DESIGN AND DEVELOP OF A CONDO PROPERTY MANAGEMENT SYSTEM WITH MOBILE APPLICATION AND WEB-BASED MANAGEMENT DASHBOARD

TEH KEH BOON

UNIVERSITI TUNKU ABDUL RAHMAN

DESIGN AND DEVELOP OF A CONDO PROPERTY MANAGEMENT SYSTEM WITH MOBILE APPLICATION AND WEB-BASED MANAGEMENT DASHBOARD

TEH KEH BOON

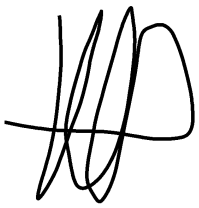
A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering

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

April 2020

DECLARATION

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

Signature : 

Name : Teh Keh Boon

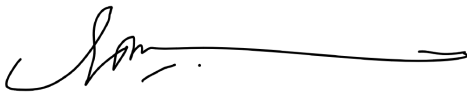
ID No. : 16UEB01628

Date : 24/04/2020

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DESIGN AND DEVELOP OF A CONDO PROPERTY MANAGEMENT SYSTEM WITH MOBILE APPLICATION AND WEB-BASED MANAGEMENT DASHBOARD"** was prepared by **TEH KEH BOON** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : Dr. Simon Lau Boung Yew

Date : 24/04/2020

ABSTRACT

It is a tough job to manage condominium and gate communities. The traditional way of managing condo property is unable to handle a lot of issues efficiently. Hence, we should make use of Information and Communication Technology to manage the condo property more smartly and easily. This will be able to help the management communicate with tenant easily, while the tenant can report issues, get latest announcement and pay the bill without going outside. The adopted software development methodology is rapid application development as the integration can be built throughout the process. The project will be involved in developing web application and mobile app for different users. In a nutshell, this project had achieved the objectives by using the solution proposed.

TABLE OF CONTENTS

DECLARATION **ii**

APPROVAL FOR SUBMISSION **iii**

ABSTRACT **v**

TABLE OF CONTENTS **vi**

LIST OF TABLES **x**

LIST OF FIGURES **xiv**

LIST OF APPENDICES **xviii**

CHAPTER

1 INTRODUCTION **19**

1.1 Introduction 19

1.2 Background 19

1.3 Problem Statement 19

1.3.1 Defective Data 20

1.3.2 Time Consuming 20

1.3.3 High Cost 20

1.4 Project Objectives 20

1.5 Project Solution 21

1.5.1 Laravel 22

1.5.2 MySQL 22

1.5.3 Flutter 22

1.6 Project Approach 23

1.7 Scope of the Project 24

1.7.1 Login 24

1.7.2 Create and delete account 24

1.7.3 Work Order 24

1.7.4 Announcement 24
1.7.5 Bill 24
1.7.6 Facilities 24

2 LITERATURE REVIEW 25
2.1 Introduction 25
2.2 Review on Documentation Management System 25
2.2.1 Traditional Documentation Management System 25
2.2.2 Electronic Documentation Management System 25
2.2.3 Evaluation 26
2.3 Review on Existing System 26
2.3.1 i-Neighbour 26
2.3.2 EcoWorld Community 27
2.3.3 Setia Community 28
2.3.4 Evaluation 29
2.4 Review on Software Development Methodologies 30
2.4.1 Waterfall Methodology 30
2.4.2 Agile 31
2.4.3 Rapid Application Development 33
2.4.4 Evaluation 34

3 METHODOLOGY AND WORK PLAN 35
3.1 Introduction 35
3.2 Methodology 35
3.3 Project Plan 36
3.3.1 Work Breakdown Structure 36
3.3.2 Gantt Chart 37

4 PROJECT INITIAL SPECIFICATION 40
4.1 Introduction 40
4.2 Fact Finding 40
4.2.1 Work Order 40
4.2.2 Announcement 40

4.2.3 Bill 41
4.3 Use Case 41
4.3.1 Use Case Diagram 41
4.3.2 Use Case Description 42
4.4 Project Requirement 46
4.4.1 Non Functional Requirements 46
4.4.2 Functional Requirements 47

5 DESIGN 48
5.1 Introduction 48
5.2 Software Architecture Design 48
5.3 Database design 49
5.3.1 Entity Relational Diagram (ERD) 49
5.3.2 Data Dictionary 51
5.4 System Design 56
5.4.1 Activity Diagram 56
5.4.2 Data Flow Diagram 70
5.5 User Interface Design 72
5.5.1 Web System 72
5.5.2 Mobile Application 74

6 IMPLEMENTATION 75
6.1 Introduction 75
6.2 Module for Admin 76
6.2.1 Login 76
6.2.2 Tenant Management 76
6.2.3 Announcement Management 78
6.2.4 Work Order Management 79
6.2.5 Bill Management 80
6.2.6 Reservation Management 81
6.3 Module for Tenant 82
6.3.1 Login 82
6.3.2 View Announcement 83

6.3.3 Payment 84
6.3.4 Booking 85
6.3.5 Problem Report 86
6.3.6 Profile Management 87

7 SOFTWARE TESTING 88
7.1 Introduction 88
7.2 Unit Test 88
7.3 Backend Testing (API) 99
7.4 Integration Testing 104
7.5 User Acceptance Test 109
7.5.1 UAT Test Case 109
7.5.2 UAT Test Result 113
7.6 Usability Test 114
7.6.1 Usability Test Form 114
7.6.2 Usability Test Result 116

