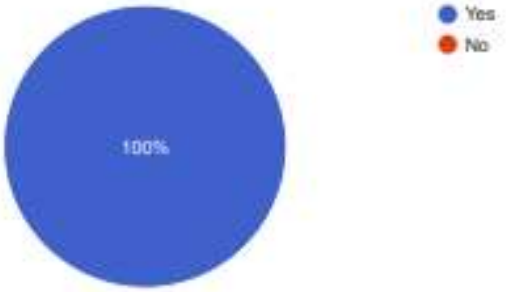
10 responses



Able to view visitation

Copy

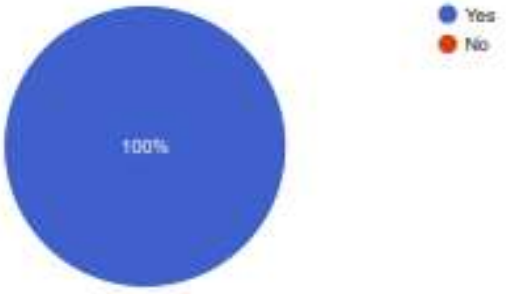
10 responses



Able to remove upcoming visitation

Copy

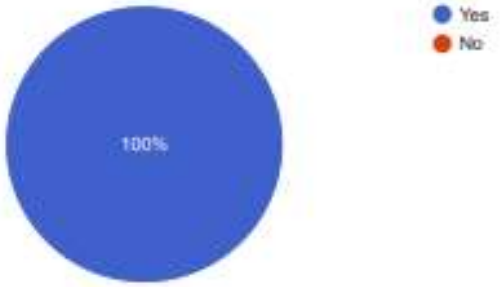
10 responses



Able to view all announcement

Copy

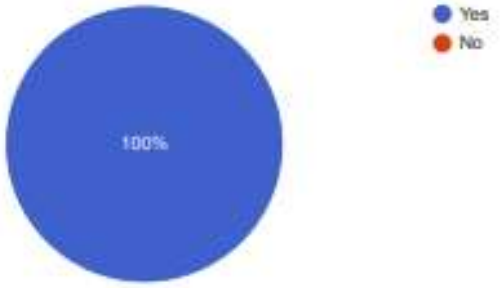
10 responses



Able to view announcement

Copy

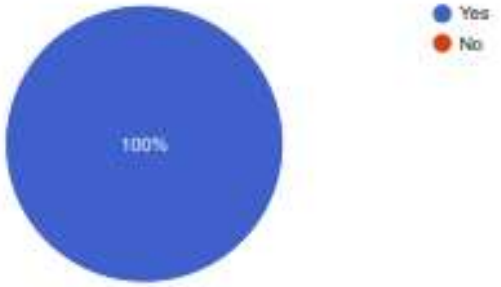
10 responses



Able to view all feedback

Copy

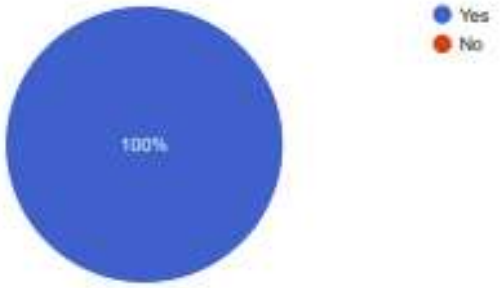
10 responses



Able to view feedback

Copy

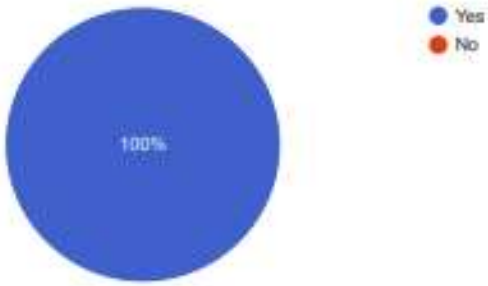
10 responses



Able to add new feedback

