# **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	:	Eefor .
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	:	Je.
Supervisor	:	MS. BEH HOOI CHING
Date	:	8 September 2022
Signature	:	
Co-Supervisor	:	
Date	:	

The copyright of this report belongs to the Low Ee Lyne under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2022, Low Ee Lyne. All right reserved.

#### 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 Universiti 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 paperbased 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**

TABLE OF CONTENTS	V
LIST OF TABLES	xi
LIST OF FIGURES	xiii
LIST OF SYMBOLS / ABBREVIATIONS	XX
LIST OF APPENDICES	xxi

# CHAPTER

1	INTR	RODUCTION	1
	1.1	Introduction	1
	1.2	Problem Background	2
	1.3	Problem Statements	3
		1.3.1 Improper Communication Channels	3
		1.3.2 Outdated Data Collection System	4
		1.3.3 Slow Visitor Registration Process	5
	1.4	Project Objectives	6
	1.5	Proposed Solution	6
	1.6	Project Approach	7
		1.6.1 Research Approach	7
		1.6.2 Development Approach	8
	1.7	Scope of Project	10
		1.7.1 Target Users	10
		1.7.2 System Scope	11
		1.7.3 Application Features / Modules	11
2	LITE	RATURE REVIEW	14
	2.1	Introduction	14
	2.2	Similar Systems Review	14
	2.3	Development Framework of Web Application and	
		Mobile Application	18

	2.4	Database used on Web Application and Mobile	
		Application	22
	2.5	Evaluation of Web Application and Mobile	
		Application	27
	2.6	Summary	29
3	METH	HODOLOGY AND WORK PLAN	30
	3.1	Introduction	30
	3.2	Software Development Methodology	30
		3.2.1 Planning	31
		3.2.2 Analysis and Design	34
		3.2.3 Development and Testing	35
		3.2.4 Closing	36
	3.3	Development Tools	36
		3.3.1 Axure RP9	36
		3.3.2 Visual Studio Code	37
		3.3.3 React Native	37
		3.3.4 ReactJS	37
		3.3.5 Next.js	38
		3.3.6 Supabase	38
		3.3.7 Expo	38
		3.3.8 Git	39
	3.4	Work Breakdown Structure (WBS)	39
	3.5	Gantt Chart	48
		3.5.1 Overview	48
		3.5.2 Planning Phase	48
		3.5.3 Analysis and Design Phase	49
		3.5.4 Development Phase 1	50
		3.5.5 Development Phase 2	50
		3.5.6 Development Phase 3	52
		3.5.7 Closing Phase	53
	3.6	Summary	54
4	PROJ	ECT SPECIFICATION	55
	4.1	Introduction	55
	4.2	Fact finding	55

	4.2.1 Responses of Questionnaire from	
	Management Teams	56
	4.2.2 Responses of Questionnaire from Resident	68
4.3	Requirement Specification	76
	4.3.1 Functional Requirement	76
	4.3.2 Non-Functional Requirement	79
4.4	Use Case Modelling	80
	4.4.1 Use Case Diagram	80
	4.4.2 Use Case Description	82
4.5	Interface Flow Diagram	108
	4.5.1 Web Application	108
	4.5.2 Mobile Application	109
4.6	Prototype interface	111
	4.6.1 Web Application	111
	4.6.2 Mobile Application	124
4.7	Summary	135
SYST	TEM DESIGN	136
5.1	Introduction	136
5.2	System Architecture Design	136
	5.2.1 Front end architecture	137
	5.2.2 Back end architecture	139
5.3	Database architecture	141
	5.3.1 Database Schema	141
	5.3.2 Table Description	142
	5.3.3 Data Flow Diagram	143
5.4	User Interface Designs	148
	5.4.1 Web Application	148
	5.4.2 Mobile Application	153
SYSY	YEM IMPLEMENTATION	173
6.1	Introduction	173
6.2	Implementation of Supabase	173
	6.2.1 Postgrest-js	174
	6.2.2 Gotrue-js	174
	6.2.3 Realtime-js	175

5

6

		6.2.4 Storage is	175
	63	Authentication and authorization	176
	0.5	6.3.1 Authentication	176
		6.2.2 Authorization	176
	<i>с</i> <b>н</b>		170
	6.4	Front end libraries used	177
	6.5	Deployment	178
	6.6	Summary	179
7	SYST	EM TESTING	180
	7.1	Introduction	180
	7.2	Traceability between Use Cases, Functional	
		Requirements and Test Cases	180
		7.2.1 Use Case Table	180
		7.2.2 Functional Requirement Table	181
		7.2.3 Test Cases Table of Unit Testing	183
		7.2.4 Test Cases Table for Integration Testing	187
		7.2.5 Tracebility Matrix	190
	7.3	User Acceptance Test	192
		7.3.1 User Acceptance Test Plan	192
		7.3.2 User Acceptance Test Cases	193
		7.3.3 User Acceptance Test Result	197
	7.4	Summary	201
8	CONC	CLUSION AND RECOMMEDATIONS	202
	8.1	Introduction	202
	8.2	Objective Examination	202
	8.3	Limitations	203
	8.4	Recommendation for Future Work	204
REFER	ENCES	5	207
APPEN	DICES		212

# LIST OF TABLES

Table 2.1 Features Provided by Similar Systems	15
Table 2.2 Summary of Measured Metrics Compared in Crha's Study	18
Table 2.3 Comparison of SQL and NoSQL	22
Table 2.4 Comparison between Supabase and Firebase	26
Table 4.1 Overview of Functional Requirement ID and Related Use Case ID	78
Table 4.2 Use Case of Login Account	82
Table 4.3 Use Case of Manage Resident's Registration	83
Table 4.4 Use Case of Track Resident's Information	84
Table 4.5 Use Case of Manage Administrators	86
Table 4.6 Use Case of Manage Security Guards	88
Table 4.7 Use Case Of Modify User Profile	90
Table 4.8 Use Case of Track Visitor's Recors	91
Table 4.9 Use Case of Manage Announcements	92
Table 4.10 Use Case of Manage Feedback	94
Table 4.11 Use Case of Submit Registration Form	96
Table 4.12 Use Case of Login Account	97
Table 4.13 Use Case of Modify User Profile	99
Table 4.14 Use Case of Register Visitors	100
Table 4.15 Use Case of Manage Registered Visitations	101
Table 4.16 Use Case of Check in Visitations	103
Table 4.17 Use Case of Verify Visitations	104
Table 4.18 Use Case of View Announcements	105
Table 4.19 Use Case of Manage Feedback	106

Table 5.1 Table Description	142
Table 6.1 Postgrest-js	174
Table 6.2 Gotrue-js	174
Table 6.3 Realtime-js	175
Table 6.4 Storage-js	176
Table 6.5 Deployment process	178
Table 7.1 Use Cases	180
Table 7.2 Functional requirements	181
Table 7.3 Unit testing test cases (web application)	183
Table 7.4 Unit testing test cases (mobile application)	185
Table 7.5 Integration testing test cases (web application)	187
Table 7.6 Integration testing test cases (mobile application)	189
Table 7.7 Tracibility matrices	190
Table 7.8 UAT test cases (web application)	193
Table 7.9 UAT test cases (mobile application)	195
Table 7.10 UAT feedback result summary (web application)	198
Table 7.11 UAT feedback result summary (web application)	198
Table 7.12 UAT participants user type	199
Table 7.13 UAT participants age group	199
Table 7.14 UAT Feedback Result Summary (mobile application)	200
Table 7.15 UAT Feedback Result Summary (mobile application)	201
Table 8.1 Recommendations	205

# LIST OF FIGURES

Figure 1.1 Overview of the Proposed solution	7
Figure 1.2 Phased Developmen-Baseed Methodology (Dennis, et al., 2015)	9
Figure 3.1 Phased Development Methodology of Resident and Visitor Maanagement System	31
Figure 3.2 Overview Of Project Schedule	48
Figure 3.3 Planning Phase Schedule	48
Figure 3.4 Planning Phase Schedule (cont.)	49
Figure 3.5 Planning Phase Schedule (cont.)	49
Figure 3.6 Analysis and Design Phase Schedule	49
Figure 3.7 Development Phase 1 Schedule	50
Figure 3.8 Development Phase 2 Schedule	50
Figure 3.9 Development Phase 2 Schedule (cont.)	51
Figure 3.10 Development Phase 2 Schedule (cont.)	51
Figure 3.11 Development Phase 2 Schedule (cont.)	52
Figure 3.12 Development Phase 2 Schedule (cont.)	52
Figure 3.13 Development Phase 3 Schedule	52
Figure 3.14 Development Phase 3 Schedule (cont.)	53
Figure 3.15 Development Phase 3 Schedule (cont.)	53
Figure 3.16 Closing Phase Schedule	53
Figure 4.1 Age of Respondents	56
Figure 4.2 Gender of Respondents	56
Figure 4.3 Role of Respondent in Management Teams	57
Figure 4.4 Collect Resident Information	57

Figure 4.5 Methods Used to Collect Information Of Resident	58
Figure 4.6 Methods of Manage and Organize Resident Information	58
Figure 4.7 Methods of Comminucation with Residents	59
Figure 4.8 Satisfaction of Respondents on Current Collecting and Managing Process	60
Figure 4.9 Methods of Respondents Used to Publish or Broadcast Announcement	60
Figure 4.10 Thought on Barriers of Communication Method Used	61
Figure 4.11 Barriers of Communication Method faced	62
Figure 4.12 Satisfaction of Respondents on the Current Communication Methods Used	62
Figure 4.13 Methods Used te Register Visitor Records	63
Figure 4.14 Satisfaction of Respondents with The Current Register Visitor Process	63
Figure 4.15 Inconveniences Faced when Using the Current Register Visitor Method	64
Figure 4.16 Any Management System Used by Respondents	65
Figure 4.17 Example of Managment Applications	65
Figure 4.18 Opinion of Respondents on Web Application or Mobile Application	66
Figure 4.19 Features Expected by The Respondents	66
Figure 4.20 Other Suggestion of Feature oof Management Application	67
Figure 4.21 Age of Respondent (Residents)	68
Figure 4.22 Gender of Respondents (Residents)	68
Figure 4.23 Communication Method Of Residents with Management Team	69
Figure 4.24 Methods of Respondents Learn the News of The Community	69
Figure 4.25 Satisfaction of Residents on the Current Communication Method Using	70

Figure 4.26 Opinion of Barriers of Communication Method Used	71
Figure 4.27 Barriers of Communication Method Using Faced	71
Figure 4.28 Method Used by Respondents to Register Visitor	72
Figure 4.29 Satisfaction of Current Register Visitor Method	72
Figure 4.30 Incoveniences Faced when Register Visitors	73
Figure 4.31 Usage of Any Management Applications	73
Figure 4.32 Examples of Management Application Used	74
Figure 4.33 Opinions of Respondents on Web Applications and Mobile Application	74
Figure 4.34 Features Expected by Respondents on the System	75
Figure 4.35 Use Case Diagram (Web Application)	80
Figure 4.36 Use Case Diagram (Mobile Application)	81
Figure 4.37 Interface Flow Diagram of Web Application	108
Figure 4.38 Interface Flow Diagram of Mobile Application (Resident)	109
Figure 4.39 Interface Flow Diagram of Mobile Application (Visitor)	109
Figure 4.40 Interface Flow Diagram of Mobile Application (Security guard)	110
Figure 4.41 Login Page (Web)	111
Figure 4.42 Home Page (Web)	111
Figure 4.43 New Resident Registration Page	112
Figure 4.44 Existing Resident Page	112
Figure 4.45 Visitor Page	113
Figure 4.46 Total Visitor List Page	113
Figure 4.47 Administrators Page	114
Figure 4.48 Add New Administrator Page	114
Figure 4.49 Security Guard Page	115

Figure 4.50 Add New Security Guard Page	115
Figure 4.51 Announcements Page	116
Figure 4.52 View Announcement Page	117
Figure 4.53 Update Announcement Page	118
Figure 4.54 Add New Announcement Page	119
Figure 4.55 New Feedback Page	120
Figure 4.56 Replied Feedback Page	121
Figure 4.57 Reply Selected Feedback Page	122
Figure 4.58 Modify User Profile Page	123
Figure 4.59 Login or Register Page	124
Figure 4.60 Register as Resident Page	125
Figure 4.61 Register as Resident Page (cont.)	125
Figure 4.62 Login as Resident Page	126
Figure 4.63 Home Page	126
Figure 4.64 Visitation Page	127
Figure 4.65 Register Visitors Page	127
Figure 4.66 View Visitation Details Page	128
Figure 4.67 Announcements Page	128
Figure 4.68 View Announcement Details Page	129
Figure 4.69 Feedback Page	130
Figure 4.70 Add Feedback Page	130
Figure 4.71 View Feedback Detail Page	131
Figure 4.72 Modify User Profile Page	131
Figure 4.73 Use as Visitors Page	132
Figure 4.74 Check-in Page	132

Figure 4.75 Login as Security Guard Page	133
Figure 4.76 Home Page	133
Figure 4.77 Verify Check-in Page	134
Figure 4.78 Confirm Verify Visitation Page	134
Figure 5.1 Separate UI Server Architecture	136
Figure 5.2 Server Side Rendering	138
Figure 5.3 React Native Architecture	138
Figure 5.4 Supabase Architecture	139
Figure 5.5 Database Schema	141
Figure 5.6 Context Diagram	143
Figure 5.7 Data Flow Diagram Level-0 (Web application)	144
Figure 5.8 Data Flow Diagram Level-0 (Mobile Application)	144
Figure 5.9 Data Flow Diagram Level-1 (2.0 Manage Resident's Registration)	145
Figure 5.10 Data Flow Diagram Level-1 (3.0 Track Resident's information)	145
Figure 5.11 Data Flow Diagram Level-1 (4.0 Manage Administrator)	145
Figure 5.12 Data Flow Diagram Level-1 (5.0 Manage Security Guards)	146
Figure 5.13 Data Flow Diagram Level-1 ( 8.0 Manage Announcement)	146
Figure 5.14 Data Flow Diagram Level-1 (9.0 Manage Feedback)	147
Figure 5.15 Data Flow Diagram Level-1 (5.0 Manage Registered Visitation)	147
Figure 5.16 Data Flow Diagram Level-1 (9.0 Manage Feedback)	147
Figure 5.17 Login Page	148
Figure 5.18 Home Page	148
Figure 5.19 New Residents Registration Page	149
Figure 5.20 Exisitng Resident Page	149

Figure 5.21 Today Visitor's Page	149
Figure 5.22 All Visitation Page	150
Figure 5.23 Adminsitrators Page	150
Figure 5.24 Security Guards Page	150
Figure 5.25 Announcements Page	151
Figure 5.26 Add New Announcement Page	151
Figure 5.27 Edit Announcement Page	151
Figure 5.28 New Feedback Page	152
Figure 5.29 Replied Feedback Page	152
Figure 5.30 Edit Profile Page	152
Figure 5.31 Welcome Page	153
Figure 5.32 Login as Resident Page	154
Figure 5.33 Signup Page	155
Figure 5.34 Home Page	156
Figure 5.35 Upcoming Visitor Page	157
Figure 5.36 Visitation Hitory Page	158
Figure 5.37 Add New Visitor Page	159
Figure 5.38 View Visitor Page	160
Figure 5.39 Announcement Page	161
Figure 5.40 View Feedback Page	162
Figure 5.41 View Pending Feedback Page	163
Figure 5.42 View Replied Feedback Page	164
Figure 5.43 Add New Feedback Page	165
Figure 5.44 View Announcement Page	166
Figure 5.45 Login as Visitor Page	167

xviii

Figure 5.46 Check in Visitation Page	168
Figure 5.47 Login as Security Guard	169
Figure 5.48 Home Page	170
Figure 5.49 Verify Check in Page	171
Figure 5.50 Add New Visitor Page (Security Guard View)	172
Figure 6.1 Building Blocks of the Supabase JavaScript Client Libraries	173
Figure 6.2 Screenshot of Supabase screenshot (RLS enabled and policy added)	177
Figure 6.3 Show Vercel Dashboard	179
Figure 6.4 Show The Successfully Deployment of The Web Application	179

# 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

APPENDIX A:	Similar Systems Review	212
APPENDIX B:	Test Cases	218
APPENDIX C:	User Acceptance Test Questionnaire	300
APPENDIX D:	User Acceptance Test Feedback Result	321

#### **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 satisfaction improves all parties' (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 faceto-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 paperbased 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 userfriendly 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 7 out of 16 respondents from the management side faced traffic found. 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 crossplatform 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 causalcomparative 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.

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

Table 2.1 Features Provided by Similar Systems

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

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

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.

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

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

		(perform better than Cross-				
		Platform Mobile				
		Application in some				
		screens)				
Memory Usage	Higher memory	Lower memory				
	consumption	consumption				
	(React Native consume					
	lesser memory than Flutter)					
Frames per	On Android, React Native	Not applicable				
Second	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					
Application Size	On Android: Flutter smaller	On Android: Smallest Size				
	than React Native.	compared to Cross-				
		platform Mobile				
	On iOS: React Native	Application				
	smaller than Flutter	On Android: Smallest Size compared to Cross- platform Mobile Application On iOS: Not determined Allow to share close to 100% of the code between				
Code Sharing	Allow to share close to	Allow to share close to				
	100% of the code between	100% of the code between				
	Android and iOS.	Android and iOS.				
Codebase Size	React Native's smaller than	Not applicable				
	Flutter's.					
Build Time	On Android: Flutter faster	On Android: Flutter faster				
	than React Native	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	Better user experience	Worst user experience
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.

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

Table 2.3 Comparison of SQL and NoSQL

Language	- SQL language	- The language used is
		determined by the type of
		NoSQL database, the
		implementation, and the
		operation.
Schemas	- Requires a specified schema	- Makes use of a dynamic
	that dictates data is stored and	schema that eliminates the
	how tables are set up, causing	need for a predefined data
	a strict structure.	structure.
	- Low flexibility	- 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	- Inconsistency among
	and combining data from	products necessitates extra
	several tables.	effort when querying data.
Maturity	- Built on mature	- Built on less mature
	technologies	technologies as well as
	- Backed up by large	supported as SQL
	developer communities	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.

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.

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

Table 2.4 Comparison between Supabase and Firebase

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 webbased 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 thinkaloud 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 eve-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 thinkaloud 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 subphases. 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.

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
102	

- 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

- 4.1.4.3 Implement Algorithm to Retrieve Selected Registration Request
- 4.1.4.4 Implement Algorithm to Update Selected Registration Request's Status
- 4.1.5 Test Manage Resident's Registration Feature
  - 4.1.5.1 Test Algorithm to Read All Registration Request
  - 4.1.5.2 Test Algorithm to Retrieve Selected Registration Request
  - 4.1.5.3 Test Algorithm to Update Selected Registration Request's Status
- 4.1.6 Create Track Resident's Information Feature
  - 4.1. 6.1 Create Residents's Information User Interface
  - 4.1. 6.2 Implement Algorithm to Retrieve All Resident's Information
  - 4.1. 6.3 Implement Algorithm to Retrieve Selected Resident's Information
  - 4.1. 6.4 Implement Algorithm to Delete Selected Resident's Information
- 4.1.7 Test Track Resident's Information Feature
  - 4.1.7.1 Test Algorithm to Retrieve All Resident's Information
  - 4.1.7.2 Test Algorithm to Retrieve Selected Resident's Information
  - 4.1.7.3 Test Algorithm to Delete Selected Resident's Information
  - 4.1.8 Create Manage Administrators Feature
    - 4.1.8.1 Create Manage Administrators User Interface
    - 4.1.8.2 Implement Algorithm to Retrieve All Administrators
    - 4.1.8.3 Implement Algorithm to Add New Administrator

- 4.1.8.4 Implement Algorithm to Remove Administrator
- 4.1.9 Test Manage Administrators Feature
  - 4.1.9.1 Test Algorithm to Retrieve All Administrators
  - 4.1.9.2 Test Algorithm to Add New Administrator
  - 4.1.9.3 Test Algorithm to Remove Administrator
- 4.1.10 Create Manage Security Guard Feature
  - 4.1.10.1 Create Security Guards List User Interface
  - 4.1.10.2 Implement Algorithm to Read All Security Guards
  - 4.1.10.3 Implement Algorithm to Add New Security Guard
  - 4.1.10.4 Implement Algorithm to Remove Security Guard

#### 4.1.11 Test Manage Security Guard Feature

- 4.1.11.1 Test Algorithm to Read All Security Guards
- 4.1.11.2 Test Algorithm to Add New Security Guard
- 4.1.11.3 Test Algorithm to Remove Security Guard
- 4.1.12 Create Modify User Profile Information Feature
  - 4.1.12.1 Create User Profile Interface
  - 4.1.12.2 Implement Algorithm to Update Administrator's Information
- 4.1.13 Test Modify User Profile Information Feature
  - 4.1.13.1 Test Algorithm to Update Administrator's Information
- 4.1.14 Create Track Visitation's Records Feature
  - 4.1.14.1 Create Visitation's Records User Interface

- 4.1.14.2 Implement Algorithm to Retrieve All Visitation Records
- 4.1.14.3 Implement Algorithm to Retrieve Selected Visitation Records
- 4.1.15 Test Track Visitation's Records Feature
  - 4.1.15.1 Test Algorithm to Retrieve All Visitation Records
  - 4.1.15.2 Test Algorithm to Retrieve Selected Visitation Records
- 4.1.16 Create Manage Announcements Feature
  - 4.1.16.1 Create Announcements User Interface
  - 4.1.16.2 Implement Algorithm to Add New Announcement
  - 4.1.16.3 Implement Algorithm to Retrieve All Announcement
  - 4.1.16.4 Implement Algorithm to Update Selected Announcement
  - 4.1.16.5 Implement Algorithm to Delete Announcement
- 4.1.17 Test Manage Announcement Feature
  - 4.1.17.1 Test Algorithm to Add New Announcement
  - 4.1.17.2 Test Algorithm to Retrieve All Announcement
  - 4.1.17.3 Test Algorithm to Update Selected Announcement
  - 4.1.17.4 Test Algorithm to Delete Announcement
- 4.1.18 Create Manage Feedback Feature
  - 4.1.18.1 Create Feedback Management User Interface
  - 4.1.18.2 Implement Algorithm to Retrieve All Feedback
  - 4.1.18.3 Implement Algorithm to Retrieve Selected Feedback

- 4.1.18.4 Implement Algorithm to Provide Reply to Selected Feedback
- 4.1.19 Test Receive Feedback Feature
  - 4.1.19.1 Test Algorithm to Retrieve All Feedback
  - 4.1.19.2 Test Algorithm to Retrieve Selected Feedback
  - 4.1.19.3 Test Algorithm to Provide Reply to Selected Feedback
- 4.1.20 Combine All Developed Features

#### 4.2 Test Web Application

- 4.2.1 Test the System's Flow
- 5.0 Development Phase Three
  - 5.1 Develop Mobile Application
    - 5.1.1 Create Mobile Application Framework
    - 5.1.2 Create Submit Registration Form Feature
      - 5.1.2.1 Create Submit Registration Form User Interface
      - 5.1.2.2 Implement Algorithm to Submit Registration Form
    - 5.1.3 Test Submit Registration Form Feature
      - 5.1.3.1 Test Algorithm to Submit Registration Form
    - 5.1.4 Create Login Feature
      - 5.1.4.1 Create Login User Interface
      - 5.1.4.2 Implement Algorithm to Allow User to Login
    - 5.1.5 Test Login Feature
      - 5.1.5.1 Test Algorithm to Allow User to Login
    - 5.1.6 Create Modify User Profile Feature
      - 5.1.6.1 Create Modify User Profile User Interface
      - 5.1.6.2 Implement Algorithm to Update User Profile

- 5.1.7 Test Modify User Profile Feature
  - 5.1.7.1 Test Algorithm to Update User Profile
- 5.1.8 Create Register Visitor Feature
  - 5.1.8.1 Create Visitor Registration User Interface
  - 5.1.8.2 Implement Algorithm Register Visitor
- 5.1.9 Test Register Visitor Feature
  - 5.1.9.1 Test Algorithm to Allow Resident to Register Visitor
- 5.1.10 Create Manage Registered Visitations Feature
  - 5.1.10.1 Create Registered Visitations User Interface
  - 5.1.10.2 Implement Algorithm to Read All Visitations
  - 5.1.10.3 Implement Algorithm to Retrieve Specific Visitation
  - 5.1.10.4 Implement Algorithm to Remove Specific Visitation
- 5.1.11 Test Manage Registered Visitation Feature
  - 5.1.11.1 Test Algorithm to Read All Visitations
  - 5.1.11.2 Test Algorithm to Retrieve Specific Visitation
  - 5.1.11.3 Test Algorithm to Remove Specific Visitation
- 5.1.12 Create Check in Visitation Feature
  - 5.1.12.1 Crate Check in Visitation User Interface
  - 5.1.12.2 Implement Algorithm Check in Visitation
- 5.1.13 Test Check in Visitation Feature
  - 5.1.13.1 Test Algorithm Check in Visitation
- 5.1.14 Create Verify Visitation's Check in Feature
  - 5.1.14.1 Create Visitor's Check in Verification User Interface

- 5.1.14.2 Implement Algorithm to Verify Visitor's Check in
- 5.1.15 Test Verify Visitation's Check in Feature
  - 5.1.15.1 Test Algorithm to Verify Visitor's Check in
- 5.1.16 Create View Announcements Feature
  - 5.1.16.1 Create View Announcement User Interface
  - 5.1.16.2 Implement Algorithm to Read All Announcements
  - 5.1.16.3 Implement Algorithm to Retrieve Specific Announcement
- 5.1.17 Test Receive Announcements Feature
  - 5.1.17.1 Test Algorithm to Read All Announcements
  - 5.1.17.2 Test Algorithm to Retrieve Specific Announcement

#### 5.1.18 Create Manage Feedback Feature

- 5.1.18.1 Create Manage Feedback User Interface
- 5.1.18.2 Implement Algorithm to Add Feedback
- 5.1.18.3 Implement Algorithm to View All Feedback
- 5.1.18.4 Implement Algorithm to View Specific Feedback
- 5.1.19 Test Manage Feedback Feature
  - 5.1.19.1 Test Algorithm to Add Feedback
  - 5.1.19.2 Test Algorithm to View All Feedback
  - 5.1.19.3 Test Algorithm to View Specific Feedback
- 5.1.20 Combine All Developed Features
- 5.2 Test Mobile Application
  - 5.2.1 Test the System's Flow

# 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

	O Search tacks	START	WDT	DUET	6	2			Feb	202	2		Mar	202	22	
	Search tasks	210011	1101	DOLT		24	31	07	14	21	28	07	14	21	28	04
•	1.0 Planning:	24/Jan	60d	15/Apr												
Ð	2.0 Analysis and Design:	21/Mar	15d	08/Apr										-		-
<b>H</b>	3.0 Development Phase One:	01/May	6d	09/May												
•	4.0 Development Phase Two:	24/Apr	49d	30/Jun												
Đ	5.0 Development Phase Three:	01/Jul	35d	18/Aug												
Đ	6.0 Closing:	19/Aug	11d	03/Sep												

Figure 3.2 Overview Of Project Schedule

# 3.5.2 Planning Phase

	O Search tacks	START	WDT	DUET	6			W	5				Ws			Fe	eb 202	22					W	8				W	9
_	- ocal cir table					29	30 31	1 2	3	4 5	6	7 8	9	10 1	1 12	13	14 15	16	17 1	8 19	20	21 3	22 23	3 24	25	26 27	28	1 2	6
	1.0 Planning:	24/Jan	60d	15/Apr			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	<ul> <li>1.1 Analysis Project Title</li> </ul>	24/Jan	1d	24/jan	۲																								
2	<ul> <li>1.2 Study Problem Background</li> </ul>	25/Jan	1d	25/jan	۲																								
3	<ul> <li>1.3 Identify Problem Statements</li> </ul>	26/Jan	1d	26/Jan																									
4	<ul> <li>1.4 Define Project Objectives</li> </ul>	27/jan	1d	27/jan																									
	<ul> <li>1.5 Propose Project Solution</li> </ul>	28/Jan	1d	28/jan																									
6	I.5.1 Research Similar Solutions	28/jan	1d	28/Jan																									
7	<ul> <li>1.5.2 Compare Similar Solutions</li> </ul>	28/Jan	1d	28/Jan																									
8	<ul> <li>1.5.3 Finalise Project Solutions</li> </ul>	28/Jan	1d	28/Jan																									
-	<ul> <li>1.6 Propose Project Approach</li> </ul>	29/Jan	2d	01/Feb					-	-	-	0																	
10	O 1.6.1 Propose Research Approach	29/jan	1d	29/jan																									
11	I.6.2 Propose Development Approach	30/Jan	1d	30/Jan	۲																								
	I.7 Define Project Scope	31/Jan	2d	01/Feb							-																		
13	<ul> <li>1.7.1 Identify Target Users</li> </ul>	31/Jan	1d	31/Jan						0																			
14	<ul> <li>1.7.2 Determine System Scope</li> </ul>	31/Jan	1d	31/Jan																									
15	<ul> <li>1.7.3 Define Modules Covered</li> </ul>	01/Feb	1d	01/Feb																									
	<ul> <li>1.8 Requirement Gathering</li> </ul>	02/Feb	32d	17/Mar	۲							-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-
	I.8.1 Conduct Online Questionnaires	02/Feb	18d	25/Feb								_	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	<ul> <li>1.8.1.1 Design Questions</li> </ul>	02/Feb	3d	04/Feb																									
19	I.8.1.2 Review Questions	05/Feb	1d	05/Feb																									
20	<ul> <li>1.8.1.3 Refine Questions</li> </ul>	06/Feb	18	06/Feb										(															
21	<ul> <li>1.8.1.4 Distribute Questionnaires</li> </ul>	07/Feb	11d	21/Feb			Oper	n work	load a	vailab	oility •	~												-					

Figure 3.3 Planning Phase Schedule



Figure 3.4 Planning Phase Schedule (cont.)