8 CONCLUSIONS AND RECOMMENDATIONS 118
8.1 Conclusions 118
8.2 Recommendations for future work 119

REFERENCES 120

APPENDICES 122

LIST OF TABLES

Table 2.1: Comparison between existing systems. 29

Table 2.2: Comparison between software development methodologies 34

Table 3.1: Description on each stage of Rapid Application Development 35

Table 4.1: Login Use Case Description 42

Table 4.2: Create Account Use Case Description 42

Table 4.3: Delete Account Use Case Description 43

Table 4.4: Upload Announcement Use Case Description 43

Table 4.5: View Announcement Use Case Description 43

Table 4.6: View Work Order Use Case Description 43

Table 4.7: Request Work Order Use Case Description 44

Table 4.8: Update Work Order Use Case Description 44

Table 4.9: View Facility Use Case Description 44

Table 4.10: Book Facility Use Case Description 45

Table 4.11: View Bill as Tenants Use Case Description 45

Table 4.12: View Bill as Management Team Use Case Description 45

Table 4.13: Pay Bill Use Case Description 46

Table 5.1: Data Dictionary for Users 51

Table 5.2: Data Dictionary for Tenants 51

Table 5.3: Data Dictionary for Announcements 52

Table 5.4: Data Dictionary for Buildings 52

Table 5.5: Data Dictionary for Floors 52

Table 5.6: Data Dictionary for Units 53

Table 5.7: Data Dictionary for Bills 53

Table 5.8: Data Dictionary for Statuses 54

Table 5.9:Data Dictionary for Categories 54

Table 5.10: Data Dictionary for WorkOrders 54

Table 5.11: Data Dictionary for Facilities 55

Table 5.12: Data Dictionary for Timeslots 55

Table 5.13: Data Dictionary for Bookings 55

Table 6.1: Module list for each user 75

Table 7.1: Unit Test Case for Admin (Login) 88

Table 7.2: Unit Test Case for Admin (Tenant Management) 90

Table 7.3: Unit Test Case for Admin (Announcement Management) 91

Table 7.4: Unit Test Case for Admin (Work Order Management) 91

Table 7.5: Unit Test Case for Admin (Bill Management) 92

Table 7.6: Unit Test Case for Admin (Reservation Management) 93

Table 7.7: Unit Test Case for Tenant (Login) 94

Table 7.8: Unit Test Case for Tenant (View announcement) 94

Table 7.9: Unit Test Case for Tenant (Payment) 95

Table 7.10: Unit Test Case for Tenant (Booking announcement) 96

Table 7.11: Unit Test Case for Tenant (Problem report) 97

Table 7.12: Unit Test Case for Tenant (Profile management) 98

Table 7.13: Backend Test Case (Login) 99

Table 7.14: Backend Test Case (Get Announcement) 99

Table 7.15: Backend Test Case (Get Bill) 99

Table 7.16: Backend Test Case (Pay Bill) 100

Table 7.17: Backend Test Case (Get Booking) 100

Table 7.18: Backend Test Case (Store Booking) 100

Table 7.19: Backend Test Case (Get Facility List) 101

Table 7.20: Backend Test Case (Get Available Timeslot) 101

Table 7.21: Backend Test Case (Get WorkOrder) 101

Table 7.22: Backend Test Case (Store WorkOrder) 102

Table 7.23: Backend Test Case (get WorkOrder's detail) 102

Table 7.24: Backend Test Case (Get Tenant's Detail) 102

Table 7.25: Backend Test Case (Update Tenant's Detail) 103

Table 7.26: Backend Test Case (Update Tenant's Password) 103

Table 7.27: Integration Test Case for Admin (Login) 104

Table 7.28: Integration Test Case for Admin (Tenant Management) 104

Table 7.29: Integration Test Case for Admin (Announcement Management) 105

Table 7.30: Integration Test Case for Admin (Work Order Management) 105

Table 7.31: Integration Test Case for Admin (Bill Management) 106

Table 7.32: Integration Test Case for Admin (Reservation Management) 106

Table 7.33: Integration Test Case for Tenant (Login) 106

Table 7.34: Integration Test Case for Tenant (Announcement) 107

Table 7.35: Integration Test Case for Tenant (Bill) 107

Table 7.36: Integration Test Case for Tenant (Booking) 107

Table 7.37: Integration Test Case for Tenant (Problem Report) 108

Table 7.38: Integration Test Case for Tenant (Profile) 108

Table 7.393: Test Case for Web System (Login and Logout) 109

Table 7.40: Test Case for Web System (Tenant and Unit) 109

Table 7.41: Test Case for Web System (Bill) 110

Table 7.42: Test Case for Web System (Work order) 110

Table 7.43: Test Case for Web System (Announcement) 110

Table 7.44: Test Case for Web System (Booking) 111

Table 7.45: Test Case for mobile application (Authentication) 111

Table 7.46: Test Case for mobile application (Announcement) 111

Table 7.47: Test Case for mobile application (Bill) 112

Table 7.48: Test Case for mobile application (Reservation) 112

Table 7.49: Test Case for mobile application (Problem) 112

Table 7.50: UAT Result for web system 113

Table 7.51: UAT Result for mobile application 113

Table 7.52: Usability Test Form for Web System 114

Table 7.53: Usability Test From for Mobile Application 115

Table 7.54: Usability Test Result for Web System 116

Table 7.55: Usability Test Result for Mobile Application 117

LIST OF FIGURES

Figure 1.1: High Level Architecture System of the project. 21

Figure 1.2: The process of Rapid Application Development 23

Figure 2.1: i-Neighbour Mobile Appilcation Screenshot 26

Figure 2.2: EcoWorld Community Mobile Appilication Screenshot 27

Figure 2.3: Setia Community Mobile Appilication Screenshot 28

Figure 2.4: The process of Waterfall Methodology 30

Figure 2.5: The process of Extreme programming 31

Figure 2.6: The process of Extreme programming 33

Figure 3.1: Work Breakdown Structure 36

Figure 3.2: Tasks of Gantt Chart 37

Figure 3.3: Gantt Chart For Analysis and Quick Design Stages 37

Figure 3.4: Gantt Chart For Prototype Cycle Stages I 38

Figure 3.5: Gantt Chart For Prototype Cycle Stages II 38

Figure 3.6: Gantt Chart For Testing Stage 39

Figure 3.7: Gantt Chart For Implementation Stage 39

Figure 4.1: Use Case Diagram 41

Figure 5.1: Software Architecture Design 48

Figure 5.2: Conceptual ERD 49

Figure 5.3: Physical ERD 50

Figure 5.4: Activity diagram for admin(login) 56

Figure 5.5: Activity diagram for admin(view tenant account) 56

Figure 5.6: Activity diagram for admin(delete tenant account) 57

Figure 5.7: Activity diagram for admin(create tenant account) 58

Figure 5.8: Activity diagram for admin(add new bill) 59

Figure 5.9: Activity diagram for admin(record payment) 59

Figure 5.10: Activity diagram for admin(view bill record) 60

Figure 5.11: Activity diagram for admin(view booking record) 60

Figure 5.12: Activity diagram for admin(view workorder) 61

Figure 5.13: Activity diagram for admin(update workorder) 61

Figure 5.14: Activity diagram for admin(handle announcement) 62

Figure 5.15: Activity diagram for admin(delete workorder) 62

Figure 5.16: Activity diagram for admin(delete booking record) 63

Figure 5.17: Activity diagram for tenant (log in) 64

Figure 5.18: Activity diagram for tenant (edit profile) 64

Figure 5.19: Activity diagram for tenant (change password) 65

Figure 5.20: Activity diagram for tenant (view announcement) 65

Figure 5.21: Activity diagram for tenant (view bill) 66

Figure 5.22: Activity diagram for tenant (pay bill) 66

Figure 5.23: Activity diagram for tenant (view reservation) 67

Figure 5.24: Activity diagram for tenant (create new reservation) 67

Figure 5.25: Activity diagram for tenant (view problem report) 68

Figure 5.26: Activity diagram for tenant (view detail of problem report) 68

Figure 5.27: Activity diagram for tenant (create problem report) 69

Figure 5.28: Data Flow Diagram for web system 70

Figure 5.29: Data Flow Diagram for mobile application 71

Figure 5.30: Single Section Design 72

Figure 5.31: Multiple Section Design 73

Figure 5.32: Multiple Section Design with pop up dialog box 73

Figure 5.33: Layout Design of Mobile Application 74

Figure 6.1: Login Page for Admin 76

Figure 6.2: Create Tenant Account Page 76

Figure 6.3: View Unit List 77

Figure 6.4: View Tenant Account Page 77

Figure 6.5: View Announcement List Page 78

Figure 6.6: Create Announcement Page 78

Figure 6.7: View WorkOrder List Page 79

Figure 6.8: View WorkOrder Detail and Update WorkOrder 79

Figure 6.9: Add new Bill 80

Figure 6.10: Record Payment 80

Figure 6.11: View Bill Record Page 81

Figure 6.12: Reservation Page 81

Figure 6.13: Login Page for tenant 82

Figure 6.14:Announcement Page 83

Figure 6.15:Bill Page 84

Figure 6.16:Payment page 84

Figure 6.17:Reservation Page 85

Figure 6.18:Create new reservation Page 85

Figure 6.19:Problem Page 86

Figure 6.20:Report problem page 86

Figure 6.21:View Problem Page 01 86

Figure 6.22:View Problem Page 02 86

Figure 6.23:Account Page 87

Figure 6.24:Change Password Page 87

Figure 7.1: Evaluation with the SUS scores. (Sauro, 2018)

LIST OF APPENDICES

APPENDIX A: Interview Result 122

APPENDIX B: User Acceptance Test Form 123

APPENDIX C: Usability Test Form 128

CHAPTER 1

INTRODUCTION

1.1 Introduction

The background of the condo property management system will be introduced in this chapter, while the problem statement and objectives of the project also are defined. and defining. Then, the project solution and project approach are proposed to reach the project's objective, while the scope of this project can be covered.

1.2 Background

Nowadays, it can be very expensive to live in a single-detached home in Malaysia. So, many Malaysian are opting for condo units rather than traditional houses. However, it is a tough job to manage condominium and gate communities.

The manager is needed to follow up on the maintenance problems and keep up with the paperwork. Besides, the management office will face the foot traffic and a lot of phone calls for the maintenance requests and facility booking. The management officers are also unable to send notification and any status instantly to the tenants and owners.

The traditional way of managing condo property is unable to handle a lot of issues efficiently. Hence, we should make use of Information and Communication Technology to manage the condo property more smartly and easily.

1.3 Problem Statement

Most of the condo property management teams still use the old traditional ways to manage the properties and serve the owners. However, it is unable to provide a great experience for the management team and owners as the old and manual traditional management systems are inefficient and time-consuming. In contrast, the management needs a computerized management system to make a lot of work that can be completed automatically and conveniently.

1.3.1 Defective Data

Data in an old traditional management system may be incomplete and outdated. There is a risk of serious problems that cause the data to get out of the order, for instance, the file is placed in the wrong document accidentally or the file is taken and forgot to return. So, it can avoid the lost data and unnecessary duplicate copy of data by using the electronic management system. Besides, all content will be backed up in an electronic management system, it also able to retrieve any accidentally deleted file from the backup medium.

1.3.2 Time Consuming

Sometimes it cost a plenty of time to access data and locate certain files in a huge paper filing system. According to Borowski (2015), the employees spend at least 6 hours looking for paper documents only during the average working week. It also decreases the efficiency of the operating process. Furthermore, the files in the traditional system are unable to edit easily or send directly. Thus, the user needs to spend more time to create new copies or redo to update old files.

1.3.3 High Cost

Since the traditional management system will spend a lot of paper, it costs money to buy different sizes of paper and store the paper. Besides, it also needs many office supplies such as stationary, ink cartridge and printer. Any paper documentation is unable to edit, the management needs to print again, but the digitized document in the database is easy to edit and transmit. Instead of printing and posting on the boards, the management team can send the announcement via the app.

1.4 Project Objectives

The goals of this project are going to build a condo property management system that able:

- To provide an efficient solution to follow up on the progress of the maintenance problem without any paperwork.
- To provide an integrated solution to accept and respond to the maintenance requests and facility bookings.
- To allow property the tenants to view the billing statement and pay the payment via mobile application.
- To allow property owners and tenants to receive instant notification and status updates.

1.5 Project Solution

The proposed solution of the web application is developed by using Laravel and MySQL, while the mobile application is developed by React Native. The front end is used to allow users to view and interact with the system, while Back end receives and processes the requests from users and sends the data back to the users. Besides, the back end also manages the way to store the data in the database. The database is necessary for a completed application as the data that stored in the database is easy to locate and modify.

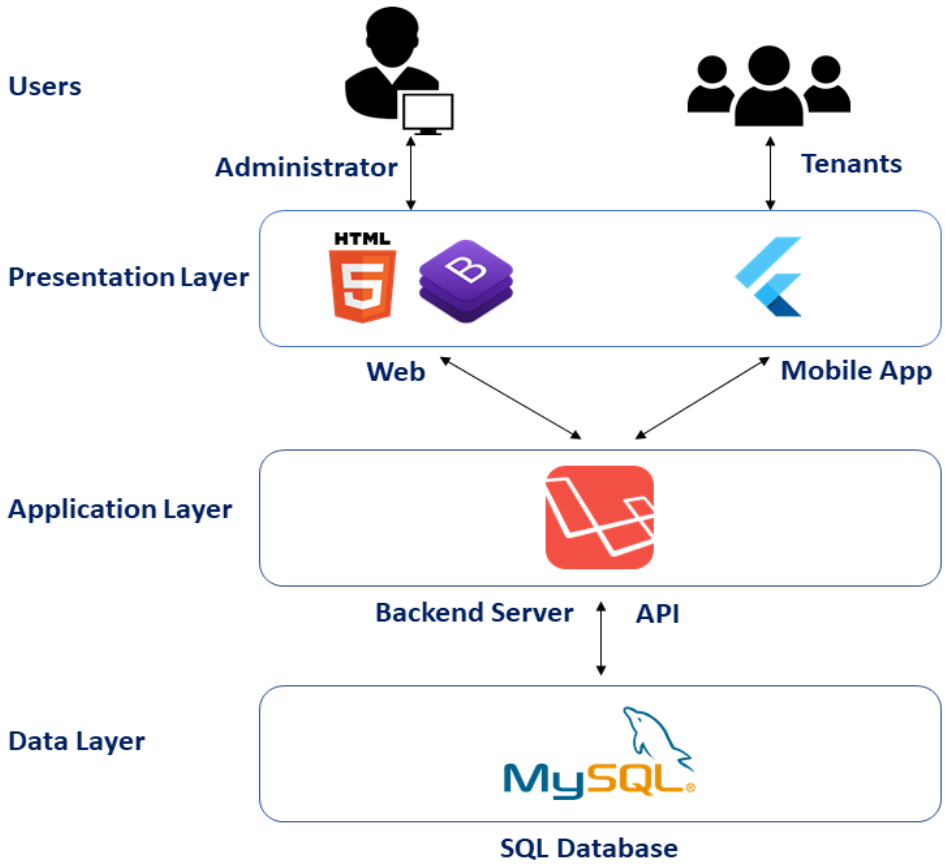


Figure 1.1: High Level Architecture System of the project.

1.5.1 Laravel

Laravel is an open-source PHP framework that will help the developer efficiently create web software. The blade template engine is one of the Laravel's features and it composes the code in an improved manner. Besides, developers are able to modify the application's database schema by using the Migration system from the Laravel (Otwell, 2019).