Copy

10 responses

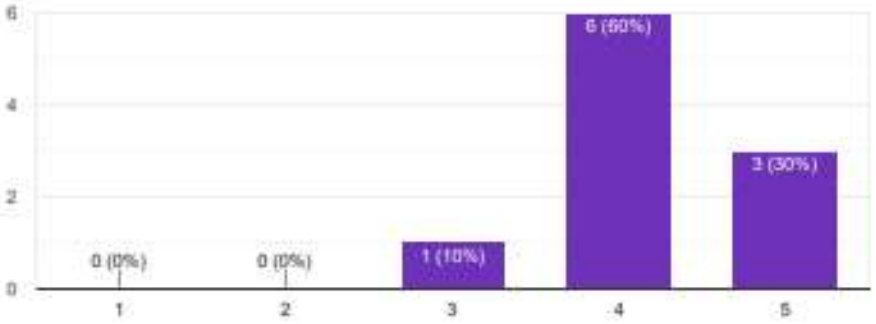


Share your feedback after performing the test.

This application will help you easier to register your visitor's information.

Copy

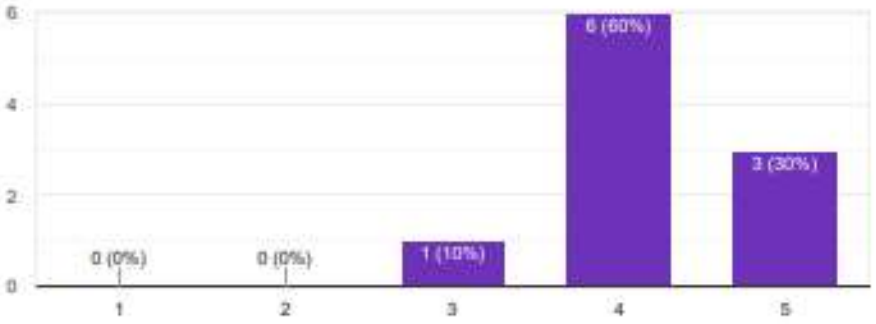
10 responses



This application will help you to manage your visitor's information.

Copy

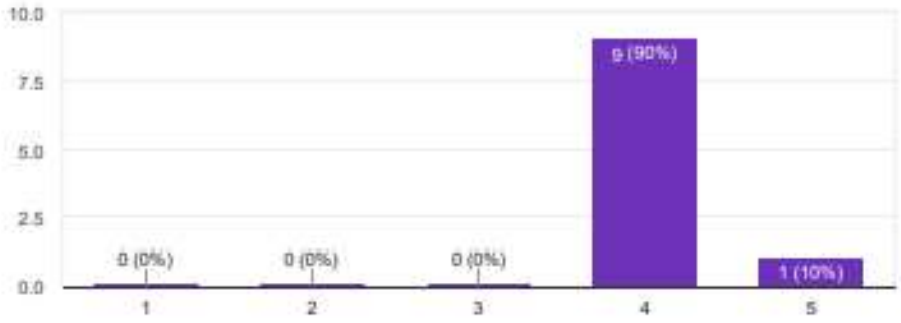
10 responses



This application will help you easier to get the latest information from management team.

Copy

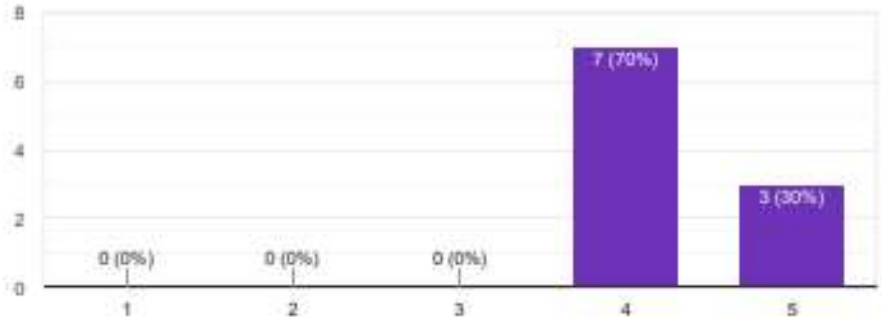
10 responses



This application will help you easier to report your issues to the management team.