00	Q Search tasks	START T	WDT	DUET	←		W10					Ma	r 202	22				W	12					W13					W	4	
	() 110 Project Scheduling	09/Mar	84	20/84ar		8	9	10	11	12 13	3 14	15	16	17 1	19	20	21 2	22 2	3 24	25	26	27 2	8 29	30	31	1 2	3	4	5 E	7	8 9
	<ul> <li>The Project Scheduling</li> </ul>	05/1101	.00	2.011000	-		<u> </u>									_															
-	<ul> <li>1.10.1 Create Work Breakdown Structure</li> </ul>	09/Mar	3d	11/Mar	(2)		-	_																							
41	<ul> <li>1.10.1.1 Define Main Activities</li> </ul>	09/Mar	1d	09/Mar																											
42	<ul> <li>1.10.1.2 Breakdown Activities into Smaller</li> </ul>	10/Mar	2d	11/Mar																											
	I.10.2 Create Gantt Chart	12/Mar	Sd	20/Mar	(2)				1	-	-	-	-	÷	-	-															
44	<ul> <li>1.10.2.1 Identify Task Dependency</li> </ul>	12/Mar	1d	14/Mar	(2)				1																						
45	<ul> <li>1.10.2.2 Estimate Duration of Project</li> </ul>	15/Mar	1d	15/Mar																											
46	<ul> <li>1.10.2.3 Draft Grantt Chart</li> </ul>	16/Mar	1d	16/Mar								1																			
47	I.10.2.4 Review Gantt Chart	17/Mar	1d	17/Mar																											
48	<ul> <li>1.10.2.5 Finalise Gantt Chart</li> </ul>	18/Mar	1d	20/Mar																											
	+ Add task + Add section																														
+	2.0 Analysis and Design:	21/Mar	15d	08/Apr												E	~							_							
Ð	3.0 Development Phase One:	01/May	6d	09/May																											

Figure 3.5 Planning Phase Schedule (cont.)

# 3.5.3 Analysis and Design Phase

	Q Search tasks	START T	WDT	DUET	←			V	/12					W1	3					W14				1	
-						10	21 2	22 3	23 24	25	26	27	28 2	19 30	31	1	2 3	4	5	6	7	8 9	10	11 1	2
Đ	1.0 Planning:	24/Jan	40d	20/Mar																					
	2.0 Analysis and Design:	21/Mar	15d	08/Apr				-	-			-	-			-					-				
51	<ul> <li>2.1 Design Use Case Diagram</li> </ul>	21/Mar	1d	21/Mar		1																			
52	<ul> <li>2.2 Generate Use Case Descriptions</li> </ul>	22/Mar	Зd	24/Mar																					
53	<ul> <li>2.3 Design Entity Relationship Diagram</li> </ul>	25/Mar	1d	26/Mar																					
54	<ul> <li>2.4 Design Interface Flow Diagrams</li> </ul>	27/Mar	1d	27/Mar																					
55	<ul> <li>2.5 Design Prototypes</li> </ul>	28/Mar	10d	08/Apr								(													

Figure 3.6 Analysis and Design Phase Schedule

# 3.5.4 Development Phase 1

•	Q Search tasks	START T	WD 🔻	DUET	←	e	Today	W18						W15	9	
_						30		3 4	5	6	7	8	9 1	0 11	12	1
Đ	1.0 Planning:	24/Jan	40d	20/Mar												
Đ	2.0 Analysis and Design:	21/Mar	15d	08/Apr												
==	3.0 Development Phase One:	01/May	6d	09/May					-	-	-	-				
	⊘ 3.1 Set up Connection	01/May	4d	05/May		-	-	-	-							
59	3.1.1 Set Up Repository for Web Application	01/May	1d	01/May												
60	3.1.2 Set Up Repository of Mobile Application	02/May	1d	02/May												
61	3.1.3 Configure Server and Database	03/May	1d	03/May												
62	3.1.4 Connect Web Application to Server and Datab	04/May	1d	04/May					)							
63	3.1.5 Connect Mobile Application to Server and Dat	05/May	1d	05/May												
8	3.2 Test Connection	06/May	Zd	09/May						-	_	-	-			
65	3.2.1 Test the Connection of Web Application, Serve	. 06/May	1d	07/May												
66	3.2.2 Test the Connection of Mobile Application, Ser	. 08/May	1d	09/May	(2)							-				

Figure 3.7 Development Phase 1 Schedule

# 3.5.5 Development Phase 2

	O Search tasks	START	WDT	DUET	4	May 2022 0 W21
_				,		9 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
	4.0 Development Phase Two:	24/Apr	49d	30/Jun		
		24/Apr	44d	23/Jun	(1)	
70	4.1.1 Create Web Application Framework	10/May	2d	11/May	2	
	4.1.2 Create Login Feature	12/May	Zđ	14/May		
72	4.1.2.1 Create Login Page User Interface	12/May	1d	12/May	•	
73	4.1.2.2 Implement Algorithm to Allow User Login	13/May	1d	14/May		
	4.1.3 Test Login Feature	15/May	1d	16/May		
75	<ul> <li>4.1.3.1 Test Login Algorithm</li> </ul>	15/May	1d	16/May		
	4.1.4 Create Manage Resident's Registration Feature	17/May	4d	20/May		
77	4.1.4.1 Create Residents's Registration List User	17/May	Zd	18/May		
78	4.1.4.2 Implement Algorithm to Read All Registr	19/May	1d	19/May		
79	4.1.4.3 Implement Algorithm to Retrieve Select	19/May	1d	19/May		
80	④ 4.1.4.4 Implement Algorithm to Update Selecte	20/May	1d	20/May	2	
81	4.1.4.5 Implement Algorithm to Add New Resid	20/May	1d	20/May		
	<ul> <li>4.1.5 Test Manage Resident's Registration Feature</li> </ul>	21/May	1d	21/May		
83	4.1.5.1 Test Algorithm to Read All Registration	21/May	1d	21/May		
84	4.1.5.2 Test Algorithm to Retrieve Selected Regi	21/May	1d	21/May		
85	4.1.5.3 Test Algorithm to Update Selected Regis	21/May	1d	21/May		
86	4.1.5.4 Test Algorithm to Add New Residents af	21/May	1d	21/May		
	<ul> <li>4.1.6 Create Track Resident's Information Feature</li> </ul>	22/May	5d	27/May		
88	4.1.6.1 Create Residents's Information User Int	22/May	1d	23/May	(2)	
89	④ 4.1.6.2 Implement Algorithm to Retrieve All Res	24/May	1d	24/May	۲	Open workload availability 🗸

Figure 3.8 Development Phase 2 Schedule
O. Search tasks	START	WDT	DUET	4	AToda						W2	ż					W23
				_	2	26	27	28 2	9 3	30 3	1 1	2	3	4	5	6	7 8
4.1.6.2 Implement Algorithm to Retrieve All Res	24/May	1d	24/May														
4.1.6.3 Implement Algorithm to Retrieve Select	25/May	1d	25/May														
4.1.6.4 Implement Algorithm to Update Selecte	27/May	1d	27/May														
4.1.6.5 Implement Algorithm to Delete Selected	27/May	1d	27/May	2													
④ 4.1.7 Test Track Resident's Information Feature	28/May	0d	29/May	2			į	-									
4.1.7.1 Test Algorithm to Retrieve All Resident's	28/May	1d	28/May				(										
④ 4.1.7.2 Test Algorithm to Retrieve Selected Resi	28/May	1d	28/May				0										
4.1.7.3 Test Algorithm to Update Selected Resid	29/May	1d	29/May					C									
④ 4.1.7.4 Test Algorithm to Delete Selected Resid	29/May	1d	29/May	2													
<ul> <li>4.1.8 Create Manage Administrators Feature</li> </ul>	30/May	2d	31/May						P	-							
4.1.8.1 Create Manage Administrators User Int	30/May	1d	30/May						C								
4.1.8.2 Implement Algorithm to Retrieve All Ad	31/May	1d	31/May														
4.1.8.3 Implement Algorithm to Add New Admi	31/May	1d	31/May														
4.1.8.4 Implement Algorithm to Remove Admin	31/May	1d	31/May														
<ul> <li>4.1.9 Test Manage Administrators Feature</li> </ul>	01/Jun	1d	01/Jun								-	¢.					
4.1.9.1 Test Algorithm to Retrieve All Administr	01/Jun	1d	01/Jun								1						
<ul> <li>4.1.9.2 Test Algorithm to Add New Administrator</li> </ul>	01/Jun	1d	01/Jun														
<ul> <li>4.1.9.3 Test Algorithm to Remove Administrator</li> </ul>	01/Jun	1d	01/Jun														
<ul> <li>4.1.10 Create Manage Security Guard Feature</li> </ul>	02/Jun	2d	03/Jun									-	-				
4.1.10.1 Create Security Guards List User Interf	02/Jun	1d	02/Jun										)				
4.1.10.2 Implement Algorithm to Read All Secur	03/Jun	1d	03/Jun														
	Q         Search tasks           Q         4.1.6.2 Implement Algorithm to Retrieve All Res           Q         4.1.6.3 Implement Algorithm to Retrieve Select           Q         4.1.6.3 Implement Algorithm to Update Selecte           Q         4.1.6.5 Implement Algorithm to Update Selecte           Q         4.1.6.5 Implement Algorithm to Update Selected           Q         4.1.6.5 Implement Algorithm to Delete Selected           Q         4.1.7 Test Track Resident's Information Feature           Q         4.1.7.1 Test Algorithm to Retrieve All Resident's           Q         4.1.7.2 Test Algorithm to Retrieve Selected Resid           Q         4.1.7.3 Test Algorithm to Delete Selected Resid           Q         4.1.7.4 Test Algorithm to Delete Selected Resid           Q         4.1.8.1 Create Manage Administrators Feature           Q         4.1.8.2 Implement Algorithm to Retrieve All Ad           Q         4.1.8.2 Implement Algorithm to Retrieve All Ad           Q         4.1.8.3 Implement Algorithm to Retrieve All Administr           Q         4.1.8.3 Implement Algorithm to Retrieve All Administr           Q         4.1.9.2 Test Algorithm to Retrieve All Administr           Q         4.1.9.3 Test Algorithm to Retrieve All Administr           Q         4.1.9.3 Test Algorithm t	Q. Search tasks       START v         Q. Search tasks       START v         Q. A.1.6.2 Implement Algorithm to Retrieve All Res       24/May         Q. 4.1.6.3 Implement Algorithm to Retrieve Select       25/May         Q. 4.1.6.4 Implement Algorithm to Dupdate Selected       27/May         Q. 4.1.6.5 Implement Algorithm to Dupdate Selected       27/May         Q. 4.1.6.5 Implement Algorithm to Delete Selected       27/May         Q. 4.1.7 Test Track Resident's Information Feature       28/May         Q. 4.1.7.1 Test Algorithm to Retrieve All Resident's       28/May         Q. 4.1.7.2 Test Algorithm to Retrieve All Resident's       28/May         Q. 4.1.7.3 Test Algorithm to Update Selected Resid       29/May         Q. 4.1.7.4 Test Algorithm to Delete Selected Resid       29/May         Q. 4.1.8.1 Create Manage Administrators Feature       30/May         Q. 4.1.8.2 Implement Algorithm to Retrieve All Ad       31/May         Q. 4.1.8.2 Implement Algorithm to Retrieve All Ad       31/May         Q. 4.1.8.2 Implement Algorithm to Retrieve All Ad       31/May         Q. 4.1.8.1 Create Manage Administrators Feature       01/Jun         Q. 4.1.8.2 Implement Algorithm to Retrieve All Ad	Q         Start         WD Y	Q         Start v         WD v         DUE v           Image: Search tasks         241.6.2 Implement Algorithm to Retrieve All Res         241.400         1d         24.400           Image: All 1.6.3 Implement Algorithm to Retrieve Select         251.400         1d         25.700           Image: All 1.6.3 Implement Algorithm to Delete Select         257.400         1d         25.700           Image: All 1.6.5 Implement Algorithm to Delete Selecte         277.400         1d         27.700           Image: All 1.6.5 Implement Algorithm to Delete Selectel         277.400         1d         27.700           Image: All 1.6.7 Test Track Resident's Information Feature         287.400         287.400         287.400           Image: All 1.7.1 Test Algorithm to Retrieve All Resident's         287.400         287.400         287.400           Image: All 1.7.2 Test Algorithm to Delete Selected Resid         297.400         24         297.400           Image: All 1.7.3 Test Algorithm to Delete Selected Resid         297.400         24         31.7400           Image: All 1.7.4 Test Manage Administrators User Int         307.400         24         31.7400           Image: All 1.8.1 Create Manage Administrators User Int         307.400         24         31.7400           Image: All.8.3 Implement Algorithm to Retrieve All Ad	Q. Search tasks       START       WD v       DUE v       ← <ul> <li>A.1.6.2 Implement Algorithm to Retrieve All Res</li> <li>24/May</li> <li>1d</li> <li>24/May</li> <li>1d</li> <li>24/May</li> <li>1d</li> <li>24/May</li> <li>25/May</li> <li>1d</li> <li>25/May</li> <li>2d</li> <li>4.1.6.3 Implement Algorithm to Retrieve Select</li> <li>27/May</li> <li>1d</li> <li>27/May</li> <li>1d</li> <li>27/May</li> <li>1d</li> <li>27/May</li> <li>2d</li> <li>4.1.6.5 Implement Algorithm to Delete Selected</li> <li>27/May</li> <li>2d</li> <li>4.1.7.1 Test Argorithm to Retrieve All Resident's</li> <li>28/May</li> <li>29/May</li> <li>28/May</li> <li>28/May&lt;</li></ul>	Q. Search tasks       START v       WD v       DUE v       €       Control	Q. Search tasks       START v       WD v       DUE v       ←       O totav       24         ○       4.1.6.2 Implement Algorithm to Retrieve All Res       24/May       1d       24/May       2         ○       4.1.6.3 Implement Algorithm to Retrieve Select       25/May       1d       25/May       2         ○       4.1.6.5 Implement Algorithm to Update Selecte       27/May       1d       27/May       2         ○       4.1.6.5 Implement Algorithm to Update Selected       27/May       1d       27/May       2         ○       4.1.6.5 Implement Algorithm to Retrieve All Resident's       28/May       0d       29/May       2         ○       4.1.7.2 Test Algorithm to Retrieve Selected Resid       29/May       1d       28/May       2         ○       4.1.7.2 Test Algorithm to Delate Selected Resid       29/May       1d       28/May       2         ○       4.1.7.4 Test Algorithm to Delate Selected Resid       29/May       1d       28/May       2         ○       4.1.8.1 Create Manage Administrators User Int       30/May       2       2       2         ○       4.1.8.2 Implement Algorithm to Retrieve All Ad       31/May       2       2       2         ○       4.1.8.1 Create Manag	Q. Search tasks       START v       WD v       DUE v       ←       Q totay       28 27         ○       4.1.6.2 Implement Algorithm to Retrieve All Res       24/May       1d       24/May       2         ○       4.1.6.3 Implement Algorithm to Retrieve Select       25/May       1d       25/May       2         ○       4.1.6.5 Implement Algorithm to Update Selecte       27/May       1d       27/May       2         ○       4.1.6.5 Implement Algorithm to Update Selected       27/May       1d       27/May       2         ○       4.1.6.5 Implement Algorithm to Retrieve Selected Resil       27/May       1d       27/May       2         ○       4.1.7.2 Test Algorithm to Retrieve Selected Resil       28/May       0d       29/May       2         ○       4.1.7.2 Test Algorithm to Delate Selected Resil       28/May       1d       28/May       2         ○       4.1.7.3 Test Algorithm to Delate Selected Resil       29/May       1d       29/May       2         ○       4.1.8.1 Create Manage Administrators User Int       30/May       2d       31/May       2         ○       4.1.8.2 Implement Algorithm to Retrieve All Ad       31/May       1d       31/May       2         ○       4.1.8.	Q. Search tasks       START V       WD V       DUE V       É       Q Total 28 27 28 2	Q. Search tasks       START v       WD v       DUE v       C       Otday 26       27       28       29       28       27       28       29       20       27       28       29       20       20       27       28       29       20       27       28       29       20       20       27       28       29       20         Q. 4.1.6.1 Implement Algorithm to Retrieve All Res       24/May       1d       25/May       1d       25/May       2	Q. Search tasks       START       WD       DUE*       C       Otday       28       27       28       29       30       3         Q. A.1.6.2 Implement Algorithm to Retrieve All Res       24/May       1a       24/May       Q       Q         Q. A.1.6.3 Implement Algorithm to Retrieve Select       25/May       1a       25/May       Q       Q         Q. A.1.6.4 Implement Algorithm to Update Selecte       27/May       1d       27/May       Q       Q         Q. A.1.6.5 Implement Algorithm to Retrieve All Resident's       28/May       Q       29/May       Q       Q         Q. A.1.7.1 Test Algorithm to Retrieve All Resident's       28/May       1d       28/May       Q       Q         Q. A.1.7.2 Test Algorithm to Delete Selected Resid       28/May       1d       28/May       Q       Q         Q. A.1.7.4 Test Algorithm to Delete Selected Resid       28/May       1d       29/May       Q	Start       WD       DUE*       Codds       26       27       28       29       30       31       1         O       4.1.6.2 Implement Algorithm to Retrieve All Res       24/May       1a       24/May       Codds       Codds       27       28       29       30       31       1         O       4.1.6.3 Implement Algorithm to Retrieve Select       25/May       1a       25/May       Codds       Codds	WD       DUE *       *       WD*       DUE *       *       WD*       DUE *       *       WD*       WD	Start       WDv       DUEv       Coddy       26       27       28       29       30       31       1       2       3	Start       WD       DUE       Control       20	Start       WDv       DUEv       C       Otob       28       27       28       28       38       1       2       4       5         O       4.1.6.2 Implement Algorithm to Retrieve All Res       24/May       1       24/May       1       24/May       1       24/May       1       24/May       1       2       3       1       2       3       4       5         O       4.1.6.3 Implement Algorithm to Retrieve Select       25/May       1       1       2       3       4       5       4       5         O       4.1.6.4 Implement Algorithm to Delete Selected       27/May       1       2	<ul> <li>Start</li> <li>WDv</li> <li>DUEv</li> <li>Construction</li> <li>Start</li> <li>WDv</li> <li>Construction</li> <li>Start</li> <li>Start</li> <li>WDv</li> <li>Construction</li> <li>Start</li> <li>Star</li> <li>Start</li> <li>Start</li></ul>

Figure 3.9 Development Phase 2 Schedule (cont.)

	C Wash tasks	START T	WDT	DUET	4	G Toylay		W23					Ju	n 20.	22	
Learr	hing center in worker.				_	2 5	6	78	9	10 1	1 12	13	14 1	15 1	6 17	7 18
109	4.1.10.2 Implement Algorithm to Read All Secur	03/Jun	1d	03/Jun												
110	4.1.10.3 Implement Algorithm to Add New Secu	03/Jun	1d	03/Jun												
111	4.1.10.4 Implement Algorithm to Remove Secur	03/Jun	1d	03/Jun												
	④ 4.1.11 Test Manage Security Guard Feature	04/Jun	1d	04/Jun												
113	4.1.11.1 Test Algorithm to Read All Security Gua	04/Jun	1d	04/Jun												
114	4.1.11.2 Test Algorithm to Add New Security Gu	04/Jun	1d	04/Jun												
115	<ul> <li>4.1.11.3 Test Algorithm to Remove Security Gu</li> </ul>	04/Jun	1d	04/Jun												
	④ 4.1.12 Create Modify User Profile Information Feat	05/Jun	1d	06/Jun			Η.									
117	<ul> <li>4.1.12.1 Create User Profile Interface</li> </ul>	05/Jun	1d	05/Jun												
118	4.1.12.2 Implement Algorithm to Update Admin	06/Jun	1d	06/Jun												
	④ 4.1.13 Test Modify User Profile Information Feature	07/Jun	1d	07/Jun			P	-								
120	4.1.13.1 Test Algorithm to Update Administrato	07/Jun	1d	07/Jun												
	④ 4.1.14 Create Track Visitation's Records Feature	08/Jun	3d	10/Jun				-	-	-						
122	4.1.14.1 Create Visitation's Records User Interfa	08/Jun	2d	09/Jun						1						
123	4.1.14.2 Implement Algorithm to Retrieve All Vi	10/Jun	1d	10/Jun												
124	4.1.14.3 Implement Algorithm to Retrieve Selec	10/Jun	1d	10/Jun												
8	④ 4.1.15 Test Track Visitation's Records Feature	11/Jun	1d	11/Jun	(1)						ч.					
126	4.1.15.1 Test Algorithm to Retrieve All Visitation	11/Jun	1d	11/Jun												
127	4.1.15.2 Test Algorithm to Retrieve Selected Visi	11/Jun	1d	11/Jun												
	④ 4.1.16 Create Manage Announcements Feature	12/Jun	Зd	15/Jun							1	-	-	-		
129	⊘ 4.1.16.1 Create Announcements User Interface	12/Jun	1d	13/Jun												

Figure 3.10 Development Phase 2 Schedule (cont.)

	Q Search tasks	START ¥	WDŦ	DUE	÷	13	Today	22 6 17	18	19	20	21	W25 22 2	23 .	24 2	5
130	4.1.16.2 Implement Algorithm to Add New Ann	14/Jun	18	14/Jun		0										
131	4.1.16.3 Implement Algorithm to Retrieve All An	14/Jun	1d	14/Jun	(2)											
132	4.1.16.4 Implement Algorithm to Update Select	15/Jun	1d	15/Jun												
133	4.1.16.5 Implement Algorithm to Delete Annou	15/Jun	1d	15/Jun												
	④ 4.1.17 Test Manage Announcement Feature	16/Jun	2d	17/Jun				-								
135	④ 4.1.17.1 Test Algorithm to Add New Announce	16/Jun	1d	16/Jun												
136	④ 4.1.17.2 Test Algorithm to Retrieve All Announc	16/Jun	1d	16/Jun			C									
137	④ 4.1.17.3 Test Algorithm to Update Selected Ann	17/Jun	1d	17/Jun												
138	④ 4.1.17.4 Test Algorithm to Delete Announcement	17/Jun	1d	17/Jun												
	4.1.18 Create Manage Feedback Feature	18/Jun	2d	21/Jun	2				-	-	-	-				
140	4.1.18.1 Create Feedback Management User Int	18/Jun	0d	19/Jun												
141	4.1.18.2 Implement Algorithm to Retrieve All Fe	20/Jun	1d	20/Jun												
142	④ 4.1.18.3 Implement Algorithm to Retrieve Selec	20/Jun	1d	20/Jun	(2)											
143	4.1.18.4 Implement Algorithm to Update Select	21/Jun	1d	21/Jun							(					
144	4.1.18.5 Implement Algorithm to Provide Respo	21/Jun	1d	21/Jun							0					
	<ul> <li>4.1.19 Test Receive Feedback Feature</li> </ul>	22/Jun	2d	23/Jun								,	-			
146	4.1.19.1 Test Algorithm to Retrieve All Feedback	22/Jun	1d	22/Jun								0				
147	④ 4.1.19.2 Test Algorithm to Retrieve Selected Fe	22/Jun	1d	22/Jun								0				
148	4.1.19.3 Test Algorithm to Update Selected Fee	23/Jun	1d	23/Jun												
149	④ 4.1.19.4 Test Algorithm to Provide Response to	23/Jun	1d	23/Jun									C			
150	4.1.20 Combine All Developed Features	24/Apr	Зd	27/Apr												

Figure 3.11 Development Phase 2 Schedule (cont.)

•	Q Search tasks	START 🔻	WDT	DUET	<del>(</del>	25 G Today		8.020	ann a'	W26		
148	4.1.19.3 Test Algorithm to Update Selected Fee	23/Jun	1d	23/Jun		2	5 26	27	28	29	30	1
149	4.1.19.4 Test Algorithm to Provide Response to	23/Jun	1d	23/Jun								
150	⊘ 4.1.20 Combine All Developed Features	24/Jun	2d	27/Jun					1			
	4.2 Test Web Application	28/Jun	3d	30/Jun					p	-	ų,	
152	<ul> <li>4.2.1 Test the System's Flow</li> </ul>	28/Jun	3d	30/Jun								

Figure 3.12 Development Phase 2 Schedule (cont.)