1.5.2 MySQL

One of the world's common relational database management system is MySQL (Wallen, 2019). It is open-source and used in a wide range of applications and set up easily in various environments such as Windows, Linux or Unix. MySQL is one of the most common refreshing dialects for web advancement as it is agreeable to PHP.

1.5.3 Flutter

It is a Google's open-source framework to build a native mobile application. There are a lot of languages is needed on different platforms, for instance, Swift for iOS and Java for Android. However, developers also can use Dart language and Flutter Framework for rendering cross platform native mobile application.

1.6 Project Approach

Rapid Application Development (RAD) is chosen as the project approach because it can minimize the planning time and emphasize the prototype iterations. RAD is a form of Agile project management strategy which common in software development. (Kelsie, 2017）

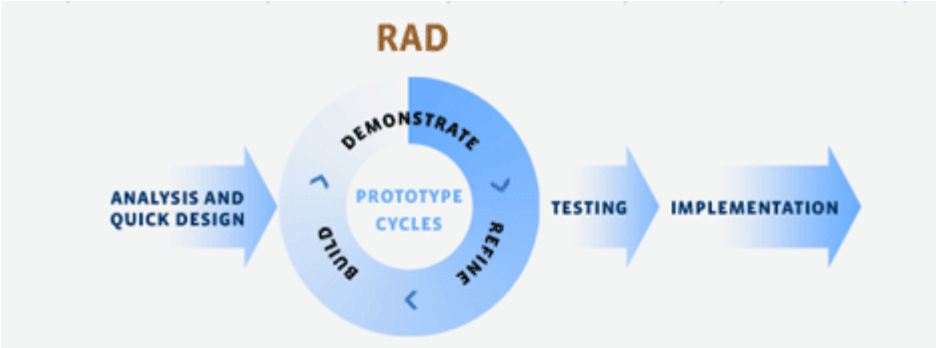


Figure 1.2: The process of Rapid Application Development

Rapid Application Development consists of several stages. First, the requirement for the project is defined and listed out to avoid miscommunication between the stakeholders. Then, the prototype is built, demonstrated and refined to ensure the needed is met. This process will be repeated until they reach a satisfactory and fully functional product. After the finalized prototype is converted into a working model and tested, it is ready for deployment.

1.7 Scope of the Project

This project is aiming to implement a web system and a mobile application. The web application is used by management and admin to manage condo and communicate with tenants and owners, while the mobile application will allow tenants to communicate with condo management.

1.7.1 Login

The project can handle the login of the users as there are a lot of tenants in a condo. The management also need is log in before using the management system.

1.7.2 Create and delete account

Tenants and owners cannot register an account themselves, only the condo management or admin can create an account for them. If the unit is sold, the previous owners account will be terminated and the new owner is needed to request management to get the user ID and temporary password.

1.7.3 Work Order

Tenants are allowed to upload a request for any maintenance work order which is under the jurisdiction of the condo management. The request can attach some descriptions and a few images. The management is allowed to view all work orders that requested by owners and tenants. They also can upload the latest progress of each work order. Hence, the user also can view the latest progress of the work orders.

1.7.4 Announcement

The management team can upload a new announcement. Tenants can view all announcements from the management team.

1.7.5 Bill

This system can generate an invoice and receipt that allow tenants to view and save. Besides, tenants and owners also can pay the bill via this system.

1.7.6 Facilities

Tenants can view any available facilities and book it. The admin can view all facilities is booked by which tenants.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The documentation management system, existing system, and software development methodologies will be discussed by reviewing published information in this chapter.

2.2 Review on Documentation Management System

Documentation management system (DMS) is a process of handling documents that the records and files are stored, managed and organized. There are 2 classes of DMS: traditional documentation management system and electronic documentation management system.

2.2.1 Traditional Documentation Management System

Before Information Technology came into widespread use, most of the governments and companies use the traditional way to store and manage the hard copy documents. This system is simple to implement and use, but it has a lot of defects. It cannot handle efficiently huge documents, because the system needs more time to search or redo old files. Besides, it might cause the lost data and unnecessary duplicate copy of data as everything is processed manually (IncludeHelp, n.t.).

2.2.2 Electronic Documentation Management System

Electronic documentation management system is more "modern" in comparison with the Paper-based documentation management system. All documents and records are stored in digital format. Since, the management system is processed without any paper or hard copy, the organization no need to spend a lot of money on stationery and storage space. According to Johnston and Bowen (2005), the management system makes information easier to find out information when required. On the other hand, the quality of processes and the outcomes will be improved while the employees can complete the work requires less time and effort.

2.2.3 Evaluation

Therefore, the traditional documentation management system is less complex and easy to use by anyone. However, the electronic documentation management system has more benefit to individual users and the organization. The organization can provide training to employees to learn and use the electronic system, so the employees can manage and handle more records and documents in more in an efficient way.

2.3 Review on Existing System

Few similar mobile applications will be studied in this section.

2.3.1 i-Neighbour

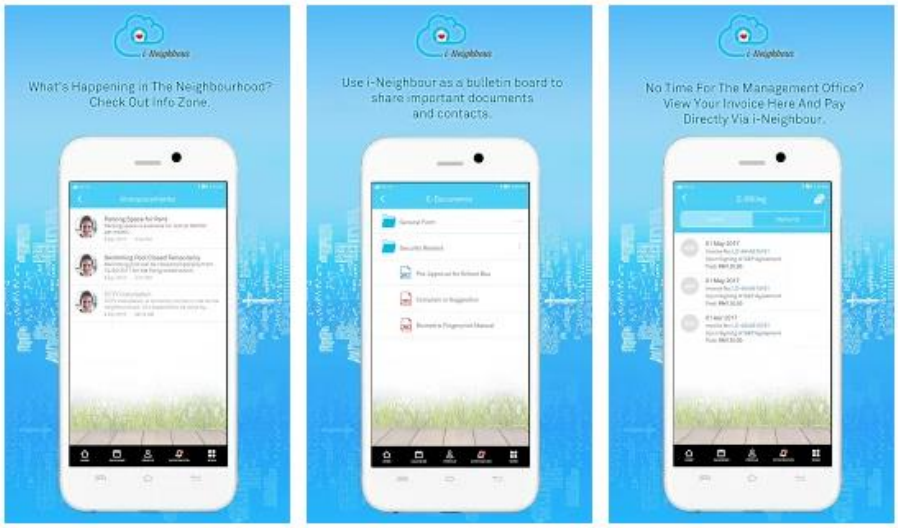


Figure 2.1: i-Neighbour Mobile Appilcation Screenshot

i-Neighbout is developed by TimeTec to provide residents a better experience to communicate with the management team. The management can share the announcements, documents, and contacts to the residential. The app also allows the user to report and view the latest status of the report. The users can pay the payment and view the invoices via the mobile app. Besides, I-Neighbour also provides some methods for the visitor management system.

2.3.2 EcoWorld Community

Figure 2.2: EcoWorld Community Mobile Appilication Screenshot

Malaysia's famous property development company, EcoWorld contract with Leaf Software Solutions to deliver the mobile application EcoWorld Community. The EcoWorld customers can get the latest news about community events and maintenance matters via the application. Besides, the mobile application allows users to book instantly the facilities and services. The users also can befriend and communication with neighbours through the social networking of the EcoWorld Community. The residents also can enjoy the IoT Security Serve via the app and certain hardware.

2.3.3 Setia Community

Figure 2.3: Setia Community Mobile Appilication Screenshot

Similarly to the EcoWorld Community, the Setia Community is developed by Leaf Software Solutions. Hence, they have almost the same features such as to receive news and updates about community events or maintenance, communicate with neighbours, book facilities and pay the bill.

2.3.4 Evaluation

Modules	i-Neighbour	EcoWorld Community	Setia Community
Login	✔	✔	✔
Announcement	✔	✔	✔
Work Order	✔	✔	✔
Book Facilities	✔	✔	✔
Payment	✔	✔	✔
Visitor management	✔	✗	✗
Social network	✔	✔	✔
Security	✔	✗	✗

Table 2.1: Comparison between existing systems.

In conclusion, the fundamental functions of a condo management system are receive announcements, create the report, view the progress, book facilities and pay the bill. However, the features of the system also can be extended when the management requires to add. There are some interesting functions must execute with the specific hardware devices such as QR code reader for visitor reservation and alert alarm for security.

2.4 Review on Software Development Methodologies

Software Delivery Life Cycle (SDLC) is a way to develop, alter, and maintain a software project. SDLC is very important as an organization can deploy faster, satisfy the stakeholders and release high quality software when they make the right choice with the methodology.

2.4.1 Waterfall Methodology

According to Lotz (2018), the waterfall is a traditional development approach and divided by several stages included gather and define the requirements, design, implementation, testing and deliver a product. In a Waterfall development project, the next stage will not begin before the previous stage is completed and any completed stage will not repeat. Hence, it does not have the way to handle when new requirements are added in traditional development methodology (Soni and Kohli, 2017).

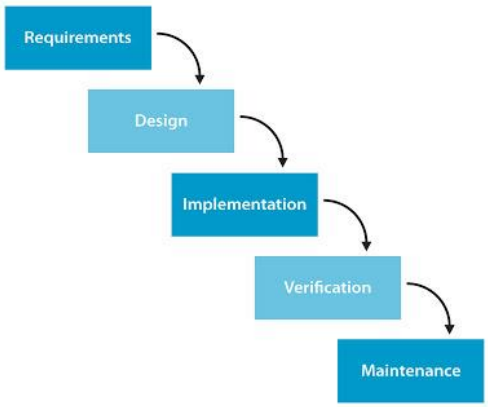


Figure 2.4: The process of Waterfall Methodology

2.4.1.1 Pros

- It is easy to understand and follow as it is a linear model.
- It is easy to manage and control as each stage will be reviewed.
- Each stage will not be overlapping as the next stage will not happen before the current stage is completed.

2.4.1.2 Cons

- It takes a long time to gather the requirement as the requirement must be stated accurately.
- It is not flexible as it is hard to make changes.

2.4.2 Agile

Agile methodology allows developers to develop a system quickly. It also helps the software team to handle rapidly when the client want to change the requirement. There are a lot of different methodologies developed by referring to the principle and concept of Agile.

Extreme Programming (XP) is a common Agile methodologies. According to Shaydulin and Sybrandt (2017), it is built around customers interacting strictly with developers throughout the project. Since the clients and development team work together around the project, clients can see the progress and modify the requirement. It is separated into several short cycles and each cycle consists of planning, designing, coding, testing, and listening.