Copy

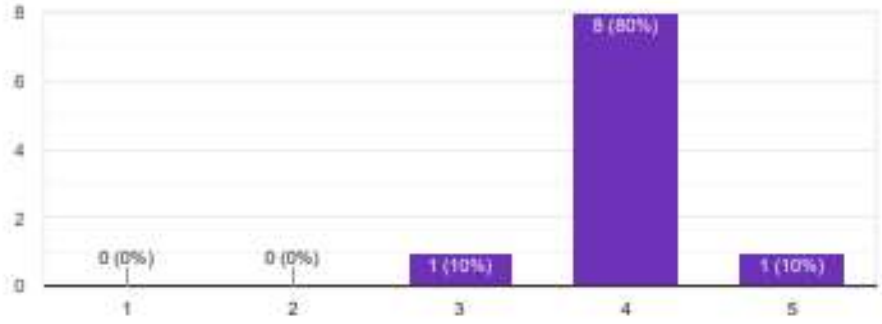
10 responses



This application will help you to facilitate communication with management teams.

Copy

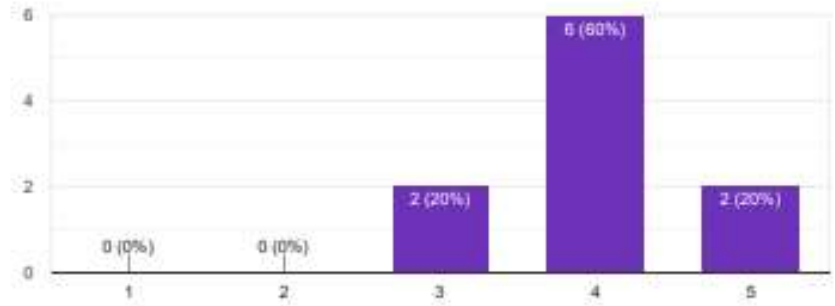
10 responses



Rate the overall user interface design

Copy

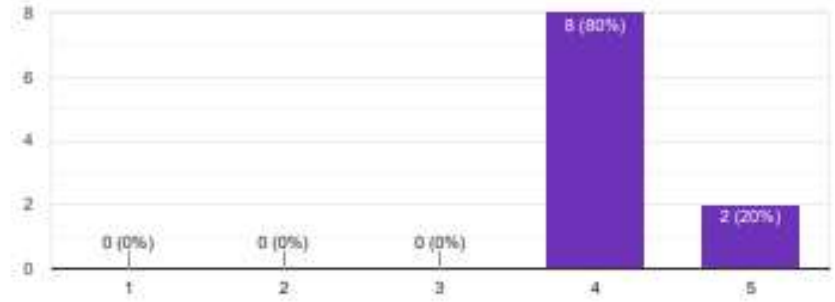
10 responses



Rate the accuracy of data in the application.

Copy

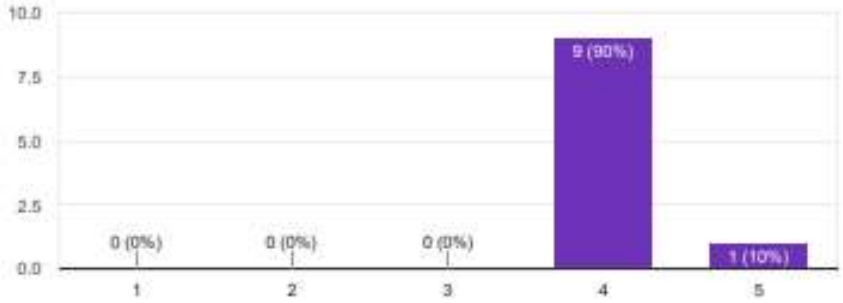
10 responses



Rate the satisfaction level in the application.

Copy

10 responses



As a user of the mobile application, would you sign off the user acceptance test?

Copy

25 responses