# 3.5.6 Development Phase 3

	O Search tasks	START	WDT	DUET	4	Today	1	W23	0				į.	W28	Jul	2022		W2
	C Search lasks	-	110 1	0011		Choday	4	56	7	8 9	9 10	0 11	12	13 14	15	16 17	18	19 21
	5.0 Development Phase Three:	01/Jul	35d	18/Aug			-	-	-	-	-	-	-	-	-	-	-	-
	S.1 Develop Mobile Application	01/jul	32d	15/Aug		-	-	+	-	-	-	÷	-	-	-	-	-	-
156	S.1.1 Create Mobile Application Framework	01/jul	1d	02/Jul														
	S.1.2 Create Submit Registration Form Feature	03/Jul	2d	05/Jul		P	-											
158	S.1.2.1 Create Submit Registration Form User Interface	03/Jul	1d	04/Jul		C												
159	S.1.2.2 Implement Algorithm to Submit Registration Form	05/Jul	1d	05/Jul														
	<ul> <li>5.1.3 Test Submit Registration Form Feature</li> </ul>	06/Jul	1d	06/Jul	۲			per	¢.									
161	<ul> <li>5.1.3.1 Test Algorithm to Submit Registration Form</li> </ul>	06/Jul	1d	06/Jul	(1)													
	5.1.4 Create Login Feature	07/Jul	2d	09/Jul					-	-	1							
163	S.1.4.1 Create Login User Interface	07/Jul	2d	08/Jul	(2)													
164	S.1.4.2 Implement Algorithm to Allow User to Login	09/Jul	1d	09/Jul														
	S.1.5 Test Login Feature	10/jul	1d	10/jul							P							
166	<ul> <li>5.1.5.1 Test Algorithm to Allow User to Login</li> </ul>	10/Jul	1d	10/Jul														
	5.1.6 Create Modify User Profile Feature	11/Jul	2d	12/Jul								E.	-					
168	S.1.6.1 Create Modify User Profile User Interface	11/jul	1d	11/Jul														
169	S.1.6.2 Implement Algorithm to Update User Profile	12/jul	1d	12/Jul	۲													
	5.1.7 Test Modify User Profile Feature	13/Jul	1d	13/Jul	(1)								)					
171	<ul> <li>5.1.7.1 Test Algorithm to Update User Profile</li> </ul>	13/Jul	1d	13/Jul	۲													
	<ul> <li>5.1.8 Create Register Visitor Feature</li> </ul>	14/Jul	2d	16/Jul	(1)									1	-	-		
173	<ul> <li>5.1.8.1 Create Visitor Registration User Interface</li> </ul>	14/Jul	2d	15/Jul														
174	5.1.8.2 Implement Algorithm Register Visitor	16/Jul	1d	16/Jul														

Figure 3.13 Development Phase 3 Schedule

• •	Q. Search tasks	START V	WD 🔻	DUE	<b>~</b>	)2 G Today	W.	29 0 21	22	23	24	25	26 V	V30 27 2	8 2	9 31	31	1	2	V31 3 4
	✓ 5.1.9 Test Register Visitor Feature	17/jul	1d	17/Jul																
176	S.1.9.1 Test Algorithm to Allow Resident to Register Visitor	17/jul	1d	17/Jul	۲															
	5.1.10 Create Manage Registered Visitations Feature	18/jul	3d	20/Jul		-	-	4												
178	S.1.10.1 Create Registered Visitations User Interface	18/jul	2d	19/Jul	(2)															
179	S.1.10.2 Implement Algorithm to Read All Visitations	20/jul	1d	20/Jul																
180	S.1.10.3 Implement Algorithm to Retrieve Specific Visitation	20/Jul	1d	20/Jul																
181	S.1.10.4 Implement Algorithm to Remove Specific Visitation	20/Jul	1d	20/Jul																
	<ul> <li>5.1.11 Test Manage Registered Visitation Feature</li> </ul>	21/Jul	2d	23/Jul				-	-	-										
183	<ul> <li>5.1.11.1 Test Algorithm to Read All Visitations</li> </ul>	21/jul	1d	21/Jul																
184	S.1.11.2 Test Algorithm to Retrieve Specific Visitation	22/jul	1d	22/Jul																
185	S.1.11.3 Test Algorithm to Remove Specific Visitation	23/jul	1d	23/Jul							l.									
8	5.1.12 Create Check in Visitation Feature	24/jul	Zd	26/Jul							-	-	-							
187	S.1.12.1 Crate Check in Visitation User Interface	24/jul	1d	25/Jul	(1)															
188	S.1.12.2 Implement Algorithm Check in Visitation	26/]ul	1d	26/Jul	(2)							1								
	<ul> <li>5.1.13 Test Check in Visitation Feature</li> </ul>	27/jul	1d	27/Jul																
190	S.1.13.1 Test Algorithm Check in Visitation	27/jul	1d	27/Jul									0							
	5.1.14 Create Verify Visitation's Check in Feature	28/Jul	2d	30/Jul	(1)									P	÷	-	1			
192	S.1.14.1 Create Visitor's Check in Verification User Interface	28/Jul	2d	29/Jul										C						
193	S.1.14.2 Implement Algorithm to Verify Visitor's Check in	30/Jul	1d	30/Jul	(2)															
	<ul> <li>5.1.15 Test Verify Visitation's Check in Feature</li> </ul>	31/Jul	1d	31/Jul													-			
195	⊘ 5.1.15.1 Test Algorithm to Verify Visitor's Check in	31/Jul	1d	31/Jul																

Figure 3.14 Development Phase 3 Schedule (cont.)

Q Search tasks	START	WDT	DUET	4	GToday	W31						3	N32					Aug	2022	2	
C Sealuridaka				_	Today	3	4	5	6	7	8	9	10	11	12	13	14	15 1	6 1	7 18	Ű 1
<ul> <li>5.1.15.1 Test Algorithm to Verify Visitor's Check in</li> </ul>	31/jul	1d	31/jul																		
5.1.16 Create View Announcement Feature	01/Aug	4d	04/Aug		-	-	-														
S.1.16.1 Create View Announcement User Interface	01/Aug	Зd	03/Aug																		
5.1.16.2 Implement Algorithm to Read All Announcements	04/Aug	1d	04/Aug																		
S.1.16.3 Implement Algorithm to Retrieve Specific Announce	04/Aug	1d	04/Aug																		
5.1.17 Test Receive Announcement Feature	05/Aug	1d	05/Aug																		
S.1.17.1 Test Algorithm to Read All Announcements	05/Aug	1d	05/Aug																		
5.1.17.2 Test Algorithm to Retrieve Specific Announcement	05/Aug	1d	05/Aug																		
5.1.18 Create Manage Feedback Feature	06/Aug	2d	09/Aug					1	_	-	-	-									
S.1.18.1 Create Manage Feedback User Interface	06/Aug	1d	08/Aug					(													
S.1.18.2 Implement Algorithm to Add Feedback	09/Aug	1d	09/Aug								0										
5.1.18.3 Implement Algorithm to View Specific Feedback	09/Aug	1d	09/Aug								0										
5.1.19 Test Manage Feedback Feature	10/Aug	1d	10/Aug																		
5.1.19.1 Test Algorithm to Add Feedback	10/Aug	1d	10/Aug									(									
S.1.19.2 Test Algorithm to View Specific Feedback		1d																			
⊘ 5.1.20 Contribute All Developed Features	11/Aug	3d	15/Aug	(1)									0								
S.2 Test Mobile Application	15/Aug	4d	18/Aug														j,	-	÷	-	4
5.2.1 Test The System's Flow	15/Aug	4d	18/Aug														1				

Figure 3.15 Development Phase 3 Schedule (cont.)

# 3.5.7 Closing Phase

	O Coards tanks	STAPT	WDT	DUET	4	C Trading			W34					Ň	/35			
	G Search lasks	21ruct 1		5011		21	22	23	24 2	5 26	27	28	29	30 3	81 1	2	3	2
٠	1.0 Planning:	24/Jan	40d	20/Mar														
÷	2.0 Analysis and Design:	21/Mar	15d	08/Apr														
Đ	3.0 Development Phase One:	01/May	6d	09/May														
•	4.0 Development Phase Two:	10/May	38d	30/Jun														
Đ	5.0 Development Phase Three:	01/Jul	35d	18/Aug														
	6.0 Closing:	19/Aug	11d	03/Sep			-	-	-	-	-	-	-	٠	-	-		
215	⊘ 6.1 Conduct User Acceptance Testing	19/Aug	6d	27/Aug	(1)													
216	⊘ 6.2 Create System Documentation	28/Aug	Зd	31/Aug							1							
217	⊘ 6.3 Finalise Documentation of Project	01/Sep	2d	03/Sep	۲													
						L												

Figure 3.16 Closing Phase Schedule

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. What is your role in management teams? <sup>16 responses</sup>



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



Does your management team collect resident information for recording purposes? <sup>16 responses</sup>

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.



What methods do your team currently used to collect information of residents? 16 responses

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.



What methods do your team currently used to manage and organize the resident's information? <sup>16</sup> responses

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.



How satisfied are you with the current process of collecting and managing resident information? <sup>16</sup> responses

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.



How do your team publish or broadcast announcement to residents?

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 reported posting the announcement in the management application.

Do you think there are any barriers to such communication? 16 responses



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.



If yes, which of the following are the barrier(s) of communication?

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.



Are there any inconveniences associated with this way of registering visitor records? <sup>16 responses</sup>

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.



If yes, do your management teams currently implementing the following management application?

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.



What kinds of features would you expect to include in the proposed management system?

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.

f you have of responses	her suggestion, kindly write in this section.
no	
May include f	ee payment
Can have mor	re other features manage facilities, pay management fee and so forth
Allow messag	ge to residents, Pay fees
more features	s to interact with residents will be better, can also add some security feature
keep resident	's information
can provide n	nore features

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.

#### condominium/apartment/guarded area? 30 responses Face to face 8 (26.7%) 17 (56.7%) WhatsApp 4 (13.3%) Telegram -2 (6.7%) WeChat 7 (23.3%) Not communicate at all 5 10 20 0 15

As a resident, how do you communicate with the management team of your

#### 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.

As a resident, how satisfied are you with the communication methods currently using? <sup>30</sup> responses



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.

Do you think there are any barriers to such communication? <sup>30</sup> responses



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.



What method is used to register visitors in your condominium/apartment/guarded area ? 30 responses

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.



As a resident, how satisfied are you with the registration methods currently using? <sup>30 responses</sup>

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.



Are there any inconveniences associated with this way of registering visitor records?  $_{\rm 30\,responses}$ 

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 mobilebased 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 <u>login their</u> account.
- W-2. The web application shall allow the management team to <u>manage</u> resident's registration.
- W-3. The web application shall allow the management team to <u>track</u> resident's information.
- W-4. The web application shall allow the management team to <u>manage</u> <u>administrators</u>.
- W-5. The web application shall allow the management team to <u>manage</u> <u>security guards</u>.
- 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 <u>track the</u> <u>visitor's records</u>.
- W-8. The web application shall allow the management team to <u>manage</u> <u>announcements</u> to all residents.
- W-9. The web application shall allow the management team to <u>manage</u> <u>feedback</u> from residents.

### 4.3.1.2 Mobile Application

- M-1. The mobile application shall allow residents to <u>submit registration</u> form to the management team.
- M-2. The mobile application shall allow residents and security guards to <u>login their accounts</u>.
- 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 <u>manage their</u> registered visitations.
- M-6. The mobile application shall allow visitors to <u>check in their</u> <u>visitations</u>.
- M-7. The mobile application shall allow security guards to <u>verify check in</u> <u>visitations</u>.
- M-8. The mobile application shall allow residents to <u>view announcements</u> 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).

Functional	Use Case ID
Requirement 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

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

### 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
	Lize Cor	a Tamas Datail Daal
Primary Actor: Management learns	Use Cas	e Type: Detail, Real
Stakeholders and Interests:	1	
Management teams - Management tea	ums can lo	ogin their account and access
their application.		
Brief Description: This use case desc	ribes how	management teams log into
their account before accessing the app	lication.	6 6
Trigger: When a management team m	ambar wa	nts to access the application
ingger. when a management team m	ennuer wa	ints to access the application.
Relationships:		
Association : Managem	ent teams	
Extend :-		
Generalization : -		
Normal Flow of Events:		
1 The management team member	r onone th	a web application
2. The management team member	er logs in l	his or her account with email
and password.		
3. The application verifies the en	nail and pa	assword.
Sub flower		
Sub-nows:		
Alternate/Exceptional Flows:		

Use Case Name: Manage resident's registration	ID: 2	Importance Level: High
Primary Actor: Management teams	Use Cas	e Type: Detail, Real
Stakeholders and Interests: Management teams - Management tea registration.	ums can	manage their resident's
Brief Description: This use case describes how manageme registration.	ent teams	manage their resident's
Trigger: When a management team me registration status of the residents' applic his or her account.	mber war ation afte	nts to check and update r successfully logins into
Relationships:		
Association : Management Include : - Extend : - Generalization : -	teams	
Normal Flow of Events:		
1. The management team member cl	icks on th	e "resident" button on the
<ol> <li>navigation bar.</li> <li>The web application shows the default</li> </ol>	new app	lications of residents by
3. The management team member vie	ews the re	sident's information from
<ul> <li>4. The management team member de to verify the residents' information</li> </ul>	ownloads n.	the supporting document
<ol> <li>The management team member resident's registration.</li> </ol>	chooses	to approve or reject the
Sub-flows:		
Alternate/Exceptional Flows:		

Table 4.3 Use Case of Manage Resident's Registration

Use (	Case Name: Tra-	ck resident's	ID: 3	Importance Level: High		
mom	anon					
Primar	y Actor: Manageme	ent teams	Use Cas	Use Case Type: Detail, Real		
				• •		
Stakeholders and Interests: Management teams can track their resident's						
information.						
Brief Description: This use case describes how management teams view or						
remove the residents.						
Trigger: When a management team member wants to view or remove the						
resident from the application after successfully logins into his or her account.						
Relationships:						
	Association	: Management	teams			
	Include	: -				
	Extend	: -				
	Generalization	: -				
Normal Flow of Events:						
1.	The management application.	team member	enters th	ne homepage of the web		
2.	2. The management team member selects the resident's list on the					
3	The management team member clicks on the existing residents tab					
<i>5</i> . 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.						

Table 4.4 Use Case of Track Resident's Information

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:

Use Case Name: administrators	Manage	ID: 4	Importance Level: High			
Primary Actor: Managem	ent teams	Use Case Type: Detail, Real				
Stakeholders and Interests: Management teams - Management teams can manage administrators in the application.						
Brief Description: This use case describes how management teams view, add, remove the administrator from administrator's list.						
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.						
Relationships:						
Association Include Extend Generalization	: Managemer : - : - : -	nt teams				
Normal Flow of Events:						
<ol> <li>The management team member enters the homepage of the web application.</li> <li>The management team member selects the administrator on the navigation bar.</li> <li>The web application shows all the administrator's information</li> </ol>						
<ol> <li>The wee appreadon shows an me administrator's information.</li> <li>The management team member is able to view all administrators' information.</li> </ol>						
5. Only the manager can choose the op 5.1 The S-1 subflo	ment team me eration (add or ow is performe	ember whe remove a d.	o has the highest authority administrator).			

Table 4.5 Use Case of Manage Administrators
Sub-flows:

S-1:

If the add administrator operation is chosen, the S-1.1 flow is performed.
 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:

Use Case Name: Manage guards	security	ID: 5	Importance Level: High
Primary Actor: Management t	eams	Use Cas	se Type: Detail, Real
Stakeholders and Interests: Management teams - Managinformation.	gement te	eams can	manage security guards'
Brief Description: This use ca or view or remove security gu	ise descrit ards.	oes how r	nanagement teams can add
Trigger: When a management team member wants to add or view or remove security guards after successfully logins his or her account.			
Relationships:			
Association : M Include : - Extend : - Generalization : -	anagemen	it teams	
Normal Flow of Events:			
<ol> <li>The management tean application.</li> <li>The management tean navigation bar.</li> <li>The web application sl</li> <li>The management team or remove a security g 5.1The S-1 subflow is</li> </ol>	n membe m memb nows all so n member uard. performed	r enters er select ecurity gu chooses t 1.	the homepage of the web s security guards on the ards' information. o add a new security guard

# Table 4.6 Use Case of Manage Security Guards

Sub-flows:

S-1:

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

Use Case Name: Modify User Profile	ID: 6	Importance Level: High		
Primary Actor: Management teams	Use Cas	e Type: Detail, Real		
Stakeholders and Interests: Management teams - Management team	ns can mo	dify their user profile.		
Brief Description: This use case describes how management teams can modify their own user profile.				
Trigger: When a management team mer after successfully logins his or her acco	nber want ount.	ts to modify their user profile		
Relationships:				
Association: Management teamsInclude: -Extend: -Generalization: -				
Normal Flow of Events:				
<ol> <li>The management team member enters the homepage of the web application.</li> <li>The management team member clicks on the profile photo.</li> <li>The web application shows the administrator's information.</li> <li>The management team member updates his or her information or password.</li> <li>The management team member clicks on the "update" button to update his or her information.</li> </ol>				
Sub-flows:				
Alternate/Exceptional Flows:				

## Table 4.7 Use Case Of Modify User Profile

Use Case Name: Track visitor's records	ID: 7	Importance Level: High			
Primary Actor: Management teams	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 describ search the visitors.	bes how 1	management teams view or			
Trigger: When a management team m visitors from the application after succes	ember wassfully log	ants to view or search the gins his or her account.			
Relationships:					
Association: Management teamsInclude: -Extend: -Generalization: -					
Normal Flow of Events:					
1. The management team member enters the homepage of the web application.					
<ol> <li>The management team member selects visitors on the navigation bar.</li> <li>The web application shows all visitors' lists</li> </ol>					
<ol> <li>The web application shows an visitors fists.</li> <li>The management team clicks on the "total visitors" tab to view all the visitors.</li> </ol>					
5. The web application shows the t	otal visito	ors' list.			
7. The management team member	clicks on 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 4.8 Use Case of Track Visitor's Recors

	Use Case Name: Announcements	Manage	ID: 8	Importance Level: High
	Primary Actor: Manageme	ent teams	Use Cas	se Type: Detail, Real
	Stakeholders and Interests Management teams - Man	: agement teams	can man	age the announcements.
	Brief Description: This use remove the announcement	e case describes	s how ma	nagement teams add, view,
	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 Include Extend Generalization	: Management : - : - : -	teams	
ļ	Normal Flow of Events:			
	<ol> <li>The management application.</li> <li>The management navigation bar.</li> <li>The web application</li> <li>The management to the management to edit draft announce 6.1The S-1 subflow</li> </ol>	team member team member on shows all the eam member v ceam member of ement or remov v is performed.	enters t er selects e annound iews all t chooses to ze annour	he homepage of the web s announcements on the cements. he announcements. o add an announcement or neement operation.

Table 4.9 Use Case of Manage Announcements

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:

Use Ca	se Name: Manage Feedback	ID: 9	Importance Level: High		
Primar	y Actor: Management teams	Use Cas	e Type: Detail, Real		
Stakeh Manag	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.					
Relatio	onships:				
Association: Management teamsInclude: -Extend: -Generalization: -					
Normal Flow of Events:					
1. The management team member enters the homepage of the web					
2.	<ol> <li>The management team member selects feedback on the navigation bar.</li> </ol>				
3.	3. The web application shows all the new feedback from residents by default				
4.	<ol> <li>If the management team member clicks on replied feedback, the S-1 subflow is performed</li> </ol>				
5.	5. The management team member views all the not replied feedback.				
6.	6. The management team member clicks on the "view" button to view the specific feedback.				
7.	7. The web application shows the feedback's title and description.				
8.	8. The management team member clicks on the "reply" button to reply to the feedback.				
<ul><li>9. The management team member enters the reply message.</li><li>10. The management team member clicks on the "send reply" button to send the reply to the resident.</li></ul>					

## Table 4.10 Use Case of Manage Feedback

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**

Use Case Name: Submit Registration Form	ID: 10	Importance Level: High	
Primary Actors: Residents	Use Case	e 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 subm	it registrat	ion form.	
Relationships:			
Association: ResidentsInclude: -Extend: -Generalization: -			
Normal Flow of Events:			
<ol> <li>The resident opens the mobile application.</li> <li>The resident clicks on the" register as a resident" button.</li> <li>The mobile application shows the registration form to the residents.</li> <li>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.</li> <li>The resident clicks on the "submit" button to submit the form and wait for the approval from the management teams.</li> </ol>			
Sub-flows:			
Alternate/Exceptional Flows:			

## Table 4.11 Use Case of Submit Registration Form

Use Case Name: Login Account	ID: 11	Importance Level: High			
Primary Actors: Residents, Security guards	Primary Actors: Residents, Security Use Case Type: Detail, Real guards				
Stakeholders and Interests:	1				
Residents - Residents can login their ac	counts.				
Security guards – Security guards can I	ogin their	accounts.			
Brief Description: This use case describ	bes how re	sidents and security guards			
log into their accounts before entering t	the applica	tion.			
Trigger: When a resident or security gu	uards want	s to logins their account to			
access functions in the application					
Relationships:					
Association : Residents					
Extend : Submit Res	pistration F	Form			
Generalization :-					
Normal Flow of Events:					
1. The resident or security guard o	pens the n	nobile application.			
2. The resident or security guard clicks on the "login" button.					
2.1 The S-1 subflow is performed	ed	• • • • • • • • • • • •			
3. The resident or security guard logs in their accounts with email and					
3.1 The password for the security guard is his or her identification					
number.					
4. The mobile application verifies	the email	and password.			
Sub-flows:					
1. If the user is resident, the resident clicks on login as the "resident" button					
2. If the user is security guard, the secu	rity guard	clicks on "login as security			
guard" button.					

# Table 4.12 Use Case of Login Account

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.

Use Case Name: Modify User Profile	ID: 12	Importance Level: High		
Primary Actors: Residents	Use Case	Type: Detail, Real		
Stakeholders and Interests: Residents - Residents can modify their user profile.				
Brief Description: This use case desc profile.	cribes how	residents modify their user		
Trigger: When a resident wants to mo logins his or her account.	dify their u	ser profile after successfully		
Relationships:				
Association: ResidentsInclude: -Extend: -Generalization: -				
Normal Flow of Events:				
<ol> <li>The mobile application shows the home page.</li> <li>The resident clicks on the profile picture on the home page.</li> <li>The mobile application shows the user information includes name, identification number, email address, phone number, password, car plates.</li> <li>The resident edit the information that he or she wants to modify.</li> <li>The resident clicks on the "update" button.</li> <li>The mobile application updates his or her user profile.</li> </ol>				
Sub-flows				
Alternate/Exceptional Flows:				

## Table 4.13 Use Case of Modify User Profile

## Table 4.14 Use Case of Register Visitors

Use Case Name: Register visitors	ID: 13	Importance Level: High			
Primary Actors: Residents, Security guards	Use Case	Type: Detail, Real			
Stakeholders and Interests: Residents - Residents can pre-register for their visitors. Security guards – Security guards can register for ad-hoc visitors.					
Brief Description: This use case des visitors and how security guards reg	cribes how gister for ad	residents pre-register for their hoc visitors.			
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.					
Relationships:					
Association: Residents, Security guardsInclude: -Extend: -Generalization: -					
Normal Flow of Events:	Normal Flow of Events:				
<ol> <li>The mobile application show</li> <li>The user clicks on register v</li> <li>The user enters visitor's info 3.1 The S-1 subflow is perforincluding name, identification date and time visited and car</li> <li>The user clicks on the "add v</li> <li>The mobile application save</li> </ol>	vs the home isitor butto ormation. ormed on number, c plate. visitor" but s the visitor	e page. n. phone number, email address, ton. r and visitation information.			
Sub-flows:					
<ul><li>S-1:</li><li>1. If the user is residents, the visitor identification number, phone numb and car plate.</li><li>2. If the user is residents, the visitor identification number, phone number</li></ul>	r's informa per, email a r's informa er, email ad	tion to be fill including name, address, date and time visited tion to be fill including name, dress, car plate and unit id.			

Use Case Name: Manage registered visitations	ID: 14	Importance Level: High		
Primary Actors: Residents	Use Cas	e Type: Detail, Real		
Stakeholders and Interests: Residents – Residents can manage their r	Stakeholders and Interests: Residents – Residents can manage their registered visitations.			
Brief Description: This use case describe registered visitation.	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: ResidentsInclude: -Extend: -Generalization: -				
Normal Flow of Events:				
<ol> <li>The mobile application shows the home page.</li> <li>The resident clicks on the "view all visitation" button.</li> <li>The mobile application shows all the visitations including upcoming visitation and the completed visitations.</li> <li>The resident chooses to view specific visitation or cancel specific visitation.</li> <li>The S-1 subflow is performed.</li> </ol>				
Sub-flows:				
<ul><li>S-1:</li><li>1. If view specific visitation is chosen, S-1.1 flow is performed.</li><li>2. If cancel specific visitation is chosen, S-1.2 flow is performed.</li></ul>				

### Table 4.15 Use Case of Manage Registered Visitations

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:

Use Case Name: Check in visitations	ID: 15	Importance Level: High			
Primary Actors: Visitors	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 de visitations.	escribes h	ow visitors check in their			
Trigger: When a visitor arrive apartments/guarded area and wants to	s at tl verify his	ne condominium/serviced or her visitations.			
Relationships:					
Association: VisitorsInclude: Verify VisitationExtend: -Generalization: -					
Normal Flow of Events:					
<ol> <li>The visitor opens the application</li> <li>The visitor clicks on the "use at a state of the visitor enters the visitate residents.</li> <li>The visitor shows the qr code to the security guard verifies the feature of the mobile application promption.</li> </ol>	on. is visitor" l ion id an to the secut qr code. ts the visita	outton. d unit id provided by the rity guard. ation is recorded.			
Sub-flows:					
Alternate/Exceptional Flows:					

### Table 4.16 Use Case of Check in 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 ca	n verify the	e 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 o guards.	or her visit	ation qr code to the security			
Relationships:					
Association: Security guardsInclude: -Extend: -Generalization: -					
Normal Flow of Events:					
<ol> <li>The security guard opens the</li> <li>The security guard logins his</li> <li>The security guard clicks on 14.</li> <li>The security guard scans the</li> <li>The mobile application show the registered visitors.</li> <li>The security guard clicks on in.</li> <li>The mobile application will r security guard id in the visita</li> </ol>	mobile app or her acco the "verify qr code sho s the visitat the "verify record the v tion.	blication. bunt. registered visitors" button. bwn by the visitor. cion id, unit id, and address of y" button to verify the check visitation time and the related			
Sub-flows:					
Alternate/Exceptional Flows:					

# Table 4.17 Use Case of Verify Visitations

Use Case Name: View announcements	ID: 17	Importance Level: High
Primary Actors: Residents	Use Case	e Type: Detail, Real
Stakeholders and Interests: Residents – Residents can receive and the management teams.	view the a	announcements publish by
Brief Description: This use case announcements publish by the managen	describe nent teams	es how residents view
Trigger: When the resident wants to che condominium/service apartment/guarde	ck the late d area afte	st information of his or her r logins his or her account.
Relationships:		
Association       : Residents         Include       : -         Extend       : -         Generalization       : -         Normal Flow of Events:       .         1.       The mobile application shows th         2.       The resident clicks on the "anno         3.       The resident views all the management teams.         4.       The resident clicks on a specific	te homepa uncement <sup>3</sup> announce announce	ge. ' button. ments published by the nents to view the details of
the announcements.		
Sub-flows:		
Alternate/Exceptional Flows:		

### Table 4.18 Use Case of View Announcements

Table 4.	19 Use Case	e of Manag	ze 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	s: 1 manage th	eir feedbac	k to the management teams.									
Brief Description: This feedback to the managem	use case of ent teams, w	describes view the sp	how residents submit their ecific feedback's details.									
Trigger: When a resident management teams and w details after logins his or l	has some op hen the resid her account.	binion and dent wants	wants to share or report to the to view the specific feedback									
Relationships:												
Association Include Extend Generalization	: Residents : - : - : -	5										

#### Table

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

Welcome back!
Login to your account to continue
Email
Password
Forgot your password?
Remember me next time
Login

Figure 4.41 Login Page (Web)