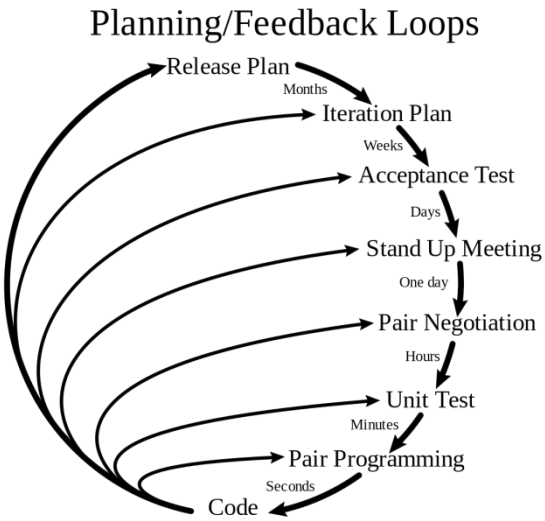


Figure 2.5: The process of Extreme programming

2.4.2.1 Pros

- It leads the team in the right direction as it offers constant feedback from clients.
- It is visible as it provides an open communication to helps each member to follow up the progress.
- It saves the cost as it helps in trimming unproductive events.

2.4.2.2 Cons

- The design of the end product may not satisfy customers as this methodology is focused on coding rather than design.
- It does not measure the quality assurance of coding.

2.4.3 Rapid Application Development

In 1991, James Martin introduced the Rapid Application Development initially, which can emphasize user to involve in every step of the design process (Shaydulin and Sybrandt, 2017). According to Geambasu et al. (2011), some project teams choose RAD as the way to develop a project because it combines the elements from traditional(such as waterfall) and agile methodology. RAD also can adapt new requirements when the process is running because RAD is based on prototype designing before improving the code quality. RAD consists of several stages: gather requirement, build and refine the prototype, then test the prototype. When the requirement is met and the client is satisfied, the process will stop repeating and implement the real product.

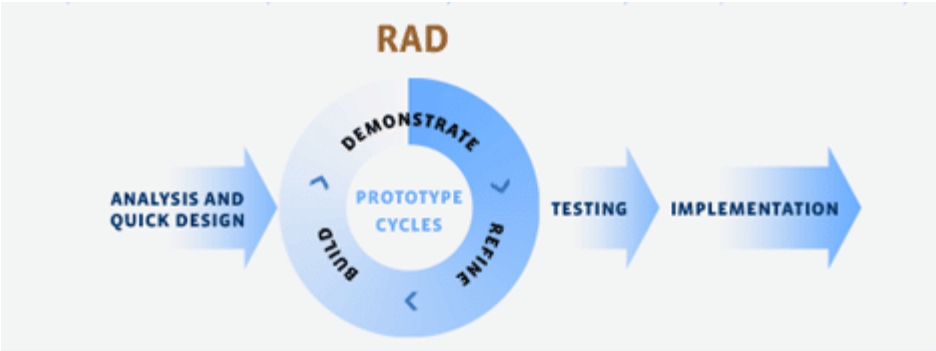


Figure 2.6: The process of Extreme programming

2.4.3.1 Pros

- It is adaptable to make changes as the team will validate and improve the requirements based on user feedback.
- It can control the risk of failure as it ensures to incorporate user feedback in the early stage.
- It has better integration as the integration will build throughout the process.

2.4.3.2 Cons

- It needs high skill developers as it is a high dependency on modeling skills.
- It cannot be completed without the commitment of developers and client

2.4.4 Evaluation

A comparison matrix can be shown the differences of each methodology based on the studies and analysis.

Criteria	Waterfall	Extreme programming (XP)	Rapid Application Development (RAD)
Cost estimation	Yes	Yes	Yes
Well defined requirement	Yes	Yes	Yes
Requirement flexibility	No	Yes	Yes
Quick validation	No	Yes	Yes
Elasticity	No	Medium	Yes
Focus on client	No	Yes	Yes
Cost	Low	High	High

Table 2.2: Comparison between software development methodologies

In a nutshell, RAD will be the most suitable approach for the software methodology. Although the cost of XP and RAD is greater than waterfall, XP and RAD have more advantages rather than the waterfall. RAD and XP are nearly the same, both of them allow customers to add or change requirements. However, the design of the final product in XP may not satisfy the clients as XP are very focus on coding rather than design. Besides, the management system consists of many components, while RAD provides a better way for integration.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

The chosen methodology will be discussed in detail. Besides, the future work plan of the project is proposed.

3.2 Methodology

The Rapid Application Development is chosen as the way to develop the project after comparing 3 types of software development methodologies in section 2.4 (Review on Software Development Methodologies).

The Rapid Application Development consists of several stages:

Stages		Description
Analysis and quick design		The quick prototype is created after the requirement is gathered, defined and listed out.
Prototype Cycle	Build	The prototype is built based on the requirements listed out in the first stages.
	Demonstrate	Gain feedback from stakeholders after showing the design and flow of the prototype.
	Refine	The prototype is improved and refined after evaluating the feedback.
Testing		When the stakeholders are satisfying the product, the product will be tested.
Implementation		After the evaluation, the product will be implemented and ready for the deployment

Table 3.1: Description on each stage of Rapid Application Development

3.3 Project Plan

The Gantt chart and the Work Breakdown Structure and will be shown in this section.

3.3.1 Work Breakdown Structure

The smaller systems and sub-deliverables can be defined by subdividing each stage of the project.

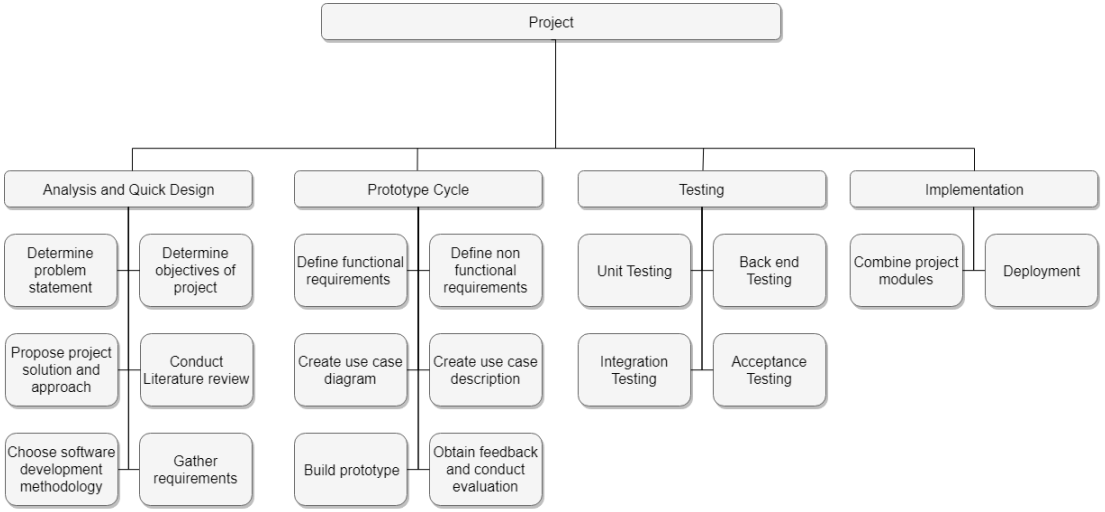


Figure 3.1: Work Breakdown Structure

3.3.2 Gantt Chart

The project schedules are illustrated by drawing the Gantt chart.

	Analysis and quick design	Teh Keh Boon	-	03/Jun	14/Jul	100%
1	Determine problem statement	Teh Keh Boon	-	03/Jun	05/Jun	100%
2	Propose project solution and approa...	Teh Keh Boon	-	06/Jun	08/Jun	100%
3	Determine objectives of project	Teh Keh Boon	-	09/Jun	10/Jun	100%
4	Conduct Literature review	Teh Keh Boon	-	11/Jun	07/Jul	100%
5	Choose software development meth...	Teh Keh Boon	-	08/Jul	10/Jul	100%
6	Gather requirements	Teh Keh Boon	-	11/Jul	14/Jul	100%
	Prototype Cycle	Teh Keh Boon	-	15/Jul	13/Mar	84%
8	Define functional requirements	Teh Keh Boon	-	15/Jul	21/Jul	100%
9	Define non functional requirements	Teh Keh Boon	-	15/Jul	21/Jul	100%
10	Create use case diagram	Teh Keh Boon	-	22/Jul	28/Jul	100%
11	Create use case description	Teh Keh Boon	-	29/Jul	01/Aug	100%
	Build Prototyoe	Teh Keh Boon	-	02/Aug	13/Mar	20%
13	Work Order Module	Teh Keh Boon	-	02/Aug	25/Aug	100%
14	Login Module	Teh Keh Boon	-	13/Jan	19/Jan	0%
15	Announcement Module	Teh Keh Boon	-	20/Jan	01/Feb	0%
16	Booking Module	Teh Keh Boon	-	02/Feb	28/Feb	0%
17	Bill Module	Teh Keh Boon	-	29/Feb	13/Mar	0%
	Testing	Teh Keh Boon	-	14/Mar	01/Apr	0%
19	Unit testing	Teh Keh Boon	-	14/Mar	22/Mar	0%
20	Back end testing	Teh Keh Boon	-	21/Mar	25/Mar	0%
21	Integration testing	Teh Keh Boon	-	26/Mar	29/Mar	0%
22	Acceptance testing	Teh Keh Boon	-	30/Mar	01/Apr	0%
	Implementation	Teh Keh Boon	-	02/Apr	06/Apr	0%
24	Combine project modules	Teh Keh Boon	-	02/Apr	04/Apr	0%
25	Deployment	Teh Keh Boon	-	05/Apr	06/Apr	0%

Figure 3.2: Tasks of Gantt Chart

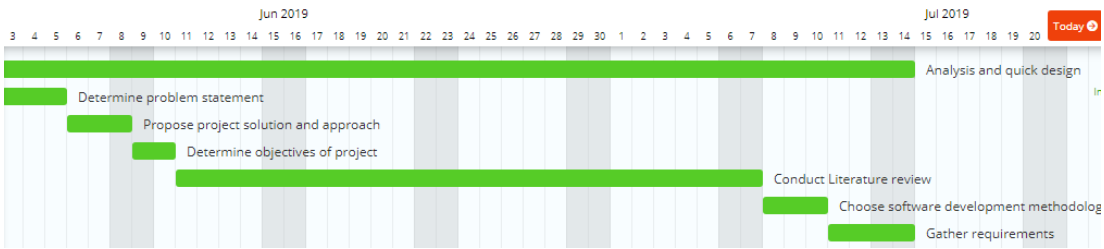


Figure 3.3: Gantt Chart For Analysis and Quick Design Stages

The problem statement, objective solution and approach of the project will be determined in the first stage. Detail understanding and knowledge on a certain topic area will be obtained by conducting the literature review. Then, software development methodology should be selected after comparison between different type of methodology. At the end of this stage, the requirements should be gathered and defined.

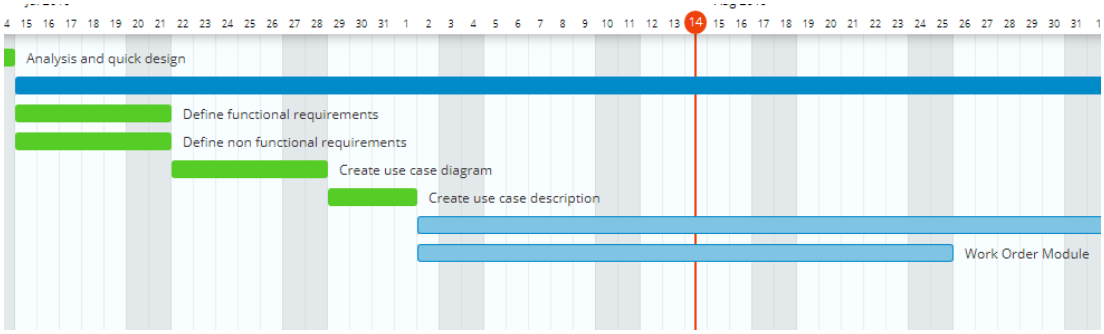


Figure 3.4: Gantt Chart For Prototype Cycle Stages I

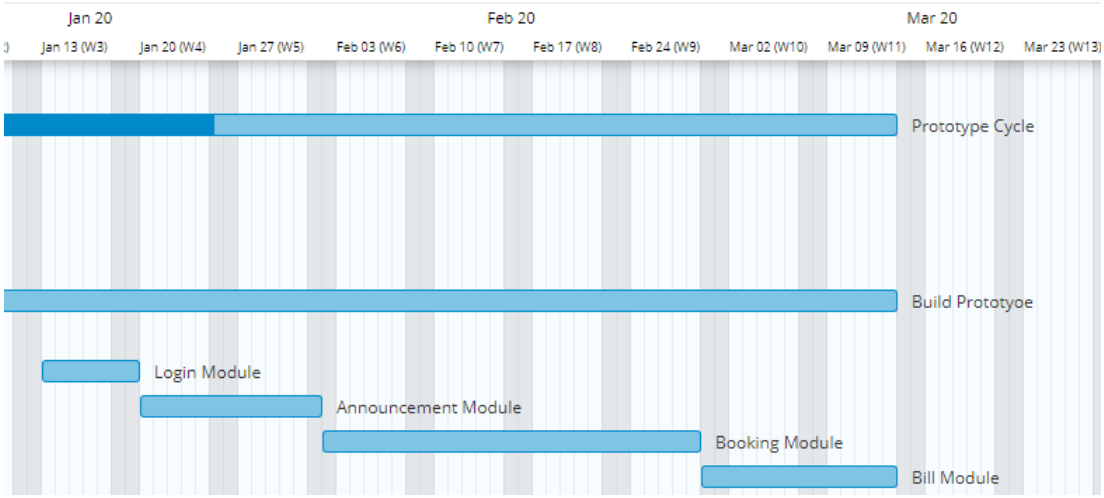


Figure 3.5: Gantt Chart For Prototype Cycle Stages II

In this stage, the functional and non functional requirement will be defined based on the research in the first stage. Then, the use case is needed to produced to have a better understanding about the design and flow of the system. The prototype will be built in several modules.

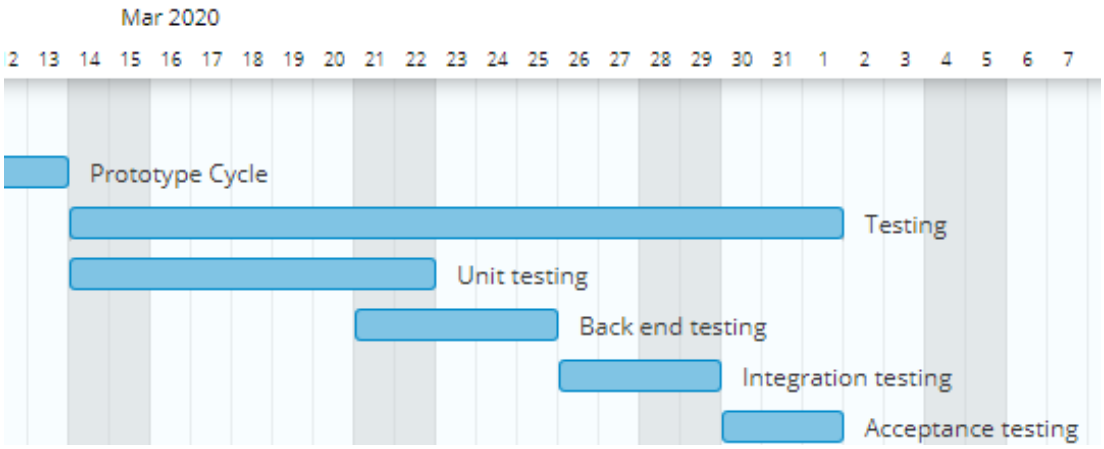


Figure 3.6: Gantt Chart For Testing Stage

After stakeholders are satisfied, the system will be tested in different ways. For example, the integration testing and acceptance testing will be conducted after complete the back end testing and unit testing.

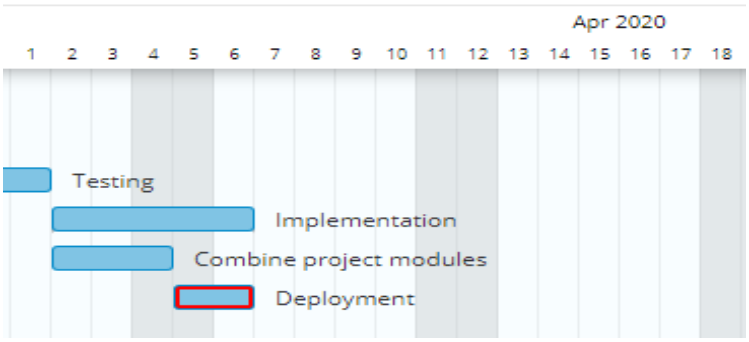


Figure 3.7: Gantt Chart For Implementation Stage

When the error and bug are reduced as possible, each project module will combine into a whole system.

CHAPTER 4

PROJECT INITIAL SPECIFICATION

4.1 Introduction

In this chapter, fact finding is done by interviewing a member of the condo management team. The UML also provided to display the detail and flow of the product. Then, the functional requirements and the non-requirements will be discussed.

4.2 Fact Finding

The interview has been conducted with a member of a member of the condo management team in Evergreen Park Scot Pine. The interview questions and the answers can be referred in appendix A. The summary and analysis of the interview will be shown below:

4.2.1 Work Order

The management team creates a WhatsApp group among the tenants and management team. The tenants are allowed to request the work order in the group. However, the information is not delivered formally so tenants might not easy to follow up the work order. Hence, they might need a system to list up the progress and allow the management team to update the status. The tenants also can search for certain work order easily.

4.2.2 Announcement

The management team will only post the announcement on the board. Although the board placed in the lifts, the announcement also might be ignored. The tenants also cannot receive any announcement instantly. This problem can be solved easily when the tenants can get any announcement via the mobile application.

4.2.3 Bill

Instead of paying in cash, tenants can online pay the bill. However, the tenants still need to go to the management department to show the invoice. There will be a better experience when tenants can pay the bill via mobile application. The system also will determine which tenant has paid the payment.

4.3 Use Case

The design and workflow of project system can be shown by drawing the use case diagram. The use case description will be discussed to show the detail of each activity.