2. Home page

	Today	ents Applica	ation								
	Name	IC no	Email Address	Phone number	Unit ID	Address	Registered Car plate	Support + Document	Approval st	itus	*
esidents	Melvin Kang	195232556	sfds@gmsid.co	0545123	324324	591 triin bukit tinggi, 51203 ki	JQ1868, J588683	Download	Pending	Rejected	Approved
	Chealsea Tan	256463	sdfdg@sdsdd .co	265456231	1412234	591 tmn bukit tinggi. 51203 ki	WQA5263	Download	Pending	Rejected	Approved
sitors	Thessa Lim	1848413324	sfdsfsd⊛dfd. com	15 4513	34234	591 tmn bukit tinggi, 51203 ki	KLD5354	Download	Pending	Rejected	Approved
dministrators											
ocurity guards											
curry guards											
nnouncements											
intouricements											
oodbacks											
eeubacks											

Figure 4.42 Home Page (Web)

3. New Resident Registration Page

Overview	New Reside	ents Applica	ation								
	Name	IC no	Email Address	Phone number	Unit ID	Address	Registered Car plate	Support .	Approval st	atus	
Residents	Melvin Kang	195232556	sfds@gmsid.co sdfdg@sdsód	0545123	324324	591 tmn bukit tinggi, 51203 kl 591 tmn bukit tinggi, 51203	JQ1868, J588683	Download	Pending	Rejected	Approve
/isitors	Thessa Lim	1848413324	.co sfdsfsdi@dfd. com	15 4513	34234	kl 591 tmn bukit tinggi, 51203 ki	KLD5354	Download	Pending	Rejected	Approve
Administrators											
Security guards											
Announcements											
Feedbacks											

Figure 4.43 New Resident Registration Page

4. Existing Resident Page

	Lim Ah Koh User ID: 12345	New Re	gistrat	ion   E	xisting	, Resid	ents					
A	Overview	Existing Res	idents									
	Residents	Name	IC no	Email Address	Phone number	Unit ID	Address	Registered Car plate	Support	Approval	* Remov	.e +
		Melvin Kang	195232556	sfds@gmsid.co	0545123	324324	591 tmn bukit tinggi, 51203 ki	JQ1868, JSB8683	Download	Approved	Remo	we
	Visitors	Chealsea Tan	256463	sdfdg@sdsdd .co	265456231	1412234	591 tmn bukit tinggi, 51203 M	WQA5263	Download	Approved	Remo	we
	VISICOIS	Thessa Lim	1848413324	sfidsfsd@dfd, com	15 4513	34234	591 tmn bukit tinggi, 51203 ki	KLD5354	Download	Approved	Remo	sve
A	Administrators											
A	Security guards											
A	Announcements											
	Feedbacks											
	Log out											

Figure 4.44 Existing Resident Page

## 5. Visitors Page

Lim Ah Koh User ID: 12345	Visitors								
Overview	153	r	4						
Residents	*		C Today						
Visitors	Total visitors		Total visitors (today)						
Administrators	Total Visit	ors List							
Socurity guards	Unit no	Name	ic no	Phone	Email Address	Visitors Car plate	Visitation date	Check in date and time	Verify by
Security guards	24324324	Kee Xiao Fu	45464564564	012325656	asdasd@dsfsd. com	JQ1868, J588683	02-jan-2022 (13:00)	02-Jan-2022 (13:00)	Guarf name
	354353243	Tan Mei Wei	436645674674 656	0256465546	asf@dssf.com	WQA5263	02-jan-2022 (13:00)	02-Jan-2022 (13:00)	Guard name+
Announcements	4535443345	Lai Hong Yi	475686786756	016515325	efs@ssd.co	KLD5354	02-Jan-2022 (13:00)	02-Jan-2022 (13:00)	Guard names
R Feedbacks	Total Visit	ors List (	today)						view more >>
	Unit no	Name	Jc no	Phone number	Email Address	Visitors Car plate	Visitation date and time	Check in date and time	Verify by
	24324324	Kee Xiao Fu	45464564564	012325656	asdasd@dsfsd. com	JQ1868, JS88683	02-jan-2022 (13:00)	02-Jan-2022 (13:00)	Guarf name
	354353243	Tan Mei Wei	436645674674 656	0256465546	ast@dssf.com	WQA5263	02-jan-2022 (13:00)	02-Jan-2022 (13:00)	Guard name=
	4535443345	Lai Hong Yi	475686786756	016515325	efs@ssd.co	KLD5354	02-jan-2022 (13:00)	02-jan-2022 (13:00)	Guard name=
	Text	Text	Text	Text	Text	Text	02-jan-2022 (13:00)	02-Jan-2022 (13:00)	Text
	Text	Text	Text	Text	Text	Text	02-Jan-2022 (13:00)	02-Jan-2022 (13:00)	Text
	Text	Text	Text	Text	Text	Text	Text	Text	Text
	Text	Text	Text	Text	Text	Text	Text	Text	Text
Log out									



# 6. Total Visitors List Page

(	Lim Ah Koh User ID: 12345	Visitor	S								
A	Overview	Total Visit	ors List								Q
•	Residents	Unit no	Name	lc no	Phone number	Email Address	Visitors Car plate	Visitation date and time	Check in date and time	Verify by	
		24324324	Kee Xiao Fu Tan Mei Wei	45464564564	012325656	com asf@dssf.com	JJQ1868, JSB8683	02-Jan-2022 (13:00) 02-Jan-2022 (13:00)	02-Jan-2022 (13:00) 02-Jan-2022 (13:00)	Guard name=	
	Visitors	4535443345	Lai Hong Yi	475686786756	016515325	efs@ssd.co	KLD5354	02-jan-2022 (13:00)	02-jan-2022 (13:00)	Guard name+	
	Administration	Text	Text	Text	Text	Text	Text	02-jan-2022 (13:00)	02-Jan-2022 (13:00)	Text	
	Administrators	Text	Text	Text	Text	Text	Text	02-jan-2022 (13:00)	02-Jan-2022 (13:00)	Text	
	Cogurity guarde	Text	Text	Text	Text	Text	Text	Text	Text	Text	
	Security guards	Text	Text	Text	Text	Text	Text	Text	Text	Text	
	Announcements										
	Feedbacks										
	Log out										

Figure 4.46 Total Visitor List Page

## 7. Administrators Page

Lim Ah Koh User ID: 12345	Admin	istrators &	Security guards				
Overview	Admin	istrators	Security Guards				
Residents	Add	new					
Nisitors	adminis	trators					
Administrators	Admin_id	Name Melvio Kang	ic 3242343289	email xxxxiligmail.com	phone number 0123456789	role president	remove
Security guards	13224	Chealsea Tan	3435345345345	xxxxx@yahoo.com	0198765432	admin	Remove
Announcements	12324	Thessa Lim	65474535345	xxxx@ukdsd.com	01123658974	hr	Remove
<b>Feedbacks</b>							
Log out							

Figure 4.47 Administrators Page

8. Add New Administrator Page

Lim Ah Koh User ID: 12345	
	Add Administrators
Overview	Karna Kelvin Hung
Residents	
Nisitors	Intenditional eac. 701112543262
Administrators	read kelvinhhh@gmail.com
Security guards	Maria tanàna 0156248246
Announcements	Admin
Feedbacks	
	Add
Log out	

Figure 4.48 Add New Administrator Page

### 9. Security Guards Page

9	Lim Ah Koh User ID: 12345		Adminis	strators & Secu	ırity guards			
n Over	view		Adminis	strators   Secu	rity Guards			
Resid	lents		Add new gua	security				
Nisit	ors							
Adn	ninistrators		Securiy_id 132	Name Melvin Kang	ic 3242343289	email xxxxi@gmail.cpm	phone number 0123456789	Remove
Sec	urity guards	1	13224 12324	Chealsea Tan Thessa Lim	3435345345345 65474535345	xxxxxi@yahos.com xxxxdBukdsd.com	0198765432 01123658974	Remove
Anr	nouncement							
Fee	dbacks							
	Log out							

Figure 4.49 Security Guard Page

10. Add New Security Guard Page

	9	Lim Ah Koh User ID: 12345			
	Over	rview		Add Security guards	
				Lorem Ilpusm	
•	Resi	dents		Mentification no.	
•	Visi	tors		880125021542	
A	Adn	ninistrators		lorem@gmail.com	
6	Sec	curity guards	1	Proze source 01851148246	
	Anı	nouncement			
	Fee	edbacks			Add
		Log out			

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





# 16. Replied Feedback Page

	Lim Ah Koh User ID: 12345	Feedback a	and Commen	t						
Ov	erview	New feedb	acks	Repli	ed f	feedback	s			
R	esidents	Feedback	😑 Complaint			Enquiry				
v	lisitors	Title	Deails	Category	÷	User	Email	Address	View	Remove
		Complaint on xxxxxxxxx	consectetur adipiscing elit ut aliquiam pupos sit amet Lorem ipsum dolor sit amet,			Adam sdsd	Adam@gmail.com	5, Jalan Indah 3/3, Taman Bukit Indah, 81200, Johor Bahru, Johor.,	View	Remove
Ad	ministrators	Complaint on xxxxxxxx	consectetur adipiscing elit ut Alipiscim opinis sit annet Lorem ipsum dolor sit annet, consectetur adipiscing elit ut	Feedback		WQA5263	WQA5263	WQA5263	View	Remove
A	nnouncements Feedbacks									
	Log out									

Figure 4.56 Replied Feedback Page

17. Reply Selected Feedback Page



Figure 4.57 Reply Selected Feedback Page
# 18. Modify User Profile Page

Lim Ah Koh User ID: 12345	Edit your profile	
Overview	Uplaad your image xxxxxxx prog uploaded	
Residents	Name Text field	Passence Text field
Nisitors		Cenfirm Password
Administrators	Text field	Text field
Security guards	Email address Text field	]
Announcements	Phone number Text field	]
Feedbacks	Role Text field	]
	Update	
Log out		

Figure 4.58 Modify User Profile Page

# 4.6.2 Mobile Application

1. Login or Register Page

	Login as
	Resident
	or
	Security Guard
	Use as Visitor
	Register as
	Residents
C	Contact Management Teams

Figure 4.59 Login or Register Page

# Residents

1. Register as Resident Page

Figure 4.60 Register as Resident Page

Address Text field	
pload supporting document Text field	
ar plate Text field	

Figure 4.61 Register as Resident Page (cont.)

2. Login as Resident Page

Login as Residents

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

< Back		
Add V	isitors:	
	Visitor's name Text field	
	Visitor's IC no. Text field	
	Mobile phone number Text field	
	Email address Text field	
	Date and Time Visited Text field	
	Car Plate Text field	
	Add Visitor	
🗆 Nav BAR		
° C <b>⊕</b>	- <b>•</b>	♠

Figure 4.65 Register Visitors Page

6. View Visitation Details Page

< Back	
View Vi	sitor:
	Visitor's name Text field
	Visitor's IC no. Text field
	Mobile phone number Text field
	Email address Text field
	Date and Time Visited Text field
	Car Plate Text field
° €	- <b>• •</b>

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

< Back				
For	n Title ext field			
De	scriptions:			
Te	extarea			
				J
Cat F	egories eedback		▽	
	Sul	bmit		
<u>⊖</u> °	-	<b>(</b> )	1	

Figure 4.70 Add Feedback Page

11. View Feedback Detail Page

< Back
21 May 2019, 03:47AM Complaint Replied
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean fames in nulla feugiat ornare aliquam. Tincidunt mauris proin dolor, dui montes, erat fringilla. Dictum nec nisi eget magna sem placerat vulputate integer consequat. A morbi quisque eros, purus. Aliquet posuere ut semper cursus. Pharetra in vivamus auctor tellus cursus. A nulla quam ut id ultricies. Amet, ac nunc gravida a egetas. Phasellus fringilla id metus, volutpat sodales nibh. Fames vivamus elit nibh praesent nisi, sit. Morbi morbi est mi condimentum. Lacus, etiam turpis nibh felis elementum, sollicitudin cum vitae vitae. Tincidunt feugiat volutpat cursus venenatis malesuada euismod. Adipiscing massa varius lorem nibh lorem
<b>Reply from Mohd Faizi:</b> sadsfdfgdfgdfgvcfbfdgsdfhknsdfsjfljslfl sjflsjfjsldfjsljflsjdfdv,nxknvddvdvdfgdf

Figure 4.71 View Feedback Detail Page

12. Modify User Profile Page

< Back	<b>9</b>
Name Text	field
IC no. Text	field
Email add Text	field
Phone nu Text	imber field
Password Text	field
Car plate Text	field
Car plate Text f	field
L	Jpdate
ĉ -	) <b>4</b> ) 🛧

Figure 4.72 Modify User Profile Page

# Visitors

1. Use as Visitors Page

Visitors	
heck-in visitation	
Unit ID	
Text field	
Visitation ID	
Toxt field	
Text field	
* You can get the unit id and	
visitation la from residents.	
Submit	

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

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, is 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.





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 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.

Table 5.1 Table Description

## 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

Welcome to Resident & Visitor Management System	
Email address	
Sign in (For First-time log-in uses your password is your identification number) Remember to chang your password after first-time login. ** Password must be more than 8 number/netters. (Special case like ====================================	

Figure 5.17 Login Page

A	Admin Role: admin	Overv	view					
•	Home	■ 1 New res	idents	5 Total residents		O Today visitors		21 Total visitors
()	Residents							
	Visitors	New F	Residents					
	Administrators	Refresh						
¢D <b></b>	Security Guards	NAME	IC NO	EMAIL ADDRESS	PHONE NUMBER	UNITID	ADDRESS	
•	Announcement	Mucha	790517062727	much@amail.com	60172020270	037	9-2-7 Evergroom	n Cymraes Pandar Syngai Long A
Ð	Feedback	Wusha	780317003727	musn@gmail.com	00173030370	027	o-z-r, Evergree	n Cypress, bandar Sungai Long, 4.
		NAME <		EMAIL ADDRESS	PHONE NUMBER	UNIT ID	ADDRESS	,
		VI:-:-						
calhost:3000	Log out	(Today)	rs					

Figure 5.18 Home Page

A	Admin Role: admin	New Residents Application Existing Residents										
•	Home	New Residents Application										
$\widehat{\hat{a}^{(\theta)}_{\theta}}$	Residents	Refresh										
	Visitors	NAME	IC NO	EMAIL ADDRESS	PHONE NUMBER	UNIT ID	ADDRESS					
e a	Administrators	Musha	780517063727	mush@gmail.com	60173838378	827	8-2-7, Evergreen Cypress, Bandar Sungai Long, 430					
₫D-	Security Guards	NAME	IC NO	EMAIL ADDRESS	PHONE NUMBER	UNITID	ADDRESS					
)	Announcement	.4					,					
	Feedback											
	Log out											



A	Admin Role: admin	New Residen	ts Application	isitng Residents			
•	Home	Exisitng	Residents				
(Å)	Residents				E	nter unit no to search	٩
00	Visitors	Refresh					
20	Administrators	NAME	IC NO	EMAIL ADDRESS	PHONE NU	IMBER UNIT ID	ADDRESS
	Security Guards	Eliza	980424014124	e@gmail.com	60125478	3541 827	8-2-7, Evergreen Cypress, Bandar Sungai Long, 43
(ا	Announcement	Aiko zq	900728014241	aiko@gmail.com	60123458	3798 87	G-8-7, Evergreen Cypress, Bandar Sungai Long, 4:
Đ	Feedback	Felicia	830505020744	f@gmail.com	60106369	9267 987	9-8-7, Evergreen Cypress, Bandar Sungai Long, 43
		Ivan	780612041424	i@gmail.com	6017452	1457 252	2-5-2, Evergreen Cypress, Bandar Sungai Long, 43
		Jacob	880112041424	j@gmail.com	60152441	424 874	8-7-4, Evergreen Cypress, Bandar Sungai Long, 43
	Log out	NAME	IC NO	EMAIL ADDRESS	PHONE NU	IMBER UNIT ID	ADDRESS

Figure 5.20 Exisitng Resident Page

A	Admin Role: admin	Today's Visitors	All Visitors						
	Home	Today's Vi	sitors						
(Ô)ġ	Residents	Refresh							
	Visitors	VISITATION ID	UNIT ID	NAME	IC NO	PHONE NUMBER	CAR PLATE	VISITATION DATE AND TIME	CHECK-IN DATE AND
A	Administrators	No visitor registe	ered today.						
₿d₿	Security Guards	VISITATION ID	UNIT ID	NAME	IC NO	PHONE NUMBER	CAR PLATE	VISITATION DATE AND TIME	CHECK IN DATE AND
)	Announcement								
Ð	Feedback								
	Log out								

Figure 5.21 Today Visitor's Page

A	Admin Role: admin	Today's Visitors All V	/isitors					
	Home	All Visitors						
in the second se	Residents	Refresh						
	Visitors				Enter	unit id to search		Q
20	Administrators	VISITATION ID		NAME	IC NO	PHONE NUMBER	CAR PLATE	VISITATION DATE AND TIME
(D)	Security Guards	430484192260	87	Visitor1	028737	01828	Jhh 123	2022-07-14, 12:48:43
))	Announcement	CY8738444	873	Koh	670312093743	60136273553	KJH2452	2022-08-14, 11:29:05
B	Feedback	CY873244	873	Lim	880216149374	60163738364	KJG3783	2022-08-14, 11:31:30
		438790654487	82772	Elsj	091828	01983	Ksj133	2022-07-19, 12:52:06
		871602308089962	87	Test sg2	8274784	017782747	Khh 1234	null, null
	Log.out	871368539239160	87	7aug	1773777	0173738	Jh 123	2022-08-07, 23:45:25



A	Admin Role: admin	Administ	rators				
•	Home	Add Administrat	ors				
(Å)	Residents	Refresh					
8	Visitors	NAME	IC NO	PHONE NUMBER	EMAIL ADDRESS	ROLE	REMOVE
<b>.</b>	Administrators	yuki	001223019833	0135568795	yuki123@gmail.com	boss	REMOVE
4D	Security Guards	BK1	991024086014	0123546657	backup1@gmail.com	staff	REMOVE
)	Announcement	Admin	991024086014	0123456789	admin@gmail.com	admin	REMOVE
	Feedback	new bk	011230053252	0123457895	newbk@gmail.com	staff	REMOVE
		Sherry	991024086014	0123456789	sherry@gmail.com	account	REMOVE
		Backup2	9910240 <mark>8</mark> 6014	0165788224	backup2@gmail.com	staff	REMOVE
	Log out	dennis	9876867	60114512453	d@gmail.com	chief	REMOVE

Figure 5.23 Adminsitrators Page

A Admin Role: admin	Security G	Security Guards									
Home	Add Security Gua	rds									
Residents	Refresh										
Visitors	NAME	IC NO	PHONE NUMBER	EMAIL ADDRESS	REMOVE						
Administrators	joss	011225018863	60164615465	joss@gmail.com	REMOVE						
Security Guards	Karmugan	780531041235	601123456789	karmugam@gmail.com	REMOVE						
Announcement	balmain	60144856765	6015524324	b@gmail.com	REMOVE						
Feedback	calvin	60157523435	60175896786	c@gmail.com	REMOVE						
	Jason	60183433534	6015551341	jason123@gmail.com	REMOVE						
	kthan	780524058798	6015869548	kthan@hotmail.com	REMOVE						
Log out	NAME	IC NO	PHONE NUMBER	EMAIL ADDRESS	REMOVE						

Figure 5.24 Security Guards Page

A	Admin Role: admin	Announcements
	Home	Add new announcement
());	Residents	Refresh
	Visitors	Your Draft Announcement
-	Administrators	No draft announcement.
¢D-	Security Guards	All Published Announcement
)	Announcement	Published by: Anson Lo
E/	Feedback	Monthly Swimming Pool Closure For Cleaning Dear residents, Kindly note that we has increased the frequency of cleaning of its Swimming Pool. In lieu of this, the pool will be closed on the 2nd Tuesday of the month from 9pm till 9am and a full day every last Wednesday of the month. The pool closures are to facilitate a thorough cleaning process of pool facilities. We apologise for any inconvenience caused. The Management 2022-09-07 1752:51
	Log out	Published by: Anson Lo



A Admin Role: admin	< Back
Home	Announcement
Residents	New Announcement
Visitors	Published by: Admin
Administrators	Title Enter announcement's title
Security Guards	Description
Announcement	Enter description
Feedback	
Log out	

Figure 5.26 Add New Announcement Page

Admin	Announcement
Role: admin	Edit Announcement
Home	Published by: Admin
Residents	Title
Visitors	Test Announcemnt
	Description
Administrators	Hello this is to test ann
Security Guards	
Announcement	Upload new image
Feedback	
Log out	Update -

Figure 5.27 Edit Announcement Page

A	Admin Role: admin	New feedbacks Replied feedba	icks			
•	Home	New feedbacks				
(Å)	Residents	Technical Support     Billing	Support Security Issue	Car Park Issue	Defect of Common Area	• Suggestion
	Visitors	• Others				
<b>.</b>	Administrators	Refresh				
€D	Security Guards	OTHERS	DEFECT OF COMMON AREA	CAR PARK ISSUE		
)	Announcement	Gym room operation hour	Rooftop leaking Found out my rooftop is leaking	Someone using parking lot	my	
	Feedback	What is the gym room operation Felicia f@gmail.com Unit id: 987 Reply	AIko 2q aiko@gmail.com Unit id: 87 Reply	Hi, would like to report AZ aiko@gmail.co Unit id: 87 Reply	that i	
	Log out					



A	Admin Role: admin	New feedbacks	Replied feedbacks						
	Home	Replied feedbacks							
(Å)	Residents			Enter unit id to search			Q		
0	Visitors	Refresh							
20	Administrators	TITLE	DESCRIPTIONS	CATEGORY	REPORTED BY	EMAIL ADDRESS	UN		
¢D <b>3</b>	Security Guards	reyertye	dfgfdhgfh		lvan	i@gmail.com	25		
<b>(</b> )	Announcement	Testing	Hdhgdhx shshhd shvxvs shhzvx shshhs	CAR PARK ISSUE	Aiko zq	aiko@gmail.com	87		
	Feedback	te feedback	dfjklsogdls sjdflsj gsjflsdj lsjf ljsljfls jsjfdlsdj	OTHERS	Aiko zq	aiko@gmail.com	87		
		Jhgfgh	Jshdhdhdh	BILLING SUPPORT	Aiko zq	aiko@gmail.com	87		
		wolalal	dgdfghdfghfghgf	OTHERS	Felicia	f@gmail.com	98		
	Log out	Yhdgd	Shvxvsbs sa fshkdfh skslfjlsjdfs jfksj fksjkfj j	DEFECT OF COMMON AREA	Aiko zq	aiko@gmail.com	87		

Figure 5.29 Replied Feedback Page

A Role: admin	Edit your profile
Home	A & Upload new image
Residents	Name
2	Admin
Visitors	
Administrators	Identification No.
Ģ Administrators	551024000014
Security Guards	Email Address
	admin@gmail.com
)) Announcement	
a	Phone Number
/ Feedback	0123456789
	Role
	admin
	Edit Password
	New Password:
	Enter password Show
	** Password must be more than 8 numbers/letters. (Special case like@# are allowed to use.)

Figure 5.30 Edit Profile Page



Login as
Resident
or
Visitor
or
Security Guard
Contact Management Team



Figure 5.31 Welcome Page

1:52 🔕 🕅 🕥 💀

× 📚 99

← Login as Resident

#### Resident

	,1001000
Plea	se enter your email address.
Eg: aar	on@gmail.com
Passv	vord
Plea	se enter your password.
** Pas: letters use.)**	sword must be more than 8 numbers/ (Special case like@# are allowed to
	l'm a new user. Sign Up
1	Login

Figure 5.32 Login as Resident Page

Register	← Register			
	Please enter your password.			
Register as Residents User Information	** Password must be more than 8 numbers/letters. (Special case like@# are allowed to use.)**			
	Car plate			
Name	Please enter your carplate number.			
Please enter your name.	(Optional) Eg: KLS1234			
Eg: Lim Kwah Ho				
IC no.	House / unit Information			
Please enter your identification number.				
Eg: 780512086632	Unit ID			
Phone number	Select your unit id			
Please enter your phone number.	Address			
Eg: 0123547895	Please enter your address.			
Email Address	Upload supporting document			
Please enter your email address.	Choose Image			
Eg: aaron@gmail.com	Eg: Water bill, or electric bill			
Password				
Please enter your password.	Sign Lin			
** Password must be more than 8 numbers/letters.	Sign Op			
(Special case like@# are allowed to use.)**				
Car plate				



1:53 🔕 🕅 🥥 😶

#### Home





## **Upcoming Visitors**



Figure 5.34 Home Page



Figure 5.35 Upcoming Visitor Page





Figure 5.36 Visitation Hitory Page
← Add New Visitor

#### Visitor's Name

Please enter visitor's name.

#### Eg: Lim Kwah Ho

#### Visitor's IC no.

Please enter visitor's identification number.

#### Eg: 780512086632

#### Phone number

Please enter visitor's phone number.

#### Eg: 60123547895

#### Date visited

Select Date

#### Date selected: 2022-9-6

#### Time visited



#### Time selected: 1:55:20

#### Car Plate

Please enter visitor's phone number.

(optional: Visitors without vehicle does not need to fill this) Eg: KLS1234

#### Unit ID

### 987



Figure 5.37 Add New Visitor Page

← View Visitor

Visitation ID: CY9877510
Visitor's Name Ho
Visitor's IC no. 830416083652
Phone number 60183439562
Date and Time visited 2022-08-15, 12:34:34
Car Plate UJG1732





# Announcement





Figure 5.39 Announcement Page

# Feedback

Home

								+
Techr	nical Suppor	rt	Billin	ig Sup	port		Secur	ity Issue
	Car Park	Issue	I	Defect	t of Co	mmo	on Area	
		Sugge	estion		Othe	rs		
<b>!</b> IC	): FD84304	40					Secu	irity Issue
Night	Patrol						ſ	pending
Lorem i	psum dol	or sit a	imet,				L	
consec	tetur adipi	iscing	elit. P	roin d	qui			
2022-09	-07 23:19:02	2						
View Mo	re							
<b>Gym I</b> Lorem i consec 2022-09- <u>View Mo</u>	0: FD7290: Room O ipsum dole tetur adipi -07 23:04:57 re	22 <b>perat</b> or sit a iscing 7	<b>ing l</b> imet, elit. P	Hou roin d	<b>rs</b> qui		SI	uggestion pending
ום <b>Playg</b> Lorem i	): FD1486 <b>round</b> ipsum dolo	91 or sit a	ımet,			Det	fect of tea	Common replied

Figure 5.40 View Feedback Page

 $\bigcirc$ 

Announcement

Feedback

Visitors

1:58 0 14 0 --

1

← View Feedback

1	ID:	FD109246	9
---	-----	----------	---

# Didn't receive my billing receipt

2022-09-06 01:58:45 Billing Support pending

Description:

I had paid the management fee for this month however, not yet receive the receipt

Your feedback will be replied by management soon.



Figure 5.41 View Pending Feedback Page

1.00	-	-	-	-	
- 22	.0.3	63	300	62	
. 46.	an a	w		~	

- Andrewson -	1000	and the second second
PW1	-	1.1000-010
		C. 10.0 . J
-		

← View Feedback



### Feedack

2022-08-24 14:38:36 Security Issue replied

Description:

Report the security guard of being rude

#### Reply from Admin

Hi Ms Felicia, thank you for your feedback. We will investigate the issue and evaluate the performance the performance of the security guards. Sorry for the inconvenient caused.