4.3.1 Use Case Diagram

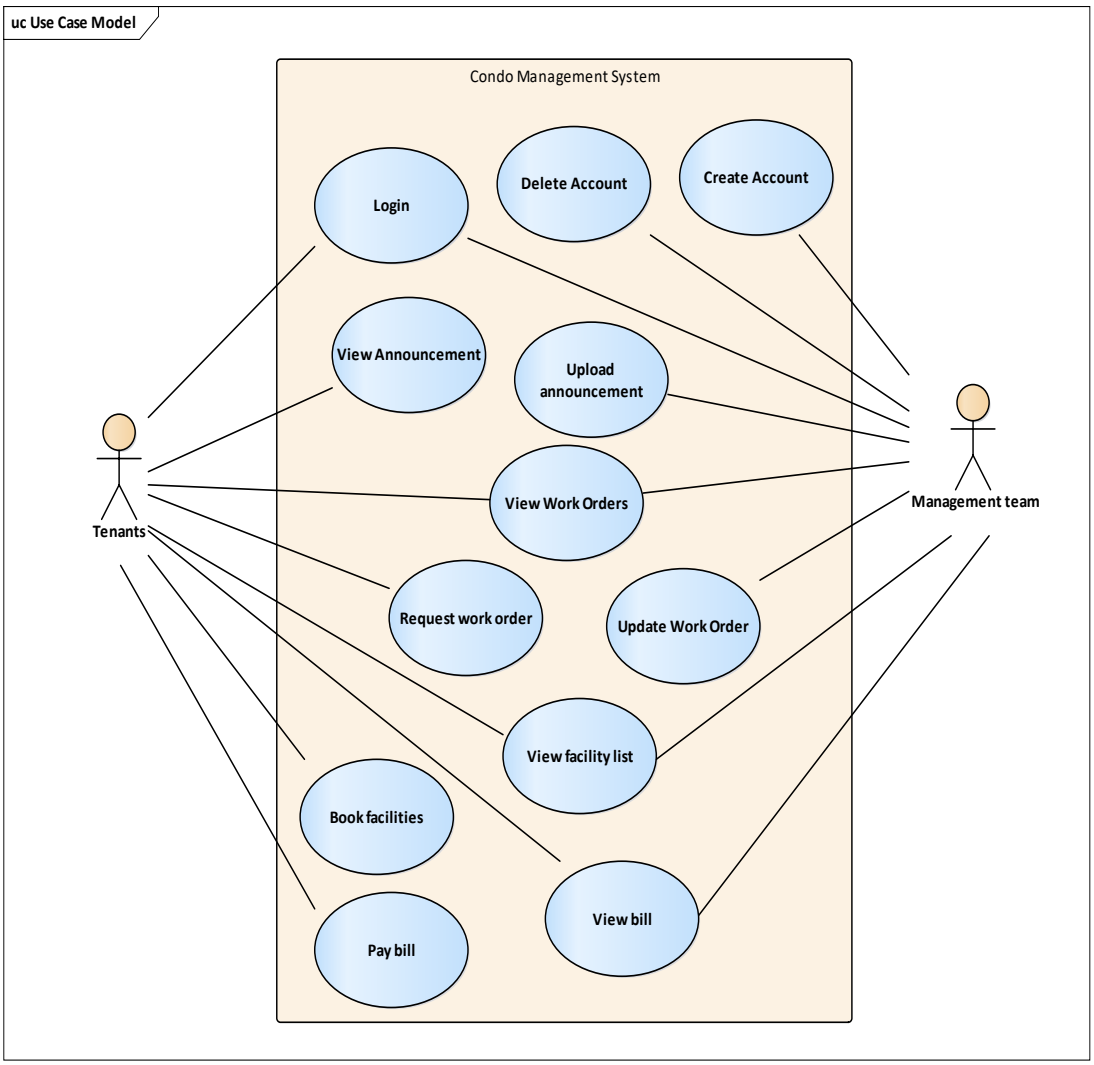


Figure 4.1: Use Case Diagram

4.3.2 Use Case Description

Use case ID	1
Use case Name	Login
Actors	Tenants and Management Team
Description	The tenants and management team log in to account
Flow of events: 1. The tenants or management team log in by entering the ID and password. 2. System displays a message indicating the login is successful.	
Alternative flow of events: 2.1 Account with entered ID and password are not found. 2.2 System displays a message indicating the incorrect ID and password is entered. 2.3 Use case terminates.	

Table 4.1: Login Use Case Description

Use case ID	2
Use case Name	Create account
Actors	Management Team
Description	Management team creates account for new tenant.
Flow of events: 1. Login use case had performed. 2. Management team chooses to create an account. 3. Management team enters the detail for new account. 4. System creates a new account and generate a random password. 5. System displays a message indicating the account is created successfully.	
Alternative flow of events:	

Table 4.2: Create Account Use Case Description

Use case ID	3
Use case Name	Delete account
Actors	Management Team
Description	Management team deletes account.
Flow of events: 1. Login use case had performed. 2. Management team chooses to view account list. 3. System displays the account list. 4. Management team select an account. 5. Management team clicks the delete button. 6. System deletes the selected account. 7. System displays a message indicating the account is deleted successfully	
Alternative flow of events:	

Table 4.3: Delete Account Use Case Description

Use case ID	4
Use case Name	Upload announcement
Actors	Management Team
Description	Management team uploads an announcement to notice tenants
Flow of events: 1. Login use case had performed. 2. Management team chooses to upload announcement. 3. Management team enters the title and description of the announcement. 4. System uploads the announcement to the database. 5. System displays a message indicating the announcement is uploaded successfully.	
Alternative flow of events:	

Table 4.4: Upload Announcement Use Case Description

Use case ID	5
Use case Name	View announcement
Actors	Tenants
Description	Tenants view the announcement from the management team.
Flow of events: 1. Login use case had performed. 2. Tenants choose to view the announcements. 3. System displays the announcement list.	
Alternative flow of events: 4. Tenants click into certain announcement. 5. System displays the detail of the announcement.	

Table 4.5: View Announcement Use Case Description

Use case ID	6
Use case Name	View Work Order
Actors	Tenants and Management Team
Description	Tenants and Management team view the work order.
Flow of events: 1. Login use case had performed. 2. Tenants or Management team choose to view work order. 3. System displays the work order list.	
Alternative flow of events: 4. Tenants or Management team click into certain work order. 5. System displays the detail of the work order.	

Table 4.6: View Work Order Use Case Description

Use case ID	7
Use case Name	Request work order
Actors	Tenants
Description	Tenants request a new work order.
Flow of events: 1. Login use case had performed. 2. Tenants choose to request work order. 3. Tenants enter the title, description and category of the work order. 4. System uploads the work order into the database. 5. System displays a message indicating the work order is uploaded successfully.	
Alternative flow of events: 3.1 Tenants attach some images.	

Table 4.7: Request Work Order Use Case Description

Use case ID	8
Use case Name	Update work order
Actors	Management team
Description	Management team updates the progress of the work order
Flow of events: 1. Login use case had performed. 2. View work order use case had performed 3. Management team click into certain work order. 4. Management team updates the progress of the selected work order. 5. System updates the status of the selected work order. 6. System displays a message indicating the work order is updated successfully.	
Alternative flow of events:	

Table 4.8: Update Work Order Use Case Description

Use case ID	9
Use case Name	View Facility list
Actors	Tenants and Management Team
Description	Tenants and Management team view the facility list.
Flow of events: 1. Login use case had performed. 2. Tenants or Management team choose to view facility list. 3. System displays the facility list.	
Alternative flow of events:	

Table 4.9: View Facility Use Case Description

Use case ID	10
Use case Name	Book facility
Actors	Tenants
Description	Tenants book the facility.
Flow of events: 1. Login use case had performed. 2. View facility list use case had performed 3. Tenants click into certain facility, 4. Tenants choose the date and time. 5. Tenants book the facility. 6. System record the reservation into database. 7. System displays a message indicating the facility is booked successfully.	
Alternative flow of events: 4.1 The facility is not available in the selected date and time. 4.2 System displays a message indicating the facility is not available. 4.3 User case terminated.	

Table 4.10: Book Facility Use Case Description

Use case ID	11
Use case Name	View Bill as Tenants
Actors	Tenants
Description	Tenants view their bill.
Flow of events: 1. Login use case had performed. 2. Tenants choose to view the bill. 3. System displays the tenant's bill.	
Alternative flow of events:	

Table 4.11: View Bill as Tenants Use Case Description

Use case ID	12
Use case Name	View Bill as Management team
Actors	Management team
Description	Management views tenants' bills.
Flow of events: 1. Login use case had performed. 2. Management team chooses to view account list. 3. System displays the account list. 4. Management team select an account. 5. System displays the detail of account include the bill.	
Alternative flow of events:	

Table 4.12: View Bill as Management Team Use Case Description

Use case ID	13
Use case Name	Pay bill
Actors	Tenants
Description	Tenants pay their bill.
Flow of events: 1. Login use case had performed. 2. View bill case had performed. 3. Tenants choose to pay the bill. 4. System will record and update the tenants' bill	
Alternative flow of events:	

Table 4.13: Pay Bill Use Case Description

4.4 Project Requirement

After interviewing with management team, the functional requirements and non-functional requirements are determined.

4.4.1 Non Functional Requirements

a. The mobile application shall be executed on multiple mobile platforms such as iOS and Android OS.
b. The mobile application and web application shall provide user-friendly interface.
c. The system shall be maintainable and reliable.

4.4.2 Functional Requirements

Mobile Application:

a) Tenants shall be able to login before using the application.
b) Tenants shall be able to view all announcements from the management team.
c) Tenants shall be able to upload a request for any maintenance work order which is under the jurisdiction of the condo management.
d) Tenants shall be able to view the latest progress of work order.
e) Tenants shall be able to view, save and pay the bill.
f) Tenants shall be able to view and book the available facilities within the condo.

Web Application:

a) The management team shall be able to login before using the web management system.
b) The management team shall be able to create an account for new tenants.
c) The management team shall be able to delete any account.
d) The management team shall be able to upload announcement to all tenants.
e) The management team shall be able to view all work orders that requested by owners and tenants.
f) The management team shall be able to update the latest progress of each work order.
g) The management team shall be able to view all facilities is booked by which tenants.

CHAPTER 5
DESIGN

5.1 Introduction

There will be 4 types of design were discussed in this chapter, namely software architecture design, database design, system design and user interface design.

5.2 Software Architecture Design

The adopted architecture design in this project is 3-tier (or layer) architecture, which frequently used in client-server system. This architecture is chosen as it is able to increase the scalability of system and efficiency of development. The overall software architecture will be shown as figure below.

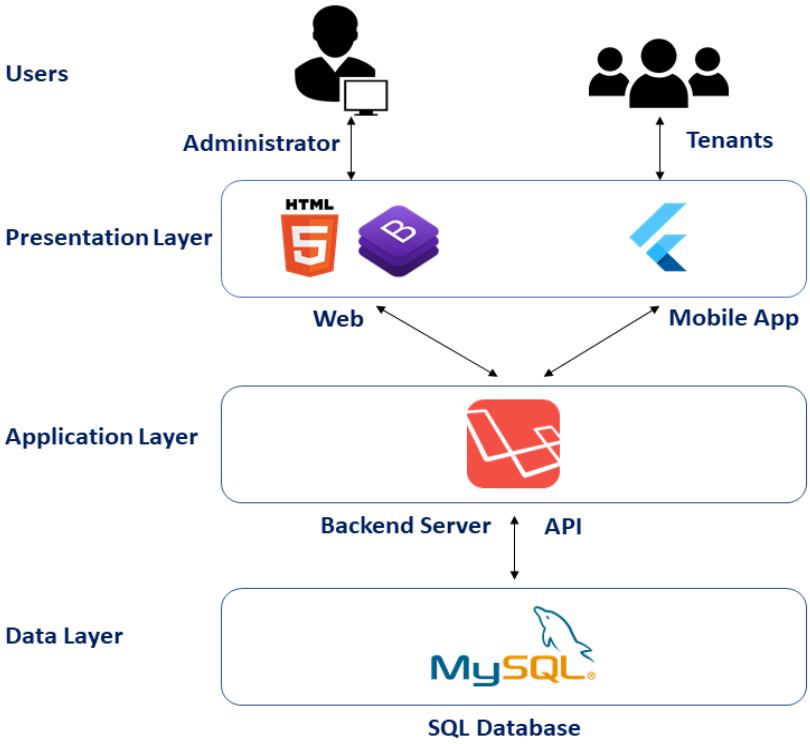


Figure 5.1: Software Architecture Design

5.3 Database design

5.3.1 Entity Relational Diagram (ERD)

Entity Relational Diagram is one of the common ways to illustrate the design of a database by describing the relationship between every entity stored in the database.