Figure 5.42 View Replied Feedback Page

2:35 🌣 🔕 🕅 🕂

← Add New Feedback

Description		
Categories		
Feedback		$\sim$
	Submit	

Figure 5.43 Add New Feedback Page

# ← View Announcement

# Monthly Swimming Pool Closure For Cleaning

Published by: Anson Lo 2022-09-07 17:52:51



Dear residents,

Kindly note that we has increased the frequency of cleaning of its Swimming Pool. In lieu of this, the pool will be closed on the 2nd Tuesday of the month from 9pm till 9am and a full day every last Wednesday of the month. The pool closures are to facilitate a thorough cleaning process of pool facilities.

We apologise for any inconvenience caused.



Figure 5.44 View Announcement Page

← Login as Visitors

# Visitor (Check-in visitations)

Visitation ID

#### Unit ID

\*\* You can get the unit id and visitation id from residents.

	Submit	
	$\bigcirc$	

Figure 5.45 Login as Visitor Page

23:36 🔕 🌣

🗶 🧙 63'

← VisitorQR

(Please show the qr code to security guards)

Cypress Condominium Visitation ID: CY876133 Unit ID: 87



Identification number: 970326072636

> Car Plate JGD3623

Mobile Number: 60163892637

Figure 5.46 Check in Visitation Page

## ← Login as Security Guards

## **Security Guard**

<b>-</b> 10		• 1			- 1		-		
⊢r	ทล	III.	Δ	a	a	r	ρ	SS	
_	110			ч	ч		-	00	

Please enter your email address.

Eg: aaron@gmail.com

#### Password

Please enter your password.

\*\* Password must be more than 8 numbers/ letters. (Special case like .\_-@# are allowed to use.)\*\*

\*\* For first time user, your password would be your ic number. \*\*



Figure 5.47 Login as Security Guard

2:09 🔕 🕅 🕑 ·

**X 斎** 96'

# HomeSG





2:09 🔕 🕅 🔮 🔸

🗙 🧙 96'

← Verify Visitor



(Scan the qr code shown by visitors)



Figure 5.49 Verify Check in Page

# ← Add New Visitor

V	is	ito	or's	N	ar	ne

Please enter visitor's name.

#### Eg: Lim Kwah Ho

#### Visitor's IC no.

Please enter visitor's identification number.

Ea:	780	051	20	86	632
				~ ~	

#### Phone number

Please enter visitor's phone number.

### Eg: 60123547895

#### Car Plate

Please enter visitor's phone number.

(optional: Visitors without vehicle does not need to fill this) Eg: KLS1234

#### Unit ID



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.

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

Table 6.1 Postgrest-js

### 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.

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

Table 6.2 Gotrue-js

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 0.5 Realtime-js			
Function	Description		
on().subscribe()	To subcribe to realtime changes in		
	database.		
removeSubscription()	To remove an active subscription and		

return

connections.

the

number

of

open

Table 6.3 Realtime-js

### 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.

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

Table 6.4 Storage-js

#### 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 defaultdeny 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.

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.

Table 6.5 Deployment process

	STATUS	ENVIRONMENT	DURATION I			
Welcome to Resident & Visitor Management System	Ready	Ready Production 1m 12s (11m ago)				
Louil edition	DOMAINS					
Pageor	rvsm.vercel	app 🖄 rvsm-eelyne	eee.vercel.app 🖄 +2			
Figure is strained and second and second and second and second second and second and second and second and second second and second and second and second and second second and second and second and second and second second and second and second and second and second and second and second second and second and s	BRANCH					
	O main					
	23a05f1 — 0	change 's'				

Figure 6.3 Show Vercel Dashboard

https://rvsm.vercel.app/signin × +		~ - a
→ C ê rvsm.vercel.app/signin		Gues
	Welcome to Resident & Visitor Management System	
	Email address	
	Password	
	Sign in	
	("For First-time log-in cuter, your password is your destitutation number) Remember to change your password after first-time login. <sup>™</sup> Password must be more than Sumphaltersts. (Special case like @# are allowed to use.)	

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

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

Table 7.1 Use Cases

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	Mobile application		
	Visitation			
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.

Functional	Functional Requirements
Requirement	
ID	
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.

Table 7.2 Functional requirements

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 <u>check</u>
	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 <u>view</u>
	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

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	vv - 1	Pass
UTC003	Test retrieve all registration request	Examine whether all the registration request able to access from the client side		Pass
UTC004	Test update selected registration request	Examine whether the update approval button return an updated status message, or a reject status message	W-2	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		Pass
UTC008	Test add administrators with valid input	Examine whether the successful add message will show after add button is pressed	W-4	Pass
UTC009	Test add administrators	Examine whether the error message will show		Pass

Table 7.3 Unit testing test cases (web application)

	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		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	W-5	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/ C	Pass
UTC016	Test modify user password with invalid input	Examine whether the error message will show after update button is pressed	W-0	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		Pass
UTC019	Test add new announcement with valid input	Examine whether the successful add message will show after add button is pressed	W-8	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		Pass
UTC025	Test update selected feedback's reply with valid input	Examine whether the successful update message will show after the update button pressed	W-9	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

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

Table 7.4 Unit testing test cases (mobile application)

UTC029	Test login with correct credential	Examine whether the system will show login success		Pass		
UTC030	Test login with incorrect credential	Examine whether the system will show error message	whether the show error			
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	101-4	Pass		
UTC035	Test read all visitation	Examine whether all the visitation is accessible		Pass		
UTC036	Test remove selected visitation	Examine whether the removed message will show after remove button is pressed	M-5	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	111-0	Pass		
UTC039	Test verify check-in visitation	Examine whether the system show success verify	M-7	Pass		
UTC040	Test retrieve all announcements Examine whether the announcement 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

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

Table 7.5 Integration testing test cases (web application)

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 Announcemen t 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

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

Table 7.6 Integration testing test cases (mobile application)

ITC040	Testverifycheckinvisitation 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.

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

Table 7.7 Tracibility matrices

UTC018, UTC019,	W-8	UC008
UTC020, UTC021,		
UTC022, UTC023		
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,	M-9	UC018
UTC043		

### 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

Test Case	Test Case	Test	Pass	Tested by	Related
ID	Name	Description	1	(user group)	Functional
			Fail		ID
UATC001	Login account	To verify the	Pass	Management	
		account is		Team	W-1
		logged in			
UATC002	Approve	To verify the	Pass	Management	
	resident's	resident's		Team	
	registration	registration is			
		approved			W 2
UATC003	Reject	To verify the	Pass	Management	VV -2
	Resident's	resident's		Team	
	registration	registration is			
		rejected			
UATC004	View all	To verify all	Pass	Management	
	resident's	the resident		Team	W 3
	information	information is			<b>VV-</b> 5
		showed			
UATC005	View all	To verify all	Pass	Management	
	Administrators	the		Team	
		administrators			
		are showed			
UATC006	Add new	To verify new	Pass	Management	
	administrator	administrator		Team	W-4
		is added			
UATC007	Remove	To verify	Pass	Management	
	administrators	administrator		Team	
		is removed			

Table 7.8 UAT test cases (web applicati	on)
---	-----

UATC008	View all	To verify all	Pass	Management	
	security	the security		Team	
	guards	guards are			
		showed			
UATC009	Add new	To verify new	Pass	Management	
	security guard	security guard		Team	W-5
		is added			
UATC010	Remove	To verify	Pass	Management	
	security guard	security guard		Team	
		is removed			
UATC011	Modify new	To verify new	Pass	Management	
	password	password can		Team	W 6
		successfully			<b>W-0</b>
		login			
UATC012	View all	To verify all	Pass	Management	
	visitation	the visitation		Team	
	information	information is			
		showed			W-7
UATC013	Search	To verify the	Pass	Management	
	visitation	related		Team	
	using unit id	visitation is			
		showed			
UATC014	View all	To verify all	Pass	Management	
	announcement	announcement		Team	
		is showed			
UATC015	Add and	To verify	Pass	Management	
	publish new	announcement		Team	
	announcement	is added and			
		published			W-8
UATC016	Add draft	To verify	Pass	Management	
	announcement	announcement		Team	
		is added and			
		save as drafted			
UATC017	Update and	To verify draft	Pass	Management	
1					
	publish draft	announcement		Team	
		is undated and			
---------	--------------	----------------	------	------------	-------
		is updated and			
		published			
UATC018	View	To verify	Pass	Management	
	announcement	announcement		Team	
		detail is			
		showed			
UATC019	Remove	To verify	Pass	Management	
	announcement	announcement		Team	
		is removed			
UATC020	View all	To verify all	Pass	Management	
	feedback	feedback is		Team	
		showed			
UATC021	Reply new	To verify new	Pass	Management	
	feedback	feedback is		Team	WO
		replied			vv -9
UATC022	View replied	To verify	Pass	Management	
	feedback	replied		Team	
		feedback is			
		showed			

# 7.3.2.2 Mobile application

Table 7.9 UAT test cases (	(mobile application)

Test Case	Test Case	Test	Pass	Tested by	Related
ID	Name	Description	1	(user	Functional
			Fail	group)	ID
UATC023	Submit	To verify the	Pass	Resident	
	registration	registration			M-1
	form	form is added			
UATC024	Login account	To verify the	Pass	Resident	
		account is			
		logged in			M-2
UATC025	Login account	To verify the	Pass	security	111 2
		account is		guard	
		logged in			

UATC026	Logout	To verify the	Pass	Resident	
		user is logged			
		out			
UATC027	Logout	To verify the	Pass	security	
	8	user is logged		ouard	
		out		guuru	
	NA 1°C	T ic	D	D 1 (	
UAIC028	Modify user	10 verify	Pass	Resident	
	profile	phone number,			
		carplate, and			
		password is			
		updated and can			M-3
		successfully			
		login after			
		change the			
		password			
UATC029	Add visitor	To verify new	Pass	Resident	
	(residents)	visitors is added			
	Add visitor	To verify new	Pass	Security	M-4
01110050	(security	visitors is added	1 455	guard	171 1
	(security	VISITOIS IS added		guaru	
	guards)	T	D	D 1 /	
UAIC031	View all	10 verify all	Pass	Resident	
	visitation	visitation is			
		showed			
UATC032	View visitation	To verify the	Pass	Resident	
		visitation detail			M-5
		is showed			
UATC033	Remove	To verify the	Pass	Resident	
	upcoming	visitation is			
	visitation	removed			
UATC034	Check in	To verify the	Pass	Visitor	
	visitation	visitation is			
		successfully			M-6
		checked in			
	Verify check in	To verify the	Pass	Security	
UAICUSS	visitation	abaak	1 455	guard	M-7
	visitation	cneck in		guard	

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

#### 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

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

Table 7.10 UAT feedback result summary (web application)

Table 7.11	UAT	feedback	result	summary	(web	application	)
------------	-----	----------	--------	---------	------	-------------	---

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

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

Table 7.12 UAT participants user type

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

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

Table 7.14 UA'	Feedback Result	Summary (	mobile app	olication)

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

Table 7.15 UAT Feedback Result Summary (mobile application)

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

No	Recommendation	Description
1	Live Notification	In order to offer users an immediate
	Update	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	The current system only records the check in
	Check Out Date and	date and time. To increase the accuracy of
	Time	data, check out date and time should be
		recorded by security guards.
3	Scan IC to confirm the	The current mobile application requires
	visitation	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	As the current system only support one reply
	and realtime chat	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.

Table 8.1 Recommendations

5	More	Language	English are the only languages used in the
	Support		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	As a system that used to assist the
	feature		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: <a href="https://www.hindawi.com/journals/wcmc/2019/5743892/">https://www.hindawi.com/journals/wcmc/2019/5743892/</a> [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: <a href="https://www.eaglesflight.com/resource/teamwork-in-the-workplace-frequent-and-effective-communication/">https://www.eaglesflight.com/resource/teamwork-in-the-workplace-frequent-and-effective-communication/</a> [Accessed 11 February 2022].

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

Government of Malaysia, 2019. Shared Prosperity Vision 2030. [online] Government of Malaysia. Available at: <a href="https://www.pmo.gov.my/wp-content/uploads/2019/10/SPV2030-summary-en.pdf">https://www.pmo.gov.my/wp-content/uploads/2019/10/SPV2030-summary-en.pdf</a>> [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: context cont

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: <a href="https://techcrunch.com/2021/09/09/supabase-raises-30m-for-its-open-source-insta-backend/">https://techcrunch.com/2021/09/09/supabase-raises-30m-for-its-open-source-insta-backend/</a>> [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: < <u>https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1215120&dswid=3922</u>> [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: <a href="https://www.freecodecamp.org/news/how-to-build-a-full-stack-application-with-tailwind-css-and-supabase-in-nextjs/">https://www.freecodecamp.org/news/how-to-build-a-full-stack-application-with-tailwind-css-and-supabase-in-nextjs/</a> [Accessed 27 March 2022].

QuestionPro. 2022. Quantitative Research: Definition, Methods, Types andExamples|QuestionPro.[online]Availableat:<https://www.questionpro.com/blog/quantitative-research/>[Accessed18February 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-analternative-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: <a href="https://www.red-gate.com/simple-talk/databases/nosql/how-to-choose-between-sql-and-nosql-databases/">https://www.red-gate.com/simple-talk/databases/nosql/how-to-choose-between-sql-and-nosql-databases/</a> [Accessed 27 March 2022].

Stack Overflow, 2022. *Stack Overflow Developer Survey 2021*. [online] Stack Overflow. Available at: <a href="https://insights.stackoverflow.com/survey/2021">https://insights.stackoverflow.com/survey/2021</a> [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/58928bb ba6fdcc1b4146c5f1/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: < <u>https://www.researchgate.net/publication/242772176\_Fundamentals\_of\_quant</u> <u>itative\_research</u>> [Accessed 19 February 2022]. Wurmser, Y., 2022. Apps Far Outpace Browsers in US Adults' Mobile TimeSpent.[online]InsiderIntelligence.Availableat:<https://www.emarketer.com/content/the-majority-of-americans-mobile-time-</td>spent-takes-place-in-apps>[Accessed 14 February 2022]

## **APPENDICES**

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









	Login Kee1123	
	Login Forget Password?	
	qwertyuiop	
	asdfghjkl	
	☆ z x c v b n m ⊗	
	.7123 space return	
eCommunity	Carrier 🗢 11:12 PM 💻 eCommunity Hill	Review 1: The
	Unit: AppReviewer Amount Payable: RM 0	application is very convenience to use, straight froward and have simple user interface.
	eNotice eBilling eBooking	Review 2: Most user
	eVisitor	friendly apps ever use compare to other apps.
	Home eComNews Guide Settings	Review 3: Provide many useful functions to help in manage unit affairs



# APPENDIX B: Test Cases

# Unit Test (web application)

Test Case	UTC001	Test	Test login with	Pass/Fail	Pass	
ID		Name	e correct			
			credential			
Test Case ]	Description	n Examine whether the system will show lo				
success						
Test Case Scenario				Test Data		
1. User e	nters the valid	and cor	rect email and	Email:		
passw 2 User n	ord to. vresses the log	in hutton		admin@gmai	l.com	
2. Oser presses the login button.			Password: 123456789			
Expected Result			Actual Result			
The system redirects to home page.			The system redirects to			
			home page.			

Test Case	UTC002	Test	Test login with	Pass/Fail	Pass	
ID		Name	incorrect			
			credential			
Test Case I	Test Case Description Examine whether the system will					
message when invalid input is entered.						
Test Case Scenario				Test Data		
1. User enters the invalid email and password.				Email: abc@g		
2. User p	resses the log	in button		Password: 123456		
Expected R	lesult			Actual Resul	lt	
The error message prompts the user entered the			user entered the	The error	message	
invalid input.			prompts	the user		
			entered th	e invalid		
				input.		

Test Case	UTC003	Test	Test retrieve all	Pass/Fail	Pass	
ID		Name	registration			
			request			
Test Case I	Description	Examine	e whether all the re	egistration requ	uest able to	
		access fr	om the client side	•		
Test Case S	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 1	registration		
			results are di	splayed.		

Test Case	UTC004	Test	Test	update	Pass/Fail	Pass	
ID		Name	selecte	d			
			registra	tion			
			request				
Test Case I	est Case Description Examine whether the upda					itton return	
		an upda	ited stat	us mess	age, or a re	ject status	
		message					
Test Case S	Test Case Scenario				Test Data	Test Data	
<ol> <li>User clicks on the approve button on the selected registration's request. OR</li> </ol>			-				
1. User c	licks on the re	ject butto	n on the	selected			
registra	tion's request	•					
Expected Result			Actual Resul	lt			
The status message will be prompt to tell the user			The status message will				
the status is updated.			be prompt	to tell the			
					user the	status is	
					updated.		

Test Case	UTC005	Test	Test retrieve all	Pass/Fail	Pass	
ID		Name	registered			
			resident's			
			information			
Test Case I	Description	Examin	e whether all th	ne registered	resident's	
information is accessible from the client side					side	
Test Case S	Test Case Scenario				Test Data	
1. User e the ext	1. User enters the residents page and clicks on the existing resident tab.			-		
Expected Result				Actual Resul	lt	
All the	existing res	ident's	information are	All the	existing	
displayed.				resident's in	nformation	
				are displayed	1.	

Test Case	UTC006	Test	Test	remove	Pass/Fail	Pass
ID		Name	selected			
			reside	nt		
Test Case I	Description	Examin	e whetl	ner the re	move message	e is shown
when remove button is clicked.						
Test Case Scenario				Test Data		
1. Use	1. User clicks on the remove button on the			-		
selected residents.						
Expected Result				Actual Result		
The status	message will	be promp	t to tell	the user	The status message will	
the residents is removed.			be prompt to tell the			
			user the re	esidents is		
			removed.			

Test	UTC007	Test	Test	retrieve	all	Pass/F	Fail	Pass	
Case ID		Name	admi	nistrators	5				
Test Case	Description	cription Examine whether all the admi							
information is accessible									
Test Case Scenario							Test Data		
1. User enters the administrator's page.						-			
Expected 1	Result					Actual Result			
All the adu	ministrator's i	nformati	on is c	lisplayed		All the administrator's			
The decimination of mornation to displayed.					information is				
							iation	15	
displayed.									

Test	UTC008	Test	Test	add	Pass/Fail	Pass	
Case ID		Name	administra	tors			
			with valid	input			
Test Case	Description	Examin	ne whether	the succ	essful add me	ssage will	
show after add button is pressed							
Test Case	Test Data						
1. User	enters the val	Name: Yann					
number, phone number and email.					IC: 880215016231		
2. 0.501	chers on the	uuu out	ton		Phone	number:	
					60123568745		
					Email:		
					yann@gmail	.com	
Expected Result					Actual Resul	t	
The succe	ss added mes	sage is p	orompted.		The succes	ss added	
					message is pr	rompted.	

Test	UTC009	Test	Test	add	Pass/Fail	Pass
Case ID		Name	administ	rators		
			with	invalid		
			inputs			
Test Case	Description	Examir	ne whether	the error	message will	show after
		ssed				
Test Case	Scenario	Test Data				
1. User	enters the in	nvalid n	ame, iden	tification	Name: Yann	23
card	number, phon	e numbe add butte	er and ema	u1l.	IC: 880215	
2. 0301	eneks on the	add out	011		Phone	number:
					012356874	
					Email: <u>yann</u>	. <u>csof</u>
Expected Result					Actual Resu	lt
The error message is prompted.					The error n	nessage is
		prompted.				

Test	UTC010	Test	Test	remove	Pass/Fail	Pass	
Case ID		Name	administ	ators			
Test Case	Description	Examin	ne whether	the succ	essful remov	e message	
		will sho	ow after th	e remove	button pressed	1	
Test Case	Scenario		Test Data				
1. User selec	clicks on t ted administra	-					
Expected I	Result				Actual Result		
The remov	ed message w	vill be pro	ompt to tel	l the user	The remove	d message	
the admini	strator is rem	oved.			will be pror	npt to tell	
					the use	er the	
					administrato	r is	
					removed.		

Test Case	UTC011	Test	Test	retrie	ve	Pass/Fail	Pass	
ID		Name	all	securi	ity			
			guard	S				
Test Case I	Description	Examine whether all				the securit	y guard's	
information is accessible								
Test Case Scenario						Test Data		
1. User of	enters the secu	irity guar	d's pag	e.		-		
Expected R	lesult					Actual Resul	lt	
All the secu	urity guard's i	nformatic	on is dis	splayed	l.	All the secur	rity guard's	
						information	is	
						displayed.		

Test Case	UTC012	Test	Test	add	Pass/Fail	Pass		
ID		Name	security					
			guards	with				
			valid in	put				
Test Case ]	Description	Examine	whether	the su	ccessful add m	essage will		
		pressed						
Test Case	Test Data							
1. Users	s enters valid	Name: Timoth	ıy					
number, phone number, and email. 2. Users clicks on the add security guard					IC: 85051201	IC: 850512013265		
butto	n.		Phone	number:				
					60123485742			
					Email:			
					timothy@gma	uil.com		
Expected I	Result				Actual Result			
The status	message will	be prompt	to tell th	e user	The status m	essage will		
the security guard is added.					be prompt to t	tell the user		
					the security guard is			
					added.			

Test Case	UTC013	Test	Test	add	Pass/Fail	Pass	
ID		Name	security				
			guards	with			
			invalid in	nputs			
Test Case I	Description	Examine	e whether	the erro	r message will	show after	
	add button is pressed						
Test Case Scenario					Test Data		
1. Users enters invalid name, identification card					Name: Timothy452		
number, phone number, and email.					IC: 8505120	13	
2. 05015	2. Users checks on the add security guard button.				Phone num	ber: 012-	
					348574		
					Email: timot	hy@gma	
Expected R	lesult				Actual Resul	t	
The error n	nessage will b	e prompt.			The error m	essage will	
				be prompt	to tell the		
				user the sec	urity guard		

Test Case	UTC014	Test	Test	remove	Pass/Fail	Pass	
ID		Name	securi	ty			
			guards	5			
Test Case I	Description	Examin	e wheth	ner the suc	ccessful remove message		
will show after the remov					ve button pressed		
Test Case S	Scenario	Test Data					
1. Users select	clicks on th ed security gu	e remove ard.	e butto	n on the	-		
Expected R	lesult				Actual Result		
The remov	ed message v	vill be pr	ompt to	o tell the	The remove	d message	
user the sec	curity guard is	removed	•		will be prompt to tell		
					the user th	e security	
					guard is rem	oved.	

Test Case	UTC015	Test	Test	modify	Pass/Fail	Pass	
ID		Name	user	password			
			with	valid			
			input	t			
Test Case I	Description	Examine	e whet	ther the su	ccessful updat	te message	
will show after update bu					atton is pressed		
Test Case Scenario					Test Data		
1. User enters the new password in valid format(more than 8 numbers or letters)					Password: 0123456789		
2. User	clicks on the u	pdate but	ton.	):			
Expected R	Result				Actual Resul	lt	
The succes	sful update m	nessage w	ill be	prompt to	The success	ful update	
tell the user the password is updated.					message will be prompt		
					to tell the	user the	
passw						updated.	

Test Case	UTC016	Test	Test	modify	Pass/Fail	Pass		
ID		Name	user	password				
			with	invalid				
			input					
Test Case I	Description	Examine	e whet	her the erro	r message will	show after		
		is pressed						
Test Case Scenario					Test Data			
1. User enters the new password in invalid					Password: sc	Password: sdfds		
forma	t. .1:.1							
Z. User C	esult	ipdate but	lon.		Actual Resul	t		
	Coult				Actual Resul	li		
The error n	nessage will b	e prompt.			The error me	essage will		
					be prompt to tell the			
					user the pa	assword is		
					updated.			

Test Case	UTC017	Test	Test		Pass/	Fail	Pass
ID		Name	retrieve	all			
			visitation	L			
			records				
Test Case Description Examine whether all						risitation	n records'
information is accessible							
Test Case S	Scenario				Test Data		
1. User the to	enters the visitation t	itation page a ab.	and clicks	on	-		
Expected R	lesult				Actu	al Resul	t
All the visi	tation informa	ation are disp	layed.		All	the	visitation
					infor	mation	are
					displa	ayed.	

Test	UTC018	Test	Test	retriev	/e	Pass/Fail	Pass	
Case ID		Name	all					
			annou	incemer	nt			
Test Case	Description	Examine	whether all			the announcement's		
information is accessible								
Test Case Scenario						Test Data		
1. User enters the announcement page.						-		
Expected 1	Result					Actual Resu	lt	
All the ann	nouncement in	nformation a	re disp	layed.		All the anno	ouncement	
						information	are	
						displayed.		

Test	UTC019	Test	Test add new	Pass/Fail	Pass		
Case ID		Name	announcement				
			with valid				
			input				
Test Case	Description	Examine v	Examine whether the successful add message will				
		show after	add button is pre	essed			
Test Case	Scenario			Test Data			
1. User	s enters valid	title, descrip	tion, and upload	Title: Upd	ated Lift		
1mag 2. User	e. s clicks on the	e save and p	ublish button.	Maintenance	e Time		
				Description:	The lift		
				maintenance	time is		
				changed to	19 Aug		
				2022. Plo	ease be		
				noticed that	the lift is		
				not fu	nctionable		
				during the p	eriod.		
				Image: lift.p	ng		
Expected Result				Actual Resu	lt		
The succe	ssful add me	ssage will b	e prompt to tell	The succes	ssful add		
the user the announcement is published.			message	will be			
				prompt to te	Il the user		
				the announ	cement is		
				published.			

Test	UTC020	Test	Test add new	Pass/Fail	Pass
Case ID		Name	announcement		
			with invalid		
			inputs		
Test Case	Description	Examine v	whether the succe	essful add me	ssage will
		show after	add button is pre	ssed	
Test Case	Scenario	Test Data			
1. Users	s enters inva	alid title, d	escription, and	Title: ""	
upload image.		e save and publish button		Description: ""	
		suve and puensi eatton.		Image: null	
Expected Result			Actual Result		lt
The arrest manager will be arrest					
I ne error i	nessage will	be prompt.		The error me	essage will
				be prompt.	

Test	UTC021	Test	Test	update	Pass/Fail	Pass
Case ID		Name	selected			
			annou	ncement		
			with	valid		
			input			
Test Case	Description	Examine v	whether	the succ	cessful update	e message
		will show a	after upo	date butto	on is pressed	
Test Case Scenario Test Data						
1. Users enters valid title, descript			tion, and	d upload	Title: Yoga	class
image.		nublish	button	Description:	Feel free	
			puonsi	roution.	to join the y	oga class.
					Contact us a	t: xxx
					Image: yoga	.png
Expected Result					Actual Result	
The error message will be prompt.					The error message wi	
					be prompt.	

Test	UTC022	Test	Test	update	Pass/Fail	Pass
Case ID		Name	selecte	ed		
			annou	ncement		
			with	invalid		
			input			
Test Case Description Examine whether the error				message will	show after	
	update button is pressed					
Test Case	Test Case Scenario				Test Data	
1. Users enters invalid title, descrip			lescripti	on, and	Title: ""	
upload image.		e save and p	ublish button.		Description: ""	
2. Users eners on the save and pe					Image: null	
Expected Result					Actual Result	
The error message will be prompt.					The error message	
					be prompt.	

Test	UTC023	Test	Test	remove	Pass/Fail	Pass	
Case ID		Name	selected				
			annou	incement			
Test Case	Description	Examine v	kamine whether the successful remove message				
		will show a	after th	e remove	button pressed	1	
Test Case	Scenario				Test Data		
1. Users clicks on the remove button on the selected announcement					-		
Expected Result					Actual Resu	lt	
The remov	ved message v	vill be promp	ot to tel	l the user	The removed message		
the annour	the announcement is removed.					will be prompt to tell	
					the use	er the	
					announceme	ent is	
					removed.		

Test Case	UTC024	Test	Test	Pass/Fail	Pass		
ID		Name	retrieve all				
			feedback				
Test Case I	Description	Examine whether all the feedback's information is					
		accessible					
Test Case S	Scenario			Test Data			
1. User	enters the feed	-					
Expected Result Actual Result					lt		
All the feedback information are displayed. The removed mess					d message		
					mpt to tell		
		the use	er the				
				announceme	nt is		
		removed.					

Test Case	UTC025	Test	Test update	Pass/Fail	Pass	
ID		Name	selected			
			feedback's			
			reply with			
			valid input			
Test Case I	Description	Examine w	xamine whether the successful update message			
		will show a	fter the update	e button presse	button pressed	
Test Case S	Scenario			Test Data	Test Data	
1. User enters valid reply message.			Content: Thank you for			
2. User clicks on send reply button.			raising out	raising out the issue.		
			We will tak	e action as		
			soon as poss	ible.		
Expected Result			Actual Result			
The succes	ssful update n	The successful update				
tell the user the message is send.				message will	l be prompt	
				to tell the	user the	
				message is s	end.	
Test Case	UTC026	Test	Test update	Pass/Fail	Pass	
---------------------------	---	----------------	-------------	----------------	-------------	
ID		Name	selected			
			feedback's			
			reply with			
			invalid			
			input			
Test Case I	Test Case Description Examine whether the error			r message will	show after	
the update button pressed						
Test Case S	Scenario			Test Data		
1. User e	enters invalid	reply messag	ge.	Content: ""		
2. User of	clicks on send	l reply buttor	1.			
Expected Result				Actual Resu	lt	
The error n	nessage will b	e prompt.		The error m	essage will	
				be prompt.		

Test	UTC027	Test	Test submit	Pass/Fail	Pass		
Case ID		Name	registration				
			form with				
			valid input				
Test Case	Description	Examine w	whether the succ	essful submit message will			
		show after	the submit but	ton pressed			
Test Case	Scenario	Test Data					
1. User enters name, identification card				Name: Julian	Bob		
numl	ber, phone word car plat	number, er	nail address,	Identification	number:		
and	upload suppo	orting docun	nents in valid	78021405465	2		
form 2 User	at. clicks on the	sion-un hut	ton	Phone	number:		
2. 0.501	cherrs on the	sign up out		60142541485			
				Email	address:		
				julianbob@gr	nail.com		
				Password: 12.	3456789		
				Car plate	number:		
				KSL1245			
				Unit ID: 1452			
				Address: 12-2	2-1, Cypress		
				Condo, Jalan	Sg Long,		
				Kajang, Selan	gor.		
				Supporting	documents:		
				waterbill.png			
Expected	Result			Actual Result			
The succes	ssful submit 1	nessage will	l be prompt to	The success	ful submit		
tell the re	gistration is	sent to the	management	message will	be prompt		
team.				to tell the reg	gistration is		
				sent to the n	nanagement		
				team.			

## Unit Test (mobile application)

Test Case	UTC028	Test	Test submit	Pass/Fail	Pass
ID		Name	registration		
			form with		
			invalid input		
Test Case I	Description	Examine w	hether the erro	r message will	show after
		the submit l	button pressed		
Test Case S	Scenario			Test Data	
1. User of	enters name, i	dentification	card number,	Name: dfd5	5
phone plate	e number, em number, uni	ail address, t id. addres	password, car s and upload	Identificatio	n number:
suppo	orting docume	ents in invalic	l format.	751554	
2. User of	clicks on the s	sign-up butto	n.	Phone nun	nber: 012-
				5454521	
				Email: sdd@	Øsd
				Password: 1	23456
				Car plate nu	mber: 1254
				Unit ID: ""	
				Address: ""	
				Supporting	document:
				····	
Expected R	lesult			Actual Resu	lt
The error n	nessage will b	e prompt.		The error m	essage will
				be prompt.	

Test Case	UTC029	Test	Test	login	Pass/Fail	Pass	
ID		Name	with correct				
			crede	ntial			
Test Case I	Description	Examine w	vhether	the s	ystem will s	show login	
		success					
Test Case Scenario					Test Data		
1. User	enters valid a	and correct e	email a	ddress	Email: <u>f@gmail.com</u>		
and pa 2. User c	assword. clicks on ente	r login buttor	1.		Password: 1	23456789	
Expected Result Actual Result				ılt			
The user wi	ill be redirect	to home pag	home page. The user will			will be	
					redirect to h	ome page.	

Test Case	UTC030	Test	Test	login	Pass/Fail	Pass
ID		Name	with			
			incorr	rect		
			crede	ntial		
Test Case I	Description	Examine w	vhether	the s	ystem will s	show error
message						
Test Case S	Test Case Scenario				Test Data	
1. User	enters inva	lid email	address	s and	Email: ss@g	gmail.
passw 2. User c	ord. clicks on ente	r login buttor	1.		Password: 12	23456
Expected Result Actual Result				lt		
The error message will be prompt.				The error m	essage will	
					be prompt.	

Test Case	UTC031	Test	Test modify	Pass/Fail	Pass	
ID		Name user				
			password			
			with valid			
			input			
Test Case I	Description	Examine w	hether the suc	ccessful upda	te message	
will show after update butt			ton is pressed	l		
Test Case S	cenario			Test Data		
1. User e 2. User e	enters new pa clicks on the e	ssword in val edit password	id format. l button.	Password: fe	elicia1234	
Expected R	esult			Actual Resu	lt	
The successful update message will be prompt.			e prompt.	The success	sful update	
				message	will be	
				prompt.		

Test Case	UTC032	Test	Test modify	Pass/Fail	Pass
ID		Name	user		
			password		
			with invalid		
			input		
Test Case I	Description	Examine wl	nether the error	r message will	show after
update button is pressed					
Test Case S	cenario			Test Data	
1. User e	enters new pa	ssword in inv	alid format.	Password: so	d#@\$@
2. User c	clicks on the e	edit password	button.		
Expected R	esult			Actual Resu	lt
The error message will be prompt.			The error m	essage will	
				be prompt.	

Test Case	UTC033	Test Name	Test add	Pass/Fail	Pass	
ID			visitor			
			with valid			
			input			
Test Case D	Description	Examine wi	nether the suc	cessful add m	nessage will	
		show after a	dd button is p	pressed		
Test Case S	cenario			Test Data		
1. User	enters vis	itor's name	e, visitor's	Visitor's nar	ne: Liew	
identif visit.	fication numb time visit, ca	per, phone n' ar plate num	umber, date ber in valid	Visitor's IC:		
format.				8804300987	880430098754	
2. Users	clicks on the	submit buttor	1.	Phone number:		
				60158756985		
				Date visit:	31-Aug-	
				2022		
				Time visit: 1	2:05:12	
				Car plate	number:	
				JSD2556		
Expected R	esult			Actual Resu	lt	
The success	sful added me	ssage will be	prompt.	The succes	sful added	
				message will	l be prompt.	

Test Case	UTC034	Test Name	Test	add	Pass/Fail	Pass
ID			visitor			
			with			
			invalid			
			input			
Test Case D	Description	Examine wł	hether the	e erro	r message wil	l show after
		add button i	s pressec	1		
Test Case S	cenario				Test Data	
1. User	enters vis	itor's name	e, visit	or's	Visitor's nar	ne: 5343r
identii visit. 1	tication numb	per, phone n r plate numb	umber, o er in inv	date valid	Visitor's IC: 880430-	
forma	t.	· · · · · · · · · · · · · · · · · · ·			09-8754	
2. Users	clicks on the	submit butto	n.		Phone number: 015	
					8756985	
					Date visit:	31-Aug-
					2022	
					Time visit: 1	2:05:12
					Car plate	number:
					3343S	
Expected R	esult				Actual Resu	lt
The error m	essage will b	e prompt.			The error m	essage will
					be prompt.	

Test Case	UTC035	Test	Test	read	Pass/Fa	ail	Pass
ID		Name	all				
			visitat	ion			
Test Case D	Description	Examine wl	hether a	ll the v	isitation	is ac	cessible
Test Case Scenario					Test D	ata	
1. User e	enters the visi	tation page.			-		
Expected R	esult				Actual	Resu	lt
All the visit	ation information	ation are disp	layed.		All	the	visitation
					inform	ation	are
					display	ved.	

Test Case	UTC036	Test	Test	Pass/Fail	Pass
ID		Name	remove		
			selected		
			visitation		
Test Case I	Description	Examine w	hether the rem	oved message	e will show
		after remov	e button is pre	ssed	
Test Case S	Scenario			Test Data	
1. User select	clicks on th ed visitation.	e remove b	utton on the	-	
Expected Result Actual Result				lt	
The remove	ed message w	ill be prompt	•	The removed messag	
				will be prom	ipt.

Test Case	UTC037	Test	Test	check	Pass/Fail	Pass
ID		Name	in vis	itation		
			with	valid		
			input			
Test Case I	Description	Examine w	hether	the syst	em will show	successful
check in						
Test Case Scenario					Test Data	
1. Us	er enters vali	d visitation ic	d and u	nit id.	Visitation	id:
2. Us	er clicks on t	he submit but	tton.		CY8754333	
					Unit id: 875	
Expected Result				Actual Resul	lt	
The successful check in message will be shown			wn	The success	ful check in	
			50 5110	** 11.	1110 50000551	
					message will	l be shown.

Test Case	UTC038	Test	Test check	Pass/Fail	Pass
ID		Name	in visitation		
			with invalid		
			input		
Test Case Description Examine whether the s				ystem will s	show error
message					
Test Case Scenario				Test Data	
1. U	ser enters inv	alid visitatio	on id and unit	Visitation id: 8754333	
id.				Unit id: cy8'	75
2. Us	er clicks on t	he submit but	tton.		
Expected R	Expected Result				lt
The error message will be prompt.			The error message will		
			be prompt.	c	

Test Case	UTC039	Test	Test verify	Pass/Fail	Pass
ID		Name	check-in		
			visitation		
Test Case I	Description	em show succ	ess verify		
Test Case S	cenario	Test Data			
1. User scan the qr code provided by visitors.				-	
Expected R	esult	Actual Resu	lt		
The successful update message is displayed				The error message will	
				be prompt.	

Test	UTC040	Test	Test retrieve all	Pass/Fail	Pass
Case ID		Name	announcements		
Test Case	Description	ouncement is	accessible		
Test Case	Scenario	Test Data			
1. U	ser enters the	nent page.	-		
Expected I	Result			Actual Resu	ılt
All the ann	nouncement i	nformation	is displayed.	All the anno	ouncement
			information	is	
		displayed.			

Test Case	UTC041	Test	Test	add	Pass/Fail	Pass
ID		Name	feedba	ack		
			with	valid		
			input			
Test Case Description Examine whether the suc					essful add m	essage will
		show after a	add butt	ton is pr	ressed	
Test Case S	Scenario				Test Data	
1. User enters title, description and categories				Title: Rooftop leaking		
in v 2 Use	alid format. r clicks on the	add button			Description: I found	
		e uuu outton.			the rooftop l	eaking,
					Category:	defeat of
					common are	a
Expected R	esult				Actual Resu	lt
The success	sful added me	essage will be	e promp	ot.	The success	sful added
			message	will be		
					prompt.	

Test Case	UTC042	Test	Test	add	Pass/Fail	Pass	
ID		Name	feedback				
			with in	valid			
			input				
Test Case I	Test Case Description Examine whether the error				message will	show after	
add button is pressed							
Test Case Scenario				Test Data			
1. U	User enters	title, des	cription	and	Title: ""		
2. Use	ategories in i r clicks on the	nvalid forma e add button.	ıt.		Description:	Description: ""	
					Category: ""	,	
Expected Result				Actual Resu	lt		
The error message will be prompt.				The error m	essage will		
The error message will be prohipt.				be prompt.			

Test Case	UTC043	Test	Test	Pass/Fail	Pass		
ID		Name	retrieved all				
			feedback				
Test Case I	Description	edback is acc	essible				
Test Case S	Scenario			Test Data	Test Data		
1. User enters the feedback page.				-			
Expected R	esult			Actual Resu	lt		
All the feed	lback informa	All the	feedback				
			information	is			
				displayed.			

## Integration Test (web application)

Test Case	ITC001	Test Name	Test	login	Pass/Fail	Pass		
ID			page					
Test Case D	Description	Examine w	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 Resu	lt			
User is able to access the login page.				User is able to access				
					the login pag	ge.		

Test Case	ITC002	Test	Test	home	Pass/Fail	Pass
ID		Name	page if	fuser is		
			authen	ticated		
Test Case Description Examine whether the user				the user	can navigate	e to home
		page if user	is auth	enticated		
Test Case Scenario				Test Data		
1. User clicks on the home page in the side navigation bar.				-		
Expected F	Result				Actual Resu	lt
User will	be redirec	ted to hon	nepage	of the	User will be	redirected
application						ge of the
			application.			

Test Case	ITC003	Test	Test	home	Pass/F	ail	Pass	
ID		Name	page i	f user is				
			not					
			authen	ticated				
Test Case Description Examine whether the user is					redirec	t to lo	gin pag	ge if
		the user is 1	not auth	enticated	•			
Test Case Scenario				Test Data				
1. User navig	clicks on the ation bar.	he home pa	ige in t	the side	-			
Expected F	Result				Actual	Resu	lt	
User will	not be redin	rected to ho	mepage	e of the	User	will	not	be
application.				redired	cted		to	
					homep	bage	of	the
					applic	ation.		

Test Case	ITC004	Test Name	Test	log	Pass/Fail	Pass	
ID			out				
Test Case D	Description	Examine wh	nether th	ne user	is removed f	rom session	
and redirect to login page							
Test Case Scenario					Test Data		
1. User clicks on the log out button.					-		
Expected Result					Actual Resu	lt	
User is redirected to login page.					User is redirected to		
					login page.		

Test Case	ITC005	Test	Test	Pass/Fail	Pass
ID		Name	resident		
			page		
Test Case Description Examine whether the user				<sup>•</sup> can navigate	to resident
		page after tl	ne user is authe	enticated	
Test Case S	cenario	Test Data			
1. User clicks on the resident's button on the side navigation bar.			-		
Expected R	esult			Actual Resu	lt
User is nav:	igated to the	resident page	after the user	User is navig	gated to the
is authenticated.				resident page after the	
		user is authe	nticated.		

Test Case	ITC006	Test	Test	update	Pass/Fail	Pass	
ID		Name	residents'				
			registr	ration's			
			reques	st			
Test Case Description Examine whether the status				of registration	n request is		
updated to database after val					lidated		
Test Case Scenario					Test Data	Test Data	
1. User 2. User select	enters the res clicks on app ed resident's	ident page. prove or reject registration	et butto request	n on the	-		
Expected R	Result				Actual Resu	lt	
The status	of the rest	ident's regis	stration	request	The status	of the	
updated in database. The residents information will				resident's r	registration		
remove from new registration tab and added to				request up	odated in		
existing res	sidents tab.				database.		

Test Case	ITC007	Test	Test	remove	Pass/Fail	Pass		
ID		Name	resident's					
			regist	ration's				
			reque	st				
Test Case I	Description	Examine whether the			resident's registration			
		informatior	ı is ren	noved from	n database			
Test Case S	Scenario				Test Data			
1. User	enters the res	sident page.			-	-		
2. User	clicks on re	ject button	on the	selected				
reside	ent's registrat	tion request.						
Expected F	Result				Actual Resu	Actual Result		
The regist	tration requ	est is remo	oved f	rom the	The 1	registration		
database ar	nd will no lor	iger show in	resider	nt's page.	request is	removed		
				from the da	tabase and			
					will no long	ger show in		
					resident's pa	age.		

Test Case	ITC008	Test	Test	remove	Pass/Fail	Pass
ID		Name	reside	ent's		
			infor	mation		
Test Case Description Examine whether the res			sident's info	rmation is		
removed from database						
Test Case S	Scenario				Test Data	
1. User	enters the res	ident page.			-	
2. User	clicks on ren	nove button	on the	selected		
Exported P	ent s informa	uon			A atual Pasu	1+
Expected N	csuit				Actual Resu	11
The resid	ent's inforn	nation is r	emove	d from	The	resident's
database ar	nd no longer	show in resid	lent's p	page.	information	is
					removed	from
					database and	l no longer
					show in	resident's
					page.	

Test Case	ITC009	Test	Test	Pass/Fail	Pass	
ID		Name	administrator			
			page			
Test Case Description Examine whether the u			iser can na	avigate to		
		administrat	or page after the	user is auther	nticated	
Test Case S	Scenario	Test Data				
1. User clicks on the administrator's button in the side navigation bar.			-			
Expected R	Result			Actual Result		
User is nav	rigated to the	administrato	or page after the	User is na	vigated to	
user is authenticated.				the administ	trator page	
				after the	user is	
				authenticated.		

Test Case	ITC010	Test	Test	add	Pass/Fail	Pass		
ID		Name	administrator					
			page					
Test Case Description Examine whether the user			r can naviga	can navigate to add				
		administrat	or page a	fter the	user is authenticated			
Test Case Scenario					Test Data			
1.	1. User clicks on the administrator's button in the side navigation bar				-			
2.	User is red	lirected to	administ	rator's				
3.	page. User clicks o	on add admir	nistrator b	outton				
Expected F	Result				Actual Resu	lt		
User is nav	igated to the	add adminis	trator pag	e after	User is na	vigated to		
the user is authenticated.					the add add	ministrator		
					page after the user is			
					authenticate	d.		

Test Case	ITC011	Test	Test	add	Pass/Fail	Pass		
ID		Name	administra	itors				
Test Case ]	Test Case Description Examine whether the adm					inistrator's information		
added to database after vali				valic	lated			
Test Case	Scenario				Test Data			
<ol> <li>User enters the administrator's page.</li> <li>User clicks on the add administrator button and redirected to add administrator page.</li> <li>User enters the valid administrator's information.</li> <li>User clicks on the add button</li> </ol>					-			
Expected I	Result				Actual Result			
New adm	ninistrator i	nformation	added to	the	New add	ministrator		
database an	nd shown in	the administr	rator's page	•	information	added to		
				the datab	ase and			
				shown	in the			
					administrator's page.			

Test Case	ITC012	Test	Test	remove	Pass/Fail	Pass		
ID		Name	admir	istrators				
Test Case Description Examine whether the admin					istrator's info	ormation is		
		removed from database						
Test Case S	Scenario				Test Data			
<ol> <li>User enters the administrator's page.</li> <li>User clicks on the remove button on the selected administrator.</li> </ol>								
Expected F	Result				Actual Result			
The admin	istrators will	be removed	d from	database	The adm	ninistrators		
and no long	ger show in t	he administr	ator pa	ge.	will be rem	oved from		
			database and	l no longer				
					show i	n the		
					administrator page.			

Test Case	ITC013	Test	Test	Pass/Fail	Pass	
ID		Name	security			
			guard page			
Test Case Description Examine whether the user			can navigate to security			
		guard page	after the user is	s authenticate	d	
Test Case S	Scenario	Test Data				
1. User the si	1. User clicks on the administrator's button in the side navigation bar.			-		
Expected R	esult			Actual Result		
User is nav	rigated to sec	curity guard p	page after the	User is navig	gated to the	
user is authenticated.			security guard page			
				after the	user is	
				authenticate	d.	

Test Case	ITC014	Test	Test	add	Pass/Fail	Pass	
ID		Name	security	7			
			guard p	age			
Test Case Description Examine whether the use			er can navig	ate to add			
		security gua	ard page a	after t	he user is authenticated		
Test Case Scenario					Test Data		
1.	User click	s on the security guard's			-		
	button in th	ne side naviga	ation bar.				
2.	User is red	irected to see	curity gu	ard's			
	page.						
3.	User click button	s on add se	ecurity g	guard			
Expected R	esult				Actual Resu	lt	
					· · ·		
User is nav	rigated to the	e add securit	y guard	page	User is navig	gated to the	
after the user is authenticated.			add security	guard page			
					after the	user is	
					authenticate	d.	

Test Case	ITC015	Test	Test	add	Pass/Fail	Pass		
ID		Name	security	r				
			guard					
Test Case Description Examine whether the secu					urity guard's i	rity guard's information		
added to database after va				ter val	idated			
Test Case ScenarioTest Data								
<ol> <li>User enters the security guard's page.</li> <li>User clicks on the add security guard button and redirected to add security guard page.</li> <li>User enters the valid security guard 's information.</li> <li>User clicks on the add button.</li> <li>Expected Result</li> </ol>				- Actual Resu	lt			
New secur	rity guard in	nformation a	added to	the	New secur	rity guard		
database ar	d shown in t	ne security gu	uard's pa	ige.	information	added to		
			the database	and shown				
					in the secur	ity guard's		
					page			

Test Case	ITC016	Test	Test remove	Pass/Fail	Pass	
ID		Name	security			
			guards			
Test Case Description Examine whether the secu			rity guard's i	nformation		
		is removed	from database			
Test Case S	cenario			Test Data		
<ol> <li>User enters the security guard's page.</li> <li>User clicks on the remove button on the selected security guards.</li> </ol>			-			
Expected R	esult			Actual Result		
The securit	y guard will	be removed f	from database	The security	guard will	
and no long	ger show in th	ne security gu	ard page.	be remov	red from	
			database and no longer			
				show in th	ne security	
				guard page.		

Test Case	ITC017	Test Name	Test u	user	Pass/Fail	Pass		
ID			profile p	age				
Test Case D	Test Case Description Examine whether the us				er can navigate to user			
profile page after the user is authenticated					ed			
Test Case Scenario					Test Data			
1. User clicks on the avatar in the side navigation bar.				tion	-			
Expected R	esult				Actual Result			
User is navi	igated to user	profile page	after the u	user	User is navig	gated to the		
is authenticated.					user profile page after			
				the user is				
					authenticated.			

Test Case	ITC018	Test	Test	update	Pass/Fa	ail	Pass
ID		Name	passv	vord			
Test Case I	Description	Examine w	hether	the new	password is updated to		
the backend							
Test Case Scenario					Test Data		
1. User clicks on the avatar in the side navigation					-		
bar an	d redirected	to the user pr	ofile p	age.			
2. User $e$	enters new va	ilia passwora					
5. User C	encks on upd	ate profile bu			Astual	Dagu	14
Expected R	lesun				Actual Result		
The passw	ord is updat	ed to the ba	ackend	of the	The	pass	word is
application	. User needs	to use the nev	w pass	word to	updated	d to t	he backend
login their	account on ne	ext login.			of the a	applic	ation. User
				needs	to us	se the new	
			password to login their				
					account on next login.		

Test Case	ITC019	Test	Test	Pass/Fail	Pass
ID		Name	visitation		
			page		
Test Case I	Description	Examine wl	hether the user	can navigate t	o visitation
		page after tl	ne user is authe	nticated	
Test Case Scenario				Test Data	
1. User o naviga	clicks on the ation bar.	visitation pa	ge in the side	-	
Expected R	esult			Actual Resu	lt
User is navigated to visitation page after the user is			fter the user is	User is na	vigated to
authenticated.			visitation pa	ge after the	
				user is authe	nticated.

Test Case	ITC020	Test	Test	Pass/Fail	Pass
ID		Name	Announcement		
			page		
Test Case	Description	Examine	whether the u	ser can na	vigate to
		announcem	nent page after the	user is authe	nticated
Test Case	Scenario			Test Data	
1. User side r	clicks on the avigation ba	e announcer r.	nent page in the	-	
Expected I	Result			Actual Resu	ılt
User is na	vigated to a	nnouncemer	nt page after the	User is na	vigated to
user is authenticated.			announceme	ent page	
				after the	user is
				authenticate	d.

Test Case	ITC021	Test	Test	add	Pass/Fail	Pass
ID		Name	announc	ement		
			page			
Test Case ]	Description	Examine v	whether t	he user	r can naviga	te to add
		announcem	nent page	after the	e user is authe	enticated
Test Case S	Scenario				Test Data	
1. User side annou 2. User	<ol> <li>User clicks on the announcement page in the side navigation bar and redirected to the announcement page.</li> <li>User clicks on the add announcement button</li> </ol>			in the to the utton.	-	
Expected F	Result				Actual Resu	ılt
User is na	vigated to ac	ld announce	ment pag	e after	User is na	vigated to
the user is authenticated.				add ann	ouncement	
			page after	the use is		
					aumenticate	a.

Test Case	ITC022	Test	Test	add	Pass/Fail	Pass
ID		Name	announce	ment		
Test Case ]	Description	Examine w	whether the	anno	uncement's in	nformation
		added to da	itabase afte	er valid	lated	
Test Case	Scenario				Test Data	
1. User	enters the a	dd announce	ment page.		-	
2. User	enters the a	nnouncemen	t informati	ion in		
valic	l format.	_				
3. User	clicks on the	e save and p	ublish butto	on		
Expected I	Result				Actual Resu	lt
The newly	y published	announceme	ent is add	ed to	The newly	published
database a	nd will be s	hown in the	e announce	ement	announceme	ent is
page.			added to database and			
				will be sho	wn in the	
					announceme	ent page.

Test Case	ITC023	Test	Test	update	Pass/Fail	Pass
ID		Name	annoui	ncement		
Test Case ]	Description	Examine w	hether	the anno	uncement's in	nformation
		updated to	database	e after va	lidated	
Test Case S	Scenario				Test Data	
<ol> <li>User enters the announcement page.</li> <li>User clicks on update button on the specific announcement.</li> <li>User enters the updated announcement information in valid format.</li> <li>User clicks on the save and publish button.</li> </ol>					-	
Expected F	Result				Actual Resu	lt
The draft a	innouncemer	nt is updated	to datal	base and	The	draft
will be sho	wn in the an	nouncement	page.		announceme	ent is
					updated to	database
				and will be	shown in	
				the anno	ouncement	
					page.	

Test Case	ITC024	Test	Test	remove	Pass/Fail	Pass
ID		Name	annou	ncement		
Test Case	Description	Examine w	hether	the annour	ncement's info	ormation is
		removed fr	om data	abase		
Test Case S	Scenario				Test Data	
1. User	enters the a	nnouncemen	t page.		-	
2. User	clicks on re	move buttor	n on the	e specific		
anno	ouncement.					
Expected F	Result				Actual Resu	lt
The annou	ncement is r	emoved to d	atabase	and will	The annour	ncement is
no longer b	be shown in t	he announce	ement p	age.	removed to	database
			and will no	longer be		
			shown	in the		
					announceme	ent page.

Test Case	ITC025	Test	Test	Pass/Fail	Pass
ID		Name	feedback		
			page		
Test Case I	Description	Examine w	hether the user	can navigate	to feedback
		page after tl	he user is authe	nticated	
Test Case S	Test Case Scenario			Test Data	
1. User cl naviga	licks on the f tion bar.	eedback butt	on in the side	-	
Expected R	esult			Actual Resu	lt
User is navigated to feedback page after the user is			User is na	vigated to	
authenticated.			feedback pa	ge after the	
				user is authe	enticated.

Test Case	ITC026	Test	Test update	Pass/Fail	Pass
ID		Name	feedback		
			reply		
Test Case I	Description	Examine wl	hether the feed	back's reply is	updated to
		database aft	er validated		
Test Case S	Scenario			Test Data	
1. User en new fe	nters the feed edback tab.	back page an	d clicks on the	-	
feedba	ck.	epiy button c	on the selected		
3. User en	nters the repl	y message in	valid format.		
4. User c	licks on send	reply button			
Expected R	esult			Actual Resu	lt
The reply n	nessage will	be updated to	database and	The reply message will	
the selected	d feedback v	vill no longe	r show in the	be updated	to database
new feedba	ck tab but in	the replied fe	eedback tab.	and the	selected
			feedback wi	ll no longer	
			show in	the new	
				feedback tab	but in the
				replied feed	back tab.