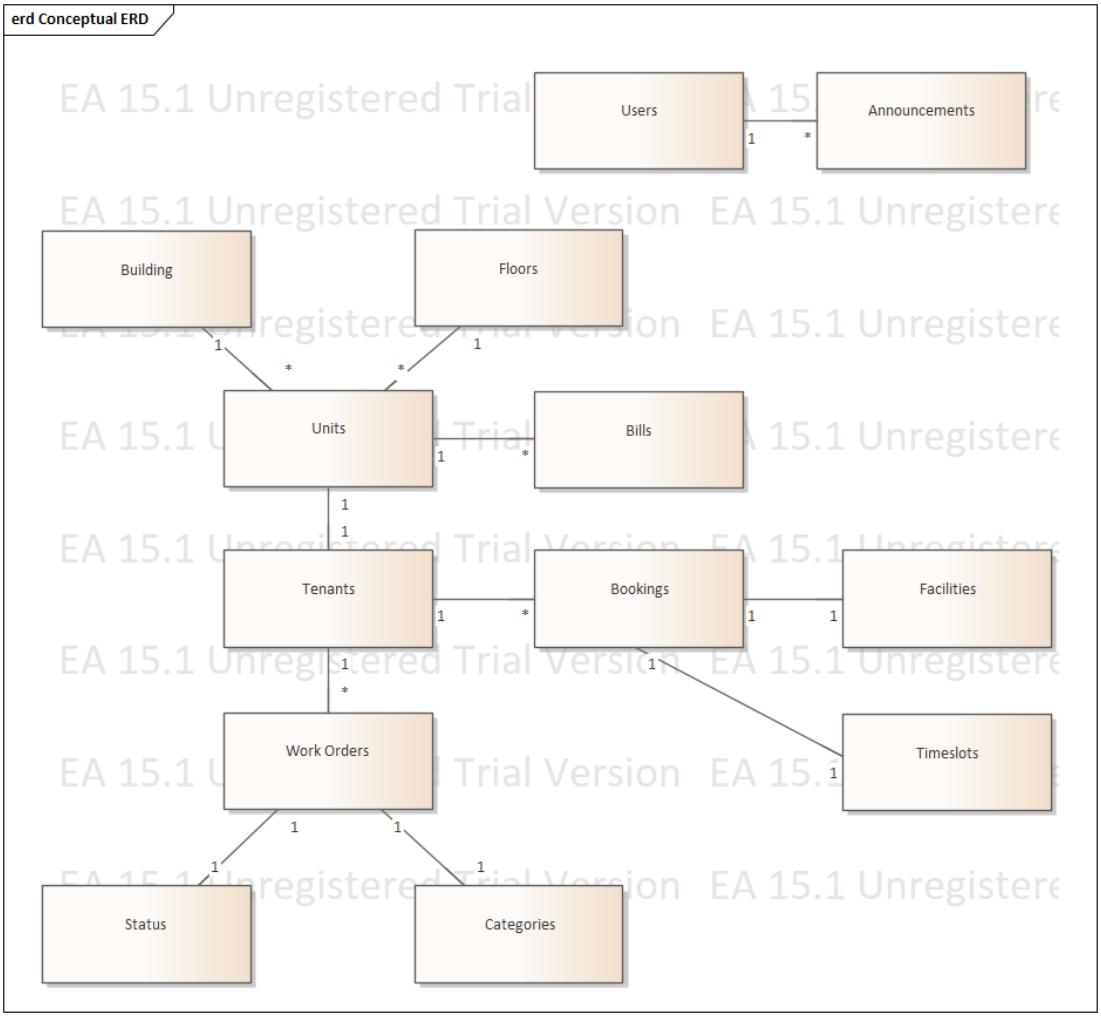


Figure 5.2: Conceptual ERD

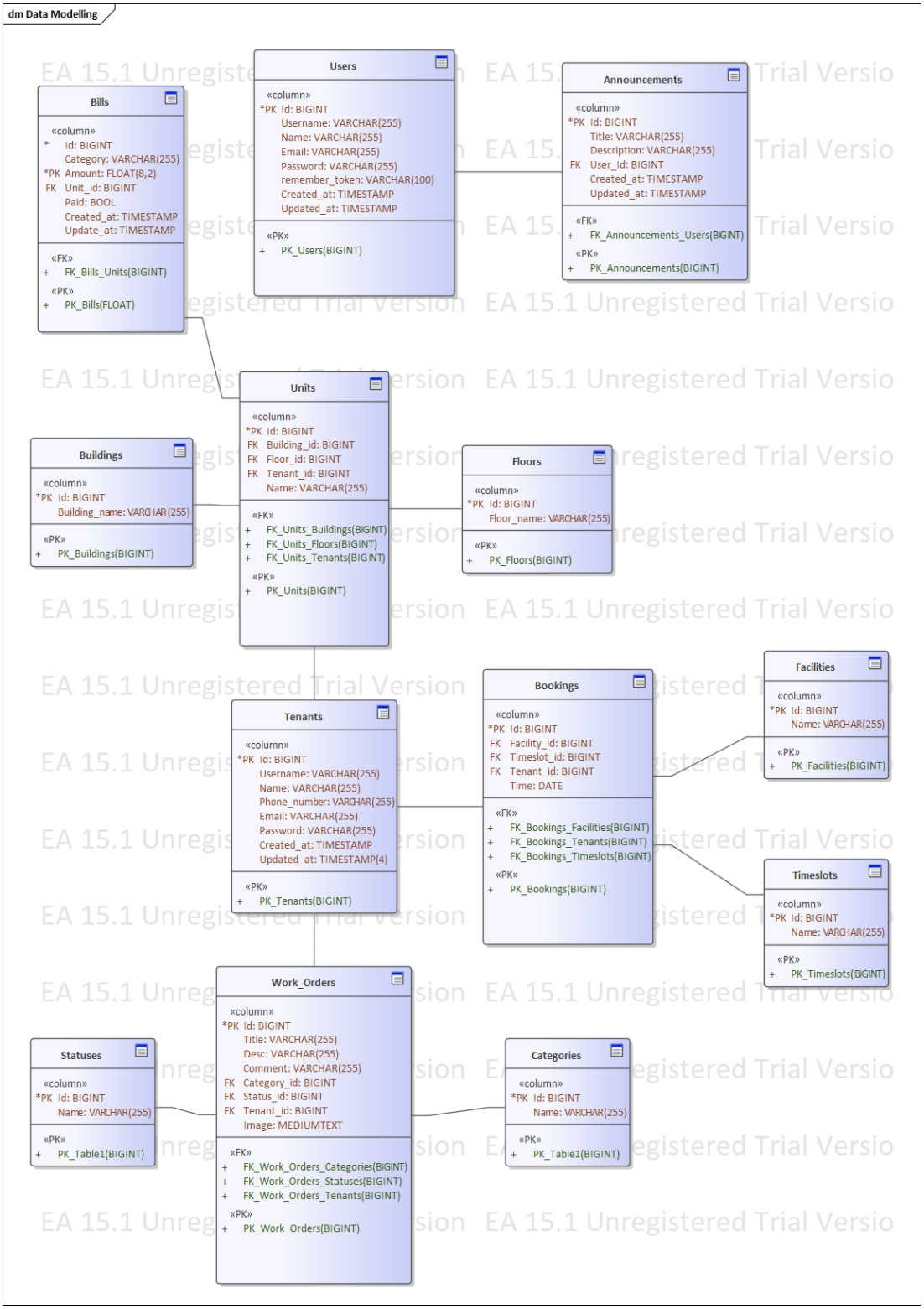


Figure 5.3: Physical ERD

5.3.2 Data Dictionary

There are 13 data dictionaries in this project's database.

Table	Users			
Column	Description	Data Type	Key	Refer to
Id	Admin's unique ID	BigInt	Primary	-
Username	Admin's unique username	Varchar	-	-
Name	Admin's name	Varchar	-	-
Email	Admin's unique email	Varchar	-	-
Password	Admin's password	Varchar	-	-
Remember_token	Remember token of then account	Varchar	-	-
Created_at	Creation date of the account	Timestamp	-	-
Updated_at	Date of last updated	Timestamp	-	-

Table 5.1: Data Dictionary for Users

Table	Tenants			
Column	Description	Data Type	Key	Refer
Id	Tenant's unique ID	BigInt	Primary	-
Username	Tenant's unique username	Varchar		
Name	Tenant's name	Varchar	-	-
Phone_number	Tenant's phone number	Varchar	-	-
Email	Tenant's email	Varchar	-	-
Password	Tenant's password	Varchar	-	-
Created_at	Creation date of the account	Timestamp	-	-
Update_at	Date of last updated	Timestamp	-	-

Table 5.2: Data Dictionary for Tenants

Table	Announcements			
Column	Description	Data Type	Key	Refer to
Id	Unique ID of the announcement	BigInt	Primary	-
Title	Title of the announcement	Varchar	-	-
Description	Description of the announcement	Text	-	-
User_id	Announcer's Unique ID	BigIng	Foreign	Users
Created_at	Release date of the anouncement	Timestamp	-	-
Updated_at	Date of last updated	Timestamp	-	-

Table 5.3: Data Dictionary for Announcements

Table	Buildings			
Column	Description	Data Type	Key	Refer to
Id	Unique ID of the building	BigInt	Primary	-
Name	Name of the building	Varchar	-	-

Table 5.4: Data Dictionary for Buildings

Table	Floors			
Column	Description	Data Type	Key	Refer to
Id	Unique ID of the floor	BigInt	Primary	-
Name	Name of the floor	Varchar	-	-

Table 5.5: Data Dictionary for Floors

Table	Units			
Column	Description	Data Type	Key	Refer
Id	Unique ID of the unit	BigInt	Primary	-
Name	Name of the unit	Varchar	-	-
Building_id	Building of the unit	BigInt	Foreign	Buildings
Floor_id	Floor of the unit	BigInt	Foreign	Floors
Tenant_id	Unique ID of the Owner of the unit	BigInt	Foreign	Tenants

Table 5.6: Data Dictionary for Units

Table	Bills			
Column	Description	Data Type	Key	Refer
Id	Unique ID of the bill	BigInt	Primary	-
Category	Category of the bill	Varchar	-	-
Amount	Bill Amount	Float	-	-
Unit_id	Unit of the bill	BigInt	Foreign	Units
Paid	Status of bill (paid/unpaid)	Bool	-	-
Created_at	Release date of the bill	Timestamp	-	-
Updated_at	Payment date of the bill	Timestamp	-	-

Table 5.7: Data Dictionary for Bills

Table	Statuses			
Column	Description	Data Type	Key	Refer to
Id	Unique ID of the status	BigInt	Primary	-
Name	Name of the status	Varchar	-	-