## Integration Test (mobile application)

Test Case	ITC027	Test	Test	Pass/Fail	Pass
ID		Name	registration		
			page		
Test Case I	Description	Examine	whether the	user can na	avigate to
registration page					
Test Case Scenario				Test Data	
1. User	clicks on reg	gister button	in login page.	-	
Expected Result				Actual Resu	lt
User is navigated to register page.			User is na	vigated to	
				register page	2.

Test Case	ITC028	Test	Test	submit	Pass/Fail	Pass
ID		Name	regist	ration		
Test Case I	Description	Examine w	vhether	the sub	mission of a	registration
		request is a	dded to	database	e	
Test Case S	Scenario				Test Data	
1. User	enters the re	gistration pa	ge.		-	
2. User	enters the	registration	inform	ation in		
	l format.	• • • •				
3. User	clicks on su	bmit button				
Expected R	Result				Actual Resu	lt
The registr	ration inform	ation is add	ed to c	latabase	The 1	registration
and the use	r will be redi	rect to welco	ome pag	ge. (The	information	is added to
registration information will be shown in resident				database an	d the user	
page in the web application.)			will be r	edirect to		
					welcome pa	ge.

Test Case	ITC029	Test	Test	login	Pass/Fail	Pass
ID		Name	reside	ent		
			page			
Test Case I	Description	Examine w	hether	the use	er can naviga	te to login
		resident pag	ge			
Test Case Scenario					Test Data	
1. User cl welcom	icks on log e page.	in as reside	ent but	ton in	-	
Expected Result				Actual Resu	lt	
User is navigated to log in as resident page.				User is navig	gated to log	
					in as residen	t page.

Test Case	ITC030	Test	Test login	Pass/Fail Pass
ID		Name	security	
			guard page	
Test Case I	Description	Examine w	hether the user	can navigate to security
		guard's logi	in page	
Test Case S	cenario			Test Data
1. User cli welcom	cks on login e page.	as security g	uard button in	-
Expected R	esult			Actual Result
User is navigated to log in as security guard page.			User is navigated to log	
			in as security guard	
				page.

Test Case	ITC031	Test	Test	home	Pass/Fail	Pass
ID		Name	page if	f user is		
			authen	ticated		
Test Case Description Examine whether the user			can navigate	e to home		
page if user is authenticated						
Test Case Scenario				Test Data		
1. User en	ters the valid	l login crede	ntials.		-	
2. User cl	icks on the lo	ogin button.				
Expected F	Result				Actual Resu	lt
User is nav	vigated to he	ome page af	ter succ	essfully	User is na	vigated to
login.					home pag	ge after
					successfully	login.

Test Case	ITC032	Test Name	Test	log	Pass/Fail	Pass
ID			out			
Test Case D	Description	Examine whether the user is removed from session				
and redirect to login page						
Test Case Scenario					Test Data	
1. User ent 2. User clie	<ol> <li>User enters the profile page.</li> <li>User clicks on the log out button</li> </ol>				-	
Expected R	esult				Actual Resul	lt
User is rem	oved from se	ssion and red	irect to 1	login	User is rem	noved from
page.				session and redirect to		
					login page.	

Test Case	ITC033	Test Name	Test	user	Pass/Fail	Pass
ID			profile	page		
Test Case Description Examine whether the us			the us	er can navig	ate to user	
profile page after the user is authenticated					ed	
Test Case Scenario				Test Data		
1. User cli page.	icks on the p	profile photo	in the	home	-	
Expected R	esult				Actual Result	
User is navigated to the profile page.				User is navigated to the		
				profile page.		

Test Case	ITC034	Test	Test update	Pass/Fail	Pass		
ID		Name	password				
Test Case Description Examine whether the new				w password is	updated to		
the backend							
Test Case S	Scenario	Test Data					
1. User en	ters profile p	-					
2. User enters new valid password.							
3. User cli	cks on updat	e password b	utton.				
Expected R	esult			Actual Resu	Actual Result		
The new password is updated in the backend. The				The new password is			
The new pa	issword is up						
application	will promp	t the user to	o re-login the	updated in th	ne backend.		
application	will promp with new pas	t the user to ssword.	o re-login the	updated in the applic	ne backend. ation will		
application	will promp with new pas	t the user to ssword.	o re-login the	updated in the the prompt the	ne backend. ation will user to re-		
application application	will promp with new pa	t the user to	o re-login the	e updated in th The applic prompt the login the	ne backend. ation will user to re- application		

Test Case	ITC035	Test	Test	add	Pass/Fail	Pass
ID		Name	visitors			
			page			
Test Case I	Description	Examine w	hether t	he us	er can navig	ate to add
visitors page after the user				is authentica	ted	
Test Case Scenario				Test Data		
1. User en	ters the visita	tion page.			-	
2. User cli	cks on the ad	d button.				
Expected R	Expected Result				Actual Result	
User is navigated to add visitor page.				User is navig	gated to add	
			visitor page.			

Test Case	ITC036	Test	Test	add	Pass/Fail	Pass
ID		Name	visitors			
Test Case Description Examine whether the vis				sitation's info	ormation is	
added to database after val				lidated		
Test Case S	cenario				Test Data	
<ol> <li>User enters the add visitor's page.</li> <li>User enters the visitation information in valid format.</li> <li>User clicks on the add button.</li> </ol>				-		
Expected R	esult				Actual Resu	lt
The visitation	on informatio	on is added to	database	and	The	visitation
will be show	wn in the visi	tation page.			information is added to	
					database an	d will be
			shown in th	e visitation		
					page.	

Test Case	ITC037	Test	Test	Pass/Fail	Pass
ID		Name	visitation		
			page		
Test Case Description Examine whether the user			hether the user	can navigate t	o visitation
		page after tl	ne user is authe	nticated	
Test Case Scenario				Test Data	
<ol> <li>User clicks on the visitation button on the bottom tab navigation bar. OR</li> </ol>			-		
2. User cli	cks on the vis	sitation butto	n on the home		
screen.					
Expected R	esult			Actual Resu	lt
User is nav	igated to the	visitation pag	ge.	User is navig	gated to the
				visitation pa	ge.

Test Case	ITC038	Test	Test remove	Pass/Fail	Pass	
ID		Name	visitation			
Test Case Description Examine whether the vis				sitation's information is		
removed from database						
Test Case Scenario   Test Data						
<ol> <li>User enters visitation page.</li> <li>User clicks on the remove button on the selected visitation.</li> </ol>						
Expected R	esult			Actual Result		
The selecte	ed visitation	is removed t	from database	The selected visitation		
and will no	longer show	in visitation	page.	is removed from		
				database an	d will no	
10				longer s	how in	
				visitation pa	ge.	

Test Case	ITC039	Test	Test	check	Pass/Fail	Pass
ID		Name	in vi	sitation		
			page			
Test Case Description Examine whether the user				the user	can navigate	to check in
visitation page						
Test Case Scenario				Test Data		
1. User clipage.	icks on use a	as visitors in	the w	relcome	-	
Expected R	esult				Actual Resu	lt
User is nav	igated to the	check in visi	tation	page.	User is navi	gated to the
			check in	visitation		
					page.	

Test Case	ITC040	Test	Test	verify	Pass/Fail	Pass
ID		Name	check	in		
			visita	tion		
			page			
Test Case I	Description	Examine w	hether	the use	r can navigat	e to verify
	check in visitation page					
Test Case Scenario					Test Data	
1. User lo	1. User login as security guards with valid login			-		
creden	tials and is re-	directed to h	ome pa	ige.		
home p	bage.	illy register		1015 111		
Expected R	lesult				Actual Resu	lt
User is nav	vigated to the	e verify chec	k in vi	sitation	User is navig	gated to the
page.				verify cl	heck in	
					visitation pa	ge.

Test Case	ITC041	Test	Test	Pass/Fail	Pass
ID		Name	announcement		
			page		
Test Case ]	Description	Examine	whether the u	ser can na	vigate to
		announcem	ent page after the	e user is authe	enticated
Test Case S	Scenario			Test Data	
<ol> <li>User clicks on the announcement button on the bottom tab navigation bar. OR</li> </ol>			-		
2. User c home	licks on the screen.	announceme	nt button on the		
Expected F	Result			Actual Resu	lt
User is nav	vigated to the	announcem	ent page.	User is nat	vigated to
			the ann	ouncement	
				page.	

Test Case	ITC042	Test	Test	Pass/Fail	Pass
ID		Name	feedback		
			page		
Test Case Description Examine whether the user				can navigate 1	to feedback
		page after tl	ne user is authe	nticated	
Test Case Scenario				Test Data	
<ol> <li>User clicks on the feedback button on the bottom tab navigation bar. OR</li> </ol>			-		
2. User cli	cks on the fe	edback butto	n on the home		
screen.					
Expected R	esult			Actual Resu	lt
User is nav	igated to the	feedback pag	ge.	User is navig	gated to the
				feedback pag	ge.

Test Case	ITC043	Test	Test	add	Pass/Fail	Pass
ID		Name	feedbac	ck		
			page			
Test Case Description Examine whethe			hether t	the us	er can navig	ate to add
feedback page afte			ige after	the use	er is authentic	ated
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	ITC044	Test	Test	add	Pass/Fail	Pass
ID		Name	feedbac	k		
Test Case I	Description	Examine whether the feedback is added to databas				
		after validated				
Test Case Scenario				Test Data		
<ol> <li>User enters add feedback page.</li> <li>User enters feedback information in valid format.</li> <li>User clicks on add feedback button</li> </ol>					-	
Expected Result					Actual Result	
The feedback information is added into database					The	feedback
and will be shown in feedback page.					information is added	
				into database and will		
				be shown in	n feedback	
				page.		

## User Acceptance Test (web application)

Test	UATC001	Test	Login	Tested by	Management	
Case ID		Name	account		Team	
Test Case	Description	logged in				
Test Case	Instruction			Test Data		
1. En	ter the applica	tion in web	browser	Email:		
2 En	th the url (). ter the valid e	admin@gmail.com				
<ol> <li>2. Enter the value chian and password.</li> <li>3. Clicks on login button.</li> </ol>			Password: 123456789			
Expected ]	Result	Actual Result				
The user i	s authenticate	The user is	s authenticated			
page.			and redire	ected to home		
			page.			
Pass / Fail			Comments			
1. Pa	1. Pass			1.		
2. Pas	SS					
3. Pas	55 55					
$\begin{array}{c c}$	Pass					
6. Pass						

Test	UATC002	Test	Approve	Tested	Management	
Case ID	Name resident's		by	Team		
			registration			
Test Case	Description	To verify t	he resident's re	gistration is approved		
Test Case	Instruction	Test Data	Test Data			
1. Lo 2. Cl na 3. Th res 4. Lo 5. Cl res	icks on reside vigation bar. application sident's registr ooks for the re icks on the ap sident row.	_				
Expected Result				Actual Result		
A successful update status message is showed and			A succe	ssful update		
added to e	xisting reside	nts tab.		status message is showed		
				and added to existing		
			residents tab.			
Pass / Fail			Comments			
1. Pa 2. Pa 3. Pa 4. Pa 5. Pa 6. Pa	SS SS SS SS SS					

Test	UATC003	Test	Reject	Tested	Management	
Case ID		Name	resident's	by	Team	
			registration			
Test Case	Description	To verify t	he resident's reg	gistration is rejected		
Test Case	Instruction			Test Data		
1. Lo 2. Cl na 3. Cl 4. Th reg 5. Lo Lin 6. Cl rov 7. Th co 8. Cl	g in the accounces on reside vigation bar. icks on the near application gistration. The result of th	-				
Expected		· · · ·	-111	Actual Re	suit	
A successful reject message is showed and removed from the resident's registration list.			A succe status mes and adde residents t	ssful update sage is showed d to existing ab.		
Pass / Fail			Comments	5		
1. Pa 2. Pa 3. Pa 4. Pa 5. Pa 6. Pa	SS SS SS SS SS					
Test	UATC004	Test	View	all	Tested	Management
--	--------------	------------	--------------	-----	---------------------------	------------
Case ID		Name	resident's		by	Team
			informatic	n		
Test Case Description To verify all the resident i					nformation	is showed
Test Case	Instruction				Test Data	
<ol> <li>Log in the account.</li> <li>Clicks on residents' button on the side navigation bar.</li> <li>Clicks on the existing residents tab</li> </ol>					-	
Expected	Result	0			Actual Result	
The appli	cation shows	all the ex	isting resid	ent	The application shows	
informatio	on.				all the existing resident	
					information.	
Pass / Fail	1				Comments	5
1.Pass2.Pass3.Pass4.Pass5.Pass6.Pass						

Test	UATC005	Test	View All	Tested	Management
Case ID		Name	Administrators	by	Team
Test Case	Description	tors are showed			
Test Case	e Instruction			Test Data	
<ol> <li>Log in the account.</li> <li>Clicks on administrator's button on the side navigation bar.</li> </ol>				-	
Expected	Result			Actual Result	
The app	lication show	ws all the	e administrator	The application shows	
informati	on.			all the	administrator
				informatio	on.
Pass / Fai	1			Comment	S
1. Pas	5				
2. Pass	5				
3. Pas	8				
4. Pas	8				
5. Pas	8				
6. Pass	5				

Test	UATC008	Test	View All	Tested	Management
Case ID		Name	Security	by	Team
			guards		
Test Case	Test Case DescriptionTo verify all the security guards are showed				
Test Case	e Instruction			Test Data	
1. Log in	the account.			-	
2. Clicks	on security g	uard's butto	on on the side		
navigatio	n bar.				
Expected	Result			Actual Result	
The app	lication show	ws all th	e administrator	The application shows	
informati	on.			all the	administrator
				informatio	on.
Pass / Fai	il			Comment	S
1. Pass	8				
2. Pas	S				
3. Pase	S				
4. Pass	S				
5. Pass	S				
6. Pass	S				

Test	UATC006	Test	Add	new	Tested	Management
Case ID		Name	adminis	trator	by	Team
Test Case	Test Case Description To verify new administrate				or is added	
Test Case	Instruction				Test Data	
<ol> <li>Log in the account.</li> <li>Clicks on administrator's button on the side navigation bar.</li> <li>The application displays all administrator's information.</li> <li>Clicks on the add administrator button.</li> <li>The application redirect to add administrator page and displays a form.</li> <li>Enters name, identification no, email address, phone number and role.</li> <li>Clicks on add button.</li> </ol>			Name: Jas Identificat 88072504 Email jasonliew( Phone nur 60152459 Role: staff	on Liew ion card no: 2154 address: <u>@gmail.com</u> nber: 854		
Expected	Result				Actual Result	
The appl	ication prom	pt the succ	essfully	added	The application prompt	
message.					the successfully added	
					message.	
Pass / Fai	1				Comments	8
1. Pa 2. Pa 3. Pa 4. Pa 5. Pa 6. Pa	155 155 155 155 155 155					

Test	UATC007	Test	Remove	Tested	Management
Case ID		Name	administrators	by	Team
Test Case	Description	To verify	administrator is 1	removed	
Test Case Instruction				Test Data	
1. Lo 2. C 3. Tl ac	og in the acco licks on admin de navigation he application Iministrator's	-			
4. Lo	ooks for the a	dministrato	r named "Phil		
5. Cl	licks on the re Iministrator re	move butto	on on the		
6. Tl	he application	prompt the	e user to ors.		
7. C	licks on confi	rm delete b	utton.	Actual Re	sult
The appl	intion prom	t the succ	antilly delated	The evelication property	
message	ication prom	of the succ	essiully deleted	the successfully deleted	
message.				massage	ssiuny deleted
	1			message.	
Pass / Fai	1			Comment	S
1. Pa 2. Pa 3. Pa 4. Pa 5. Pa 6. Pa	ass ass ass ass ass				

Test	UATC009	Test	Add 1	new	Tested by	Management	
Case ID		Name	security	r		Team	
			guard				
Test Case Description To verify new security guard is added						d	
Test Case	Test Case Instruction					Test Data	
1. Log in the account.					Name: Kei	th Sia Yuan	
2. Cli	cks on securit e navigation b	y guard's bu ar.	utton on t	he	Identificati	on card no:	
3. Th	e application of	displays all s	security		880423054	-524	
gua 4. Cli	ard's informat	ion. d security gu	ard butto	on.	Email	address:	
5. Th	e application i	redirect to a	dd securit	ty	keithyuan@gmail.com		
gua 6 En	ard page and d	lisplays a for ntification n	rm. o email		Phone number:		
ado	dress, and pho	ne number.	io, emun		60182548658		
7. Cli	cks on add bu	tton.					
Expected I	Result				Actual Result		
The applie	cation prompt	t the succes	sfully ad	lded	The application prompt		
message.					the succes	ssfully added	
					message.		
Pass / Fail					Comments		
1. Pas	SS						
2. Pas	SS						
3. Pas	SS						
4. Pas	SS						
5. Pas	SS						
6. Pas	SS						

Test	UATC010	Test	Remove	Tested	Management
Case ID		Name	security	by	Team
			guard		
Test Case Description To verify security guard is removed					
Test Case	Instruction	Test Data			
<ol> <li>Log in the account.</li> <li>Clicks on security guard's button on the side navigation bar.</li> <li>The application displays all security guard's information.</li> <li>Looks for the security guard named "Kok Kim Jun".</li> <li>Clicks on the remove button on the security guard row.</li> <li>The application prompt the user to confirm delete security guard.</li> <li>Clicks on the confirm delete button.</li> </ol>					
Expected I	Kesult			Actual Result	
The applic	cation prompt	the success	fully deleted	The applie	cation prompt
message.				the succes	sfully deleted
				message.	
Pass / Fail				Comments	
1. Pas 2. Pas 3. Pas 4. Pas 5. Pas 6. Pas	55 55 55 55 55 55				

Test	UATC011	Test	Modify new	Tested	Management
Case ID		Name	password	by	Team
Test Case Description To verify new password c			an successfully login		
Test Case	Instruction	Test Data			
<ol> <li>Log in the account.</li> <li>Clicks on the profile photo on the side navigation bar.</li> <li>The application will redirect to the profile page.</li> <li>Enters new valid password.</li> <li>Clicks on update button</li> </ol>				New 987654321	Password:
Expected I	Result			Actual Result	
The applic new passw	cation prompt ord message.	t the succes	sfully update	The applie the succes new passw	cation prompt ssfully update ord message.
Pass / Fail				Comments	;
1. Pas 2. Pas 3. Pas 4. Pas 5. Pas 6. Pas	55 55 55 55 55 55 55				

Test	UATC012	Test	View	all	Tested	Management	
Case ID		Name	visitation		by	Team	
			informatio	on			
Test Case	Description	To verify a	all the visit	ation	informatio	on is showed	
Test Case	Instruction				Test Data		
<ol> <li>Log in the account.</li> <li>Clicks on the visitors button on the side navigation bar.</li> <li>The application will redirect to the visitation page.</li> <li>Clicks on the total visitors' tabs</li> </ol>					-		
Expected	Result				Actual Result		
All the checked in visitation will be displayed.			All the visitation	checked in will be			
					displayed	•	
Pass / Fail	1				displayed Comment	IS	

Test	UATC013	Test	Search	Tested	Management	
Case ID		Name	visitation	by	Team	
			using unit id			
Test Case	Test Case DescriptionTo verify the related visitation is showed					
Test Case Instruction				Test Data		
1. Lo	g in the accou	nt.		Unit id: 87	,	
2. Cli	icks on the vis	itors button	on the side			
na	vigation bar.					
3. Th	e application	will redirect	to the			
vis	itation page.					
4. Cli	4. Clicks on the total visitors' tabs.					
5. En	ters unit id in	the search b	ar.			
Expected	Result			Actual Res	sult	

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

Test	UATC014	Test	View	all	Tested	Management	
Case ID		Name	announcer	nent	by	Team	
Test Case	Description	To verify	all announc	emen	t is showed		
Test Case Instruction					Test Data		
<ol> <li>Log in the account.</li> <li>Clicks on the announcement button on the side navigation bar.</li> <li>The application displays all the announcement</li> </ol>							
Expected	Result				Actual Result		
All ann	ouncements	are disp	layed on	the	All annou	incements are	
announce	ment.				displayed	on the	
					announce	ment.	
Pass / Fai	1				Comment	s	
1. Pa	ass						
2. Pa	iss						
3. Pa	ass						
4. Pa	iss						
5. Pa	iss						
6. Pa	ass						

Test	UATC015	Test	Add	and	Tested	Management	
Case ID		Name	publish	new	by	Team	
			announce	ement			
Test Case	Description	To verify	announcer	nent is	added and	published	
Test Case	e Instruction				Test Data		
1. Lo	og in the acco	unt.			Title:	Update	
2. Cl	licks on the a de navigation	nnounceme bar.	nt button o	on the	Swimmin	g Pool	
3. Tl	he application	displays al	l the		Opening 7	Гime	
ar 4. C	nouncement. licks on add n	ew announ	cement bu	tton.	Descriptio	on: The	
5. Ei	nters title, des	cription, an	d upload		operation	hours is	
in 6 C	nage. licks on the se	we and nub	lish buttor	h	updated to 8:00pm.		
0. 0.	neks on the se	ive and pub		1.	Image: pool.png		
Expected	Result				Actual Result		
The succe	essfully added	and public	shed mess	age is	The successfully added		
prompted	and redirect	to the anno	uncement	page.	and published message		
					is prompte	ed and redirect	
					to the	announcement	
					page.		
Pass / Fai	1				Comment	S	
1. Pa	ass						
2. Pa	ass						
$\begin{array}{c c} 3. & Pa \\ \hline 4 & Pa \end{array}$	iss						
5. Pa	155 155						
6. Pa	ass						

Test	UATC016	Test	Add	draft	Tested	Management	
Case ID		Name	annound	cement	by	Team	
Test Case	Description	To verify	announce	ement is	added and	save as drafted	
Test Case	e Instruction				Test Data		
1. Lo	og in the acco	unt.			Title: U	Jpdate Gym	
2. Cl	licks on the and the and the navigation	nnounceme bar.	nt button	on the	Room Op	ening Time	
3. Tl	he application	displays al	ll the		Description	on: The	
ar 4. C	nouncement. licks on add n	ew announ	cement b	utton.	operation	hours is	
5. Ei	nters title, des	cription, an	id upload		updated to	o 6:00pm.	
image. 6. Clicks on the save and publish button.				Image: gym.png			
Expected	Result	•			Actual Re	Actual Result	
The succe	essfully added	l and save a	as draft m	nessage	The succe	essfully added	
is promp	ted and redi	rect to the	announ	cement	and sav	re as draft	
page.					message	is prompted	
					and redi	rect to the	
					announce	ment page.	
Pass / Fai	1				Comment	s	
1. Pa	ass						
2. Pa	ass						
3. Pa	ass						
4. Pa	ass						
5. Pa 6. Pa	155 155						

Test	UATC017	Test	Update	and	Tested	Management
Case ID		Name	publish	draft	by	Team
			announce	ement		
Test Case	Description	To verify	announcen	nent is	added and	save as drafted
Test Case	Instruction				Test Data	
1. L	og in the acco	unt.			Descriptio	on: The
2. C	licks on the au	nnounceme bar.	nt button o	on the	managem	ent fee will be
3. T	he application	displays al	ll the		adjusted to	o RM800 from
ar	nouncement.				1 Jan 202.	3 onwards.
4. C.	nouncement	with title "N	e Manageme	nt		
Fe	ee Adjustmen	t"	iunugenie	110		
5. T	he application	redirects to	o the edit			
ar	nouncement	page.	1 .			
$\begin{bmatrix} 0. \\ -7 \\ 0 \end{bmatrix}$	licks on save	tion with the	he test data	l <b>.</b>		
Expected	Result				Actual Re	sult
The succ	essfully adde	d and pub	lish messa	ige is	The successfully added	
prompted	and redirect	to the anno	uncement	page.	and publi	sh message is
					prompted	and redirect to
					the annou	ncement page.
Pass / Fai	1				Comment	S
1. Pa	ass					
2. Pa	ass					
	ass					
4. Pa	155					
6. Pa	ass					

Test	UATC018	Test	View	Tested	Management	
Case ID		Name	announcement	by	Team	
Test Case	Description	To verify	announcement de	tail is showed		
Test Case	Instruction			Test Data		
<ol> <li>Log in the account.</li> <li>Clicks on the announcement button on the side navigation bar.</li> <li>The application displays all the announcement.</li> <li>Clicks on view button on the announcement with title "Management Fee Adjustment"</li> </ol>			_			
Expected	Result			Actual Re	sult	
The ap	plication re	edirects t	to the view	The	application	
announce	ment page	and the	title, updated	redirects to the view		
description	on in test case	e "update a	nd publish draft	announcement page and		
announce	ment" (UAT	C015), imag	ge are showed in	the title, updated		
the annou	ncement deta	ils page.		description in test case		
				"update and publish		
				draft a	nnouncement"	
				(UATC01	5), image are	
				showed	in the	
				announce	ment details	
				page.		
Pass / Fai	1			Comment	s	
1. Pa 2. Pa 3. Pa 4. Pa 5. Pa 6. Pa	155 155 155 155 155 155					

Test	UATC019	Test	Remove	Tested	Management
Case ID		Name	announcement	by	Team
Test Case	Description	To verify	announcement is	removed	
Test Case	Instruction			Test Data	
<ol> <li>Log in the account.</li> <li>Clicks on the announcement button on the side navigation bar.</li> <li>The application displays all the announcement.</li> <li>Clicks on remove button the announcement with the title "Management Fee Adjustment 2022"</li> <li>The application prompt the user to confirm remove announcement.</li> <li>Clicks on confirm delete button</li> </ol>				_	
Expected	Result			Actual Result	
The appl	ication prom	ot the succ	essfully deleted	The appli	cation prompt
message.				the succes	ssfully deleted
				message.	
Pass / Fai	1			Comment	S
<ol> <li>Pass</li> <li>Pass</li> <li>Pass</li> <li>Pass</li> <li>Pass</li> <li>Pass</li> <li>Pass</li> <li>Pass</li> </ol>					

Test	UATC020	Test	View all	Tested	Management	
Case ID		Name	feedback	by	Team	
Test Case	Description	To verify a	Ill feedback is	showed		
Test Case	Instruction			Test Data		
1. Lo	g in the accou	nt.		-		
2. Cli	cks on the fee	dback butto	n on the side			
nav	vigation bar.					
3. Th	e application	displays all 1	the feedback			
in	2 different tab	•				
4. Cli	cks on replied	l feedback ta	ıb.			
Expected 1	Result			Actual Result		
All the rep	lied feedback	is showed i	n a list.	All the replied feedback		
				is showed in a list.		
Pass / Fail				Comments	6	
1. Pas	SS					
2. Pas	SS					
3. Pass						
4. Pas	SS					
5. Pa	SS					
6. Pa	SS					

Test	UATC021	Test	Reply new	Tested	Management	
Case ID		Name	feedback	by	Team	
Test Case	Description	To verify r	new feedback i	s replied		
Test Case	Instruction			Test Data		
<ol> <li>Log in the account.</li> <li>Clicks on the feedback button on the side navigation bar.</li> <li>Clicks on the new feedback tab.</li> <li>Clicks on the reply button on the new feedback named "Roof Leaking"</li> <li>The application redirects to the reply feedback page.</li> <li>Clicks on reply button.</li> <li>Enters the reply message.</li> <li>Clicks on send reply button.</li> </ol>				-		
Expected 1	Result			Actual Result		
The succes	ssful updated	message is p	prompted and	The successful updated		
the applica	ation redirects	to feedback	page.	message is	message is prompted and	
				the application	ation redirects	
				to feedbac	k page.	
Pass / Fail				Comments	5	
1. Pas 2. Pas 3. Pas 4. Pas 5. Pas 6. Pas	55 55 55 55 55 55 55			Tester 2 should allo to 1 feedba	2 responded ows more reply ack.	

Test	UATC022	Test	View	Tested	Management	
Case ID		Name	replied	by	Team	
			feedback			
Test Case	Description	showed				
Test Case Instruction				Test Data		
<ol> <li>Log in the account.</li> <li>Clicks on the feedback button on the side navigation bar.</li> <li>The application displays all the feedback in 2 different tab.</li> <li>Clicks on replied feedback tab.</li> <li>Clicks on view button with the feedback name "Complaint on swimming pool operating hours"</li> <li>The application redirects to replied feedback page.</li> </ol>						
Expected I	Result			Actual Result		
The feedb	oack details a	and replied	message by	The feedba	ack details and	
administra	tor is showed			replied	message by	
				administra	tor is	
				showed		
Pass / Fail				Comments	\$	
1. Pas 2. Pas 3. Pas 4. Pas 5. Pas 6. Pas	55 55 55 55 55 55					

## User Acceptance Test (Mobile application)

Test	UATC023	Test	Submit	Tested by	Resident	
Case ID		Name	registration			
			form			
Test Case	Description	To verify t	he registration f	orm is added		
Test Case	Instruction			Test Data		
1. En	ter the applica	tion on a mo	obile device.	Name: Lily	7 Tan	
2. Cli 3. The	ck on log in a e mobile appli	s resident bu	utton. ects to the	IC no: 9804	401080456	
log	in as resident	page.		Email:		
4. Cli 5. Th	cks on sign up e application i	p button. redirects to t	he register as	lilytan@gn	nail.com	
res	idents page.		8	Password:	123456789	
6. Entern	ters name, ide ail address, pa	ntification c assword, car	ard number, plate, unit id,	Carplate: K	LDD8752	
address, and uploads supporting				Unit id: 812		
7. Cli	cuments. cks on signup	button.		Address: 8-1-2, Jalan		
	0 1			Sungai Long, 43000,		
				Kajang.		
				Supporting document:		
				water_bill.png		
Expected I	Result			Actual Res	ult	
A successf	ully registered	l message is	prompted, and	А	successfully	
the application	ation will red	irect the use	er to welcome	registered	message is	
page.				prompted,	and the	
				application	will redirect	
				the user	to welcome	
				page.		
Pass / Fail				Comments		
1. Pas	58					
2. Pas 3. Pas	SS SS					
4. Pas	58					
5. Pas 6. Pas	58 58					

Test Case	UATC022	Test	Login	Tested by	Resident	
ID		Name	account			
Test Case	Description	To verify the	he account is l	ogged in		
Test Case	Instruction			Test Data		
<ol> <li>Enter the application in a mobile device.</li> <li>Click on log in as resident button.</li> <li>Enter valid email and password.</li> <li>Click on log in button.</li> </ol>				Email: <u>resident@gmail.com</u> Password: 123456789		
Expected I	Result			Actual Result		
The user is	s authenticate	d and redire	cted to home	The user is authenticated		
page.				and redirected to home		
				page.		
Entry Crite	eria			Exit Criteria	a	
Pass / Fail	Pass / Fail					
7. Pass						
8. Pass						
9. Pass						
10. Pas	SS					
11. Pas	SS					
12. Pas	SS					

Test Case	UATC025	Test	Login	Tested by	Security	
ID		Name	account		guards	
Test Case ]	Description	To verify th	ne account is l	ogged in		
Test Case	Instruction			Test Data		
1. Enter the	e application i	n a mobile d	evice.	Email: <u>s@g</u>	mail.com	
2. Click on	log in as resi	dent button.		Password: 1	23456789	
3. Enter va	lid email and	password.				
Expected F	Result	•		Actual Result		
Linpetter	cosult.					
The user is	s authenticate	d and redired	cted to home	The user is authenticated		
page.				and redirected to home		
				page.		
Pass / Fail				Comments		
13. Pas	S					
14. Pas	S					
15. Pass						
16. Pass						
17. Pas	S					
18. Pas	S					

Test Case	UATC026	Test	Logout	Tested by	Security
ID		Name			guards
Test Case	Description	To verify the	he user is logg	ed out	
Test Case	Instruction	Test Data			
1. Clicks o	n the home pa	ige button.		-	
2. Clicks o	n the button o	t protile pict	ture.		
3. The app	n log out but	cts to prome	e page.		
Expected I	Result	511.		Actual Result	
The user is	redirected to	login page.		The user is redirected to	
				login page.	
Pass / Fail				Comments	
1. Pas	SS				
2. Pas	2. Pass				
3. Pass					
4. Pass					
5. Pas	SS				
6. Pass					

Test Cas	e UATC027	Test	Logout	Tested by	Residents	
ID		Name				
Test Cas	e Description	To verify t	he user is logg	ed out		
Test Cas	e Instruction			Test Data		
1. Clicks	on log out but	ton on home	page.	-		
Expected	l Result			Actual Result		
The user	is redirected to	login page.		The user is redirected to		
				login page.		
Pass / Fa	il			Comments		
1. F	ass					
2. Pass						
3. Pass						
4. Pass						
	ass					
6. Pass						

Test Case	UATC028	Test	Modify	Tested by	Resident
ID		Name	user profile		
Test Case I	Description	To verify p	hone number,	car plate, and	d password is
		updated and	l can successfu	ılly login aft	er change the
		password			
Test Case I	nstruction			Test Data	
1. Login th	e account			Phone	number:
2. Clicks of	n the home pa	ge button.		601542357	/89
3. Clicks of	n the button of	f profile pict	ure.	Carplate	number:
4. The appl	ication redire	cts to profile	page.	FS242	
5. Enter ne	w phone num	ber, carplate	number, and	Password: 987654321	
password.					
6. Clicks of	n edit profile l	outton.			
Expected R	lesult			Actual Res	ult
The new phone number, carplate number are				The new pl	none number,
updated an	d added mess	age is promp	t. The user is	carplate 1	number are
				updated.	The user is

automatically logged out and able to log in with	automatically logged
new password.	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	UATC029	Test	Add visitor	Tested by	Resident	
ID		Name				
Test Case I	Description	To verify n	ew visitors is	added	1	
Test Case I	nstruction	1		Test Data		
1. Login th	e account			Name: Josł	nua	
2. Clicks o	n the visitors	button on th	e bottom tab	Identificati	on card no:	
navigation	bar.			980201054	654	
3. Clicks of	n add visitors	button.		Phone	number:	
4. Enter v	visitor's name	, identificati	on card no,	601275189	75	
phone num	ber, date visit	ed, time visit	ed, car plate.	Date visited: 13		
5. Clicks of	n submit butto	on.		September 2022		
				Time visited: 12:00pm		
Expected R	lesult			Actual Result		
The succes	sfully added n	nessage will	be prompt.	The successfully added		
				message are prompt.		
Pass / Fail				Comments		
1. Pas	s					
2. Pas	S					
3. Pas	S					
4. Pas	S					
6. Pas	5 S					

Test Case	UATC030	Test	Add visitor	Tested by	Security
ID		Name			guard
Test Case I	Description	To verify no	ew visitors is a	added	
Test Case I	nstruction	1		Test Data	
1. Login th	e account.			Name: Onr	n Sha Sha
2. Clicks o	n register ad-l	hoc visitors ł	outton on the	IC no: 7804	415048756
home page.				Phone	number:
3. The appl	ication redired	ets to add vis	itor page.	016874654	5
4. Enters	visitor's name	e, visitor's IO	C no, phone	Car plate: JSD5456	
number, ca	r plate, unit id			Unit id: 846	
5. Clicks or	n submit butto	n.			
Expected R	lesult			Actual Result	
The succes	sfully added n	nessage will	be prompt.	The successfully added	
				message are prompt.	
Pass / Fail				Comments	
1. Pas	S				
2. Pas	S				
3. Pas	S				
4. Pas	S				
5. Pas	S				
6. Pas	S				

Test Case	UATC031	Test	View al	1 Tes	sted by	Resident
ID		Name	visitation			
Test Case I	Description	To verify a	ll visitation i	show	ed	
Test Case I	Instruction			Tes	st Data	
1. Login th	e account.			-		
2. Clicks o	on the visitors	button on th	ne bottom ta	<b>b</b>		
navigation	bar.					
Expected F	Result			Ac	Actual Result	
All the visi	tation informa	ation are sho	wed.	All	the	visitation
				inf	ormatio	n are
				shc	wed.	
Pass / Fail				Co	mments	
1. Pas	S					
2. Pass						
3. Pass						
4. Pass						
5. Pas	S					
6. Pass						

Test Case	UATC032	Test	View	Tested by	Resident	
ID		Name	visitation			
Test Case	Description	To verify the	ne visitation de	tail is showe	d	
Test Case	Instruction	1		Test Data		
1. Login th	e account.			-		
2. Clicks o	on the visitors	button on t	he bottom tab			
navigation	bar.					
3. Clicks o	n the view but	tton on visita	tion card with			
the visitor's	s named "Oak	lynn".				
4. The app	lication redire	cts to view v	visitors page.			
Expected F	Result			Actual Result		
The applie	cation will b	e navigated	to visitation	The application will be		
detail page	and the visito	or's details an	e showed.	navigated	to visitation	
				detail pag	ge and the	
				visitor's	details are	
				showed.		
Pass / Fail				Comments		
1. Pas	S					
2. Pass						
3. Pass						
4. Pas						
6. Pas	S					

Test Case	UATC033	Test	Remove	Tested by	Resident	
ID		Name	upcoming			
			visitation			
Test Case ]	Description	To verify the	he visitation is	removed		
Test Case	Instruction	1		Test Data		
1. Login th	e account.			-		
2. Clicks o	on the visitors	button on t	he bottom tab			
navigation	bar.					
3. Clicks o	on the remove	e button on	visitation card			
with the vi	sitor's named	"Apollo".				
4. The ap	plication pro	ompt a aler	t message to			
confirm us	er remove act	ion.				
5. Clicks o	n ok button.					
Expected I	Result			Actual Result		
The succes	ssfully deleted	d message w	vill be prompt	The	successfully	
and the vis	itation will be	e deleted from	n database.	deleted n	nessage are	
				prompt	and the	
				visitation	are deleted	
				from datab	ase.	
Pass / Fail		Comments				
1. Pas 2. Pas 3. Pas 4. Pas 5. Pas	55 55 55 55 55					
6. Pas	S					

Test Case	UATC034	Test	Check	in	Tested by	Visitor
ID		Name	visitation	1		
Test Case I	Description	To verify the	ne visitatio	n is s	successfully	checked in
Test Case I	nstruction	1			Test Data	
1. Enter the	e application c	on a mobile d	evice.		Visitation	id:
2. Click on	log in as visit	tors.			CY875645	
3. Enters vi	isitation id and	d unit id.			Unit ID:87	
4. The app	lication show	the generate	ed qr code	for		
this visitati	on.					
5. Show the	e qr code to se	curity guard				
Expected R	lesult				Actual Result	
The applic	ation will sh	ow the qr c	code with	the	The	application
visitation of	letail and abl	e to scan by	y the secu	ırity	showed th	ne qr code
guard.					with the	visitation
					detail and	able to be
					scanned by	the security
					guard.	
Pass / Fail					Comments	
1. Pas	S					
2. Pas	S					
3. Pass						
4. Pas	S					
6. Pas	s S					

Test Case	UATC035	Test	Verify	Tested by	Security	
ID		Name	check in		guard	
			visitation			
Test Case I	Description	To verify the	he check in visi	tation is ver	ified	
Test Case ]	Instruction			Test Data		
1. Login th	e account.			QR shown	by visitors	
2. Clicks o	n the verify re	gistered visi	tors button on			
the home p	bage.					
3. Scan the	e qr code show	n by visitors				
4. The ap	plication will	show the o	details of the			
visitation.						
5. Clicks o	n the verify bu	utton.				
Expected F	Result			Actual Result		
The applic	ation will be	navigated to	o the verified	The application will be		
visitors pa	age. The visi	itation upda	ted with the	navigated	to the	
checkin da	te and time an	d the securit	y guard id.	verified visitors page.		
				The visitation updated		
				with the c	checkin date	
				and time	and the	
				security gu	ard id.	
Pass / Fail				Comments		
1. Pas	S					
2. Pas	S					
3. Pas	S					
4. Pas 5 Pas	s					
6. Pas	S					

Test	UATC036	Test	View all	Tested	Resident
Case ID		Name	announcement	by	
Test Case	Description	To verify a	all announcemen	t is showed	
Test Case	Instruction			Test Data	
1. Login tl	he account.			-	
2. Clicks c	on the annound	cement butto	on on the bottom		
tab naviga	tion bar.				
Expected	Result			Actual Result	
All annou	ncement is she	owed.		All announcement is	
				showed.	
Pass / Fail				Comments	5
1. Pa	SS				
2. Pa	SS				
3. Pa	SS				
4. Pa	SS				
5. Pa	SS				
6. Pa	SS				

Test	UATC037	Test	View	Tested	Resident	
Case ID		Name	announcement	by		
Test Case	Description	To verify a	nnouncement de	tail is showe	ed	
Test Case	Instruction	I		Test Data		
1. Login th	ne account.			-		
2. Clicks o	on the annound	cement butto	on on the bottom			
tab naviga	tion bar.					
3. The app	lication redire	ects to annot	uncement page.			
4. Clicks o	on the view mo	ore button of	n announcement			
with the ti	tle "Updated ]	Managemen	t Fee Notice."			
Expected I	Result			Actual Result		
The applic	ation will nav	igate to view	v announcement	The application		
page and	the details	of the ann	nouncement are	navigates	to view	
showed.				announcement page		
				and the details of the		
				announcen	nent are	
				showed.		
Pass / Fail				Comments	5	
1. Pas	ss					
2. Pass						
3. Pass						
4. Pas	55					
6 Pas	55 55					

Test Case	UATC038	Test	View a	11	Tested by	Resident	
ID		Name	feedback				
Test Case I	Description	s s	howed				
Test Case Instruction					Test Data		
1. Login th	e account.				-		
2. Clicks o	n the feedbacl	x button on t	he bottom ta	b			
navigation bar.							
Expected Result					Actual Result		
All the feedbacks are showed.				All the fe	edbacks are		
				showed.			
Pass / Fail				Comments			
1. Pas	S						
2. Pass							
3. Pas	S						
4. Pas	S						
5. Pas	S						
6. Pas	S						

298

Test Case	UATC039	Test	View	Tested by	Resident	
ID		Name	feedback			
Test Case 1	Description	To verify fo	eedback detail	is showed		
Test Case Instruction				Test Data		
1. Login th	e account.			-		
2. Clicks o	n the feedbacl	k button on t	he bottom tab			
navigation	bar.					
3. The app	lication redire	cts to feedba	ck page.			
4. Clicks o	n the view mo	ore button on	the feedback			
with the title "Rooftop leaking issue".						
Expected Result			Actual Res	ult		
The application will navigate to view feedback			The	application		
page and all the feedback details and replies will			navigates	to view		
show.			feedback page and all			
				the feedback details and		
			replies are	showed.		
Pass / Fail			Comments			
1. Pas	s					
2. Pas	S					
3. Pas	S					
4. Pas	S					
5. Pas	s s					

Test Case	UATC040	Test	Add new	Tested by	Resident		
ID		Name	feedback				
Test Case D	Description	ed					
Test Case Instruction				Test Data			
1. Login the	e account.	Title: Complaint on					
2. Click on	the feedback	button on t	he bottom tab	Level 7 Neighbour			
navigation l	bar.			Description	n: Always		
3. The appli	ication redire	cts to feedba	ck page.	hear the n	oise coming		
4. Click on	the add butto	n on the feed	lback page.	from my upstairs after			
5. The application redirects to add feedback page.				12am.			
6. Enter title, description, categories of feedback.			Categories	: others			
7. Clicks or	n submit butto						
Expected Result				Actual Res	ult		
The success	sfully added r	nessage will	be prompt.	The succes	The successfully added		
			message is prompt.				
Pass / Fail			Comments				
1. Pass							
2. Pass	5						
3. Pass	5						
4. Pass							
5. Pass 6 Pass	5						

## User Acceptance and Satisfactory Form (Web application)

User Acceptance and Sa	tisfactory Form
Hi, I am Low Ee Lyne, an undergraduate student of Uni pursuing on Bachelor of Science (Honours) Software I conducted as part of the data collection of the course objective of this survey is to collect user acceptance a and Visitor Management System.	versiti Tunku Abdul Rahman (UTAR) Engineering. This survey is of Final Year Project. The main nd feedback towards the Resident
Resident and Visitor Management System consists of application. The web application is to help the manage administrative work while the mobile application is to between the management teams and residents.	web application and mobile ement team simplify their facilitate the communication
This survey requires you as a management team ment Visitor Management System through this link (https:// answering it. This web application is responsive, you a computer device. This survey contains 3 sections.	bers to explore on the Residents and rvsm-rho.vercel.app) before re able to use it on mobile, tablet or
Your details and responses will be kept entirely confid- information collected will be kept anonymously and th for academic practice and research purposes.	ential, the questionnaire and the e feedback received are only used
If you have any questions about this survey questions email address (eelyne1099@1utar.my). I am willing to require further explanation. I appreciate your cooperat providing precious time, accurate information and tho questionnaire.	aire, kindly contact me through the answer or clarify any questions that ion in completing this survey and ughtful suggestions to complete this
Yours faithfully,	
Low Ee Lyne	
Lee Kong Chian Faculty of Engineering and Science	
Universiti Tunku Abdul Rahman (UTAR)	
Sign in to Google to save your progress. Learn more	
*Required	
Which age group are you in?*	
O 20-30	
O 31-40	
O 41-50	
O >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? *	
<ol> <li>Enter the application in web browser.</li> <li>Enter the valid email and password. (Email: admin@gmail.com, Password: 123456)</li> <li>Clicks on login button.</li> </ol>	789)
O Yes	
O 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.

O Yes

O 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.

O Yes

O No

Able to view all resident's information *
<ol> <li>Log in the account.</li> <li>Clicks on residents' button on the side navigation bar.</li> <li>Clicks on the existing residents tab.</li> </ol>
O Yes
O No

Able	to 1	/iew	all	ad	min	115	tra	tor	8.*

1. Log in the account.

2. Clicks on administrator's button on the side navigation bar.

) Yes

) No

Able to add n	ew adr	minist	rator *
---------------	--------	--------	---------

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.

 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

O 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.

O Yes

O No
Able to view all security guard \*

1. Log in the account.

2. Clicks on security guard's button on the side navigation bar.

O Yes

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.

O Yes

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.

O Yes

O 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.

O Yes

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.

O Yes

O 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)

O Yes

O 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.

O Yes

O 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

Able to	add d	raft	ann	oun	cen	nent *
---------	-------	------	-----	-----	-----	--------

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.

O Yes

O 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.
- Edit the description. (Description: The management fee will be adjusted to RM800 from 1 Jan 2023 onwards.)
- 7. Clicks on save and publish button.

	~	1.2.1	
	- A.	· V	0.0
×.			60
-	~		

O 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.

 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

O Yes

O 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.

O Yes

O 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

1. Log in the account.	
2. Clicks on the feedback button on the side navigation bar.	
J. Clicks on the new feedback tab. A. Clicks on the renk button on the new feedback named "Roof Least	daa*
<ol> <li>Circles of the reply ballot of the reply feedback harries noor cear</li> <li>The application redirects to the reply feedback page.</li> </ol>	ung.
6. Clicks on reply button.	
7. Enters the reply message.	
8. Clicks on send reply button.	
O Yes	
O No	

This form was created inside Universiti Tunku Abdul Rahman. Report Abuse

## User Acceptance and Satisfactory Form

Sign in to Google to cave your progress. Learn more

\*Required

	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
This application will he	elp you e	asier to	manage	residen	t's inform	nation. *
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
This application will he	e <mark>lp y</mark> ou e	asier to	manage	visitor's	informa	tion *
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
This application will he	elp you e	asier to	publish a	announc	ement. *	
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
This application will he	elp you e	asier to	manage	residen	t's feedb	ack. *
This application will he	elp you e 1	asier to 2	manage 3	residen 4	t's feedb 5	ack. *

	T	2	3	4	5	
	ji) Vest	975) 1427	100	- 8 79277		
Strongly Disagree	0	0	0	0	0	Strongly Agree
This application will he	lp you to	o simply	adminis	trative v	vork. *	
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
Rate the overall user in	terface	design *				
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
Rate the accuracy of d	ata in th	e applica	ation. *			
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
Rate the satisfaction le	evel in th	e applic	ation. *			
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
As a user of the web a ( Sign off - you agree tha free and ready for deploy O Yes O No	pplicatic at they ha yment)	on, would ve thorou	d you sig ughly test	n off the	e user ac olution, ar	cceptance test? * nd agree that it is bug-

This form was created inside of Universiti Tunku Abdul Rahman. Report Abuse

309



#### 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 Abdol 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

*Required	gle to cave your progress. Learn more	
Visitors		
Able to chec	k in visitations *	
1. Enter the aj	plication on a mobile device.	
<ol> <li>Enters visit</li> </ol>	ation id and unit id. (Visitation id: CY8756	45, Unit ID:87)
<ol> <li>The applica</li> <li>Show the q</li> </ol>	tion show the generated qr code for this r code to security guard.	visitation.
○ Yes		
○ No		

### 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.		
<ol> <li>Enter valid email and password. (Email: s@gmail.com, Password: 123456789)</li> <li>Click on log in hyttop</li> </ol>		
O Yes O No		
Able to verify check in visitation? *		
1. Login the account.		
<ol> <li>Clicks on the verify registered visitors button on the home page.</li> <li>Count the new device the second seco</li></ol>		
Scan the gr code shown by visitors.     A The application will show the details of the visitation		
5. Clicks on the verify button.		
O Yes		
O No		

Able to add ad-hoc visitors *	
1. Login the account.	
2. Clicks on register ad-hoc visitors button on the home pa	age.
<ol><li>The application redirects to add visitor page.</li></ol>	
<ol><li>Enters visitor's name, visitor's IC no, phone number, car</li></ol>	plate, unit id.
(Name: Onn Sha Sha, IC no: 780415048756, phone numbe	r: 0168746545, carplate: JSD5456,
5 Clicks on submit button	
O Yes	
O No	
Able to logout account *	
1. Clicks on log out button on home page.	
O Yes	
O No	
Back Submit	Clear for
This form was created inside Universiti Tunko Abdu	Rahman. <u>Report Abuse</u>

### For residents

Sign in to Google to save yo	our progress. Learn more
Required	
Resident	
Able to submit registration	on form *
1. Enter the application on a	a mobile device.
2. Click on log in as residen	t button.
<ol> <li>The mobile application re Clicks on sign up button     </li> </ol>	edirects to the login as resident page.
<ol> <li>The application redirects</li> </ol>	to the register as residents page
6. Enters name, identificatio	on card number, email address, password, car plate, unit id,
address, and uploads supp	orting documents.
(Name: Lily Tan, IC no: 980-	401080456, Email: lilytan@gmail.com, Password: 123456789
Carplate: KDD8752, Unit id: Supporting document: wate	812, Address: 8-1-2, Jalan Sungai Long, 43000, Kajang., at bill png)
7. Clicks on signup button.	bin.pig/
O Yes	
0	
() No	
Able to login account *	
1. Enter the application in a	mobile device.
2. Click on log in as residen	t button.
3. Enter valid email and pas	sword. (Email: resident@gmail.com, Password: 123456789)
<ol><li>Click on log in button.</li></ol>	
O Yes	
O 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.

O Yes

O 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.

O Yes

O 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.

 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)
 Clicks on submit button.

O Yes

Δ	ble to view all visitation *
1	Login the account
2.	Clicks on the visitors button on the bottom tab navigation bar.
C	) Yes
C	) No
A	ole 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.
C	) Yes
C	) No
Al 1. 2 3. 4. 5. C	ble to remove upcoming visitation * Login the account. Clicks on the visitors button on the bottom tab navigation bar. Clicks on the remove button on visitation card with the visitor's named 'Apollo'. The application prompt a alert message to confirm user remove action. Clicks on ok button. ) Yes ) No
A	ole to view all announcement *
1.	Login the account.
2	Clicks on the announcement button on the bottom tab havigation bar.
C	) Yes
C	N Ma

C47	
Able to view ann	ouncement *
1. Login the accou	nt.
2. Clicks on the an	nouncement button on the bottom tab navigation bar.
3. The application	redirects to announcement page.
<ol> <li>Clicks on the vie Notice."</li> </ol>	w more button on announcement with the title. Updated Management
O Yes	
O No	
0	
Able to view all fe	eedback *
1. Login the accou	nt.
2. Clicks on the fee	dback button on the bottom tab navigation bar.
0 ***	
O res	
O No	
9.000 (1.000 (1.00)	
Able to view feed	fback *
1. Login the accou	nt.
2. Clicks on the fee	dback button on the bottom tab navigation bar.
3. The application	redirects to feedback page.
4. Glicks on the vie	winnie button on the recoustick with the thre. Roonop reaking issue .
O Yes	
STREET STREET	
-	

1. Login the acc	ount.
2. Click on the fe	edback button on the bottom tab navigation bar.
<ol> <li>The application</li> <li>Click on the a</li> </ol>	n rearects to reedback page. dd button on the feedback page
5. The applicatio	on redirects to add feedback page.
6. Enter title, des	cription, categories of feedback. (Title: Complaint on Level 7 Neighbor,
Description: Alw	ays hear the noise coming from my upstairs after 12am. Categories: others)
7. Clicks on sub	mit button.
~	
() Yes	
O Yes	
O Yes	

This form was created inside Universiti Tunku Abdul Rahman. Report Abuse

# User Acceptance and Satisfactory Form

Sign in to Google to save your progress. Learn more \*Required

Share your feedback after performing the test.

	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
his application will he	alp you to	o manag	e your v	isitor's ir	formatio	on. *
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
his application will he nanagement team.	lp you e	asier to	get the l	atest inf	ormation	from
his application will he nanagement team.	elp you e	asier to 2	get the I	atest inf	ormation 5	from
his application will he nanagement team. Strongly Disagree	lp you e	2	get the I 3 O	atest inf	ormation 5	from Strongly Agree
his application will he nanagement team. Strongly Disagree his application will he eam.	elp you e	asier to	get the I 3 O	atest inf 4 O	ormation 5 O	from Strongly Agree
This application will he management team. Strongly Disagree This application will he eam.	lp you e	2 O asier to	get the I 3 O	atest inf 4 O	ormation 5 O	from Strongly Agr

	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
Rate the overall user in	terface	design *				
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agree
Rate the accuracy of d	ata in th	e applica	ation. <mark>*</mark>			
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agre
Rate the satisfaction le	evel in th	e applic	ation. *			
	1	2	3	4	5	
Strongly Disagree	0	0	0	0	0	Strongly Agre

#### APPENDIX D: User Acceptance Test Feedback Result

### User Acceptance and Satisfactory Form Result (Web application)



### User Acceptance and Satisfactory Form

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





















This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy.

Google Forms

### User Acceptance and Satisfactory Form Result (Mobile application)















Share your feedback after performing the test.







This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

338