Table 5.8: Data Dictionary for Statuses

Table	Categories			
Column	Description	Data Type	Key	Refer to
Id	Unique ID of the category	BigInt	Primary	-
Name	Name of the category	Varchar	-	-

Table 5.9:Data Dictionary for Categories

Table	Work Orders			
Column	Description	Data Type	Key	Refer
Id	Unique ID of the work order	BigInt	Primary	-
Title	Title of the work order	Varchar		
Desc	Description of the work order	Varchar	-	-
Comment	Comment of the admin	Varchar	-	-
Category_id	ID of the category of the work order	BigInt	Foreign	Categories
Status_id	ID of the status of the work order.	BigInt	Foreign	Statuses
Tenant_id	Reporter’s ID	BigInt	Foreign	Tenants
Image	Image name of the work order	MEDIUMTEXT	-	-

Table 5.10: Data Dictionary for WorkOrders

Table	Facilities			
Column	Description	Data Type	Key	Refer to
Id	Unique ID of the facility	BigInt	Primary	-
Name	Name of the facility	Varchar	-	-

Table 5.11: Data Dictionary for Facilities

Table	Timeslots			
Column	Description	Data Type	Key	Refer to
Id	Unique ID of the timeslots	BigInt	Primary	-
Name	Detail of the timeslot	Varchar	-	-

Table 5.12: Data Dictionary for Timeslots

Table	Bookings			
Column	Description	Data Type	Key	Refer
Id	Unique ID of the booking order	BigInt	Primary	-
Time	Date of the booking order	Date	-	-
Facility_id	ID of the facility of the booking order	BigInt	Foreign	Facilities
Timeslot_id	ID of the timeslot of the booking order.	BigInt	Foreign	Timeslots
Tenant_id	ID of the tenant who make booking	BigInt	Foreign	Tenants

Table 5.13: Data Dictionary for Bookings

5.4 System Design

5.4.1 Activity Diagram

The behaviour and workflow of system can be illustrated by the activity diagrams.

5.4.1.1 Web System

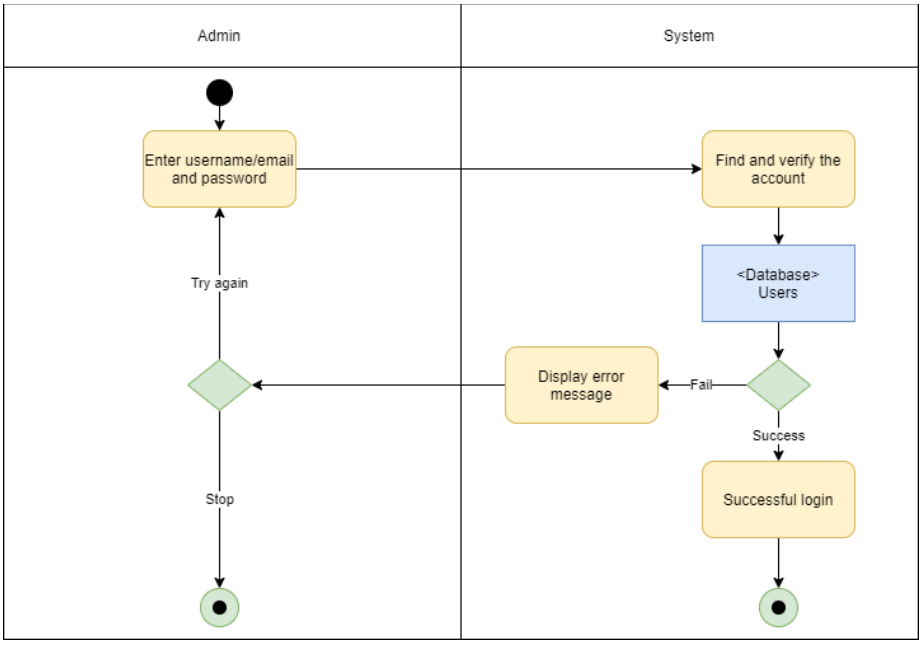


Figure 5.4: Activity diagram for admin(login)

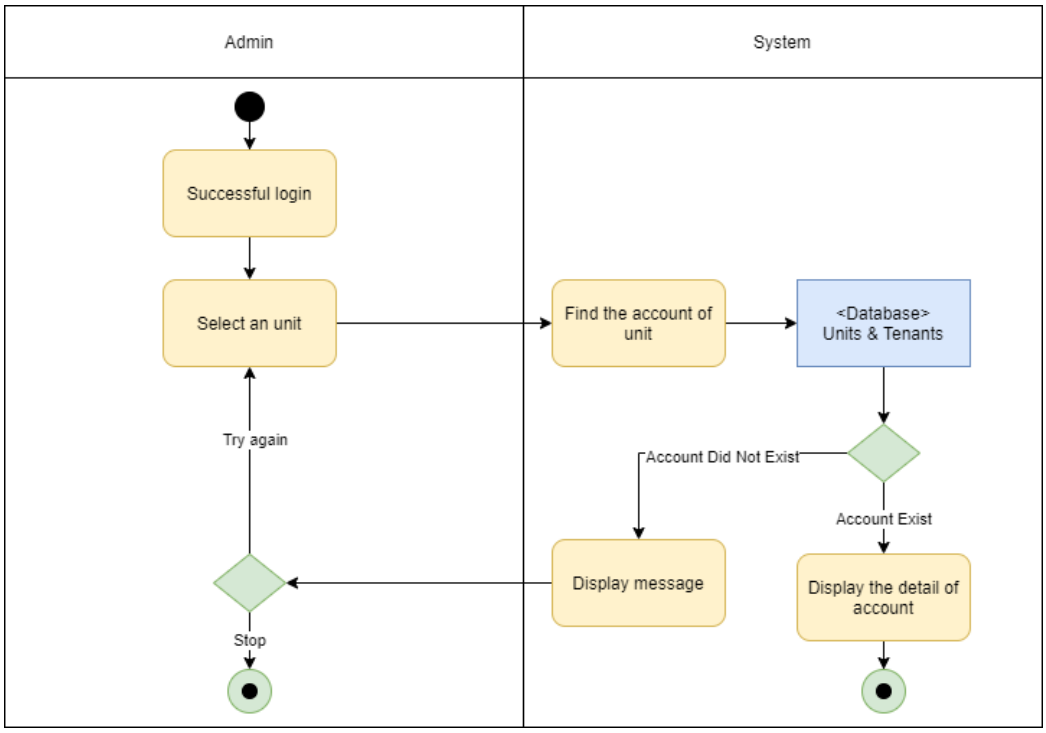


Figure 5.5: Activity diagram for admin(view tenant account)

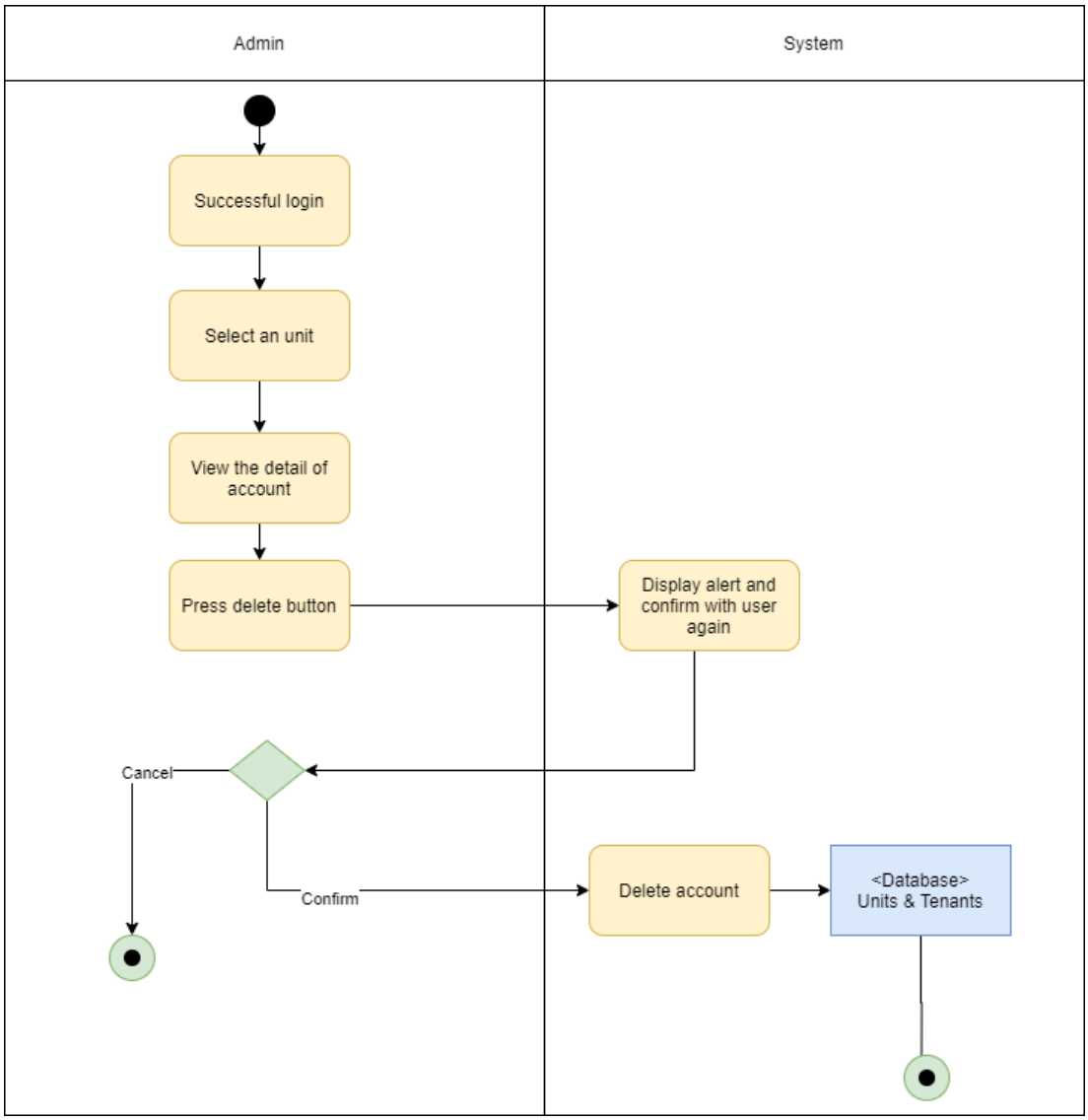


Figure 5.6: Activity diagram for admin(delete tenant account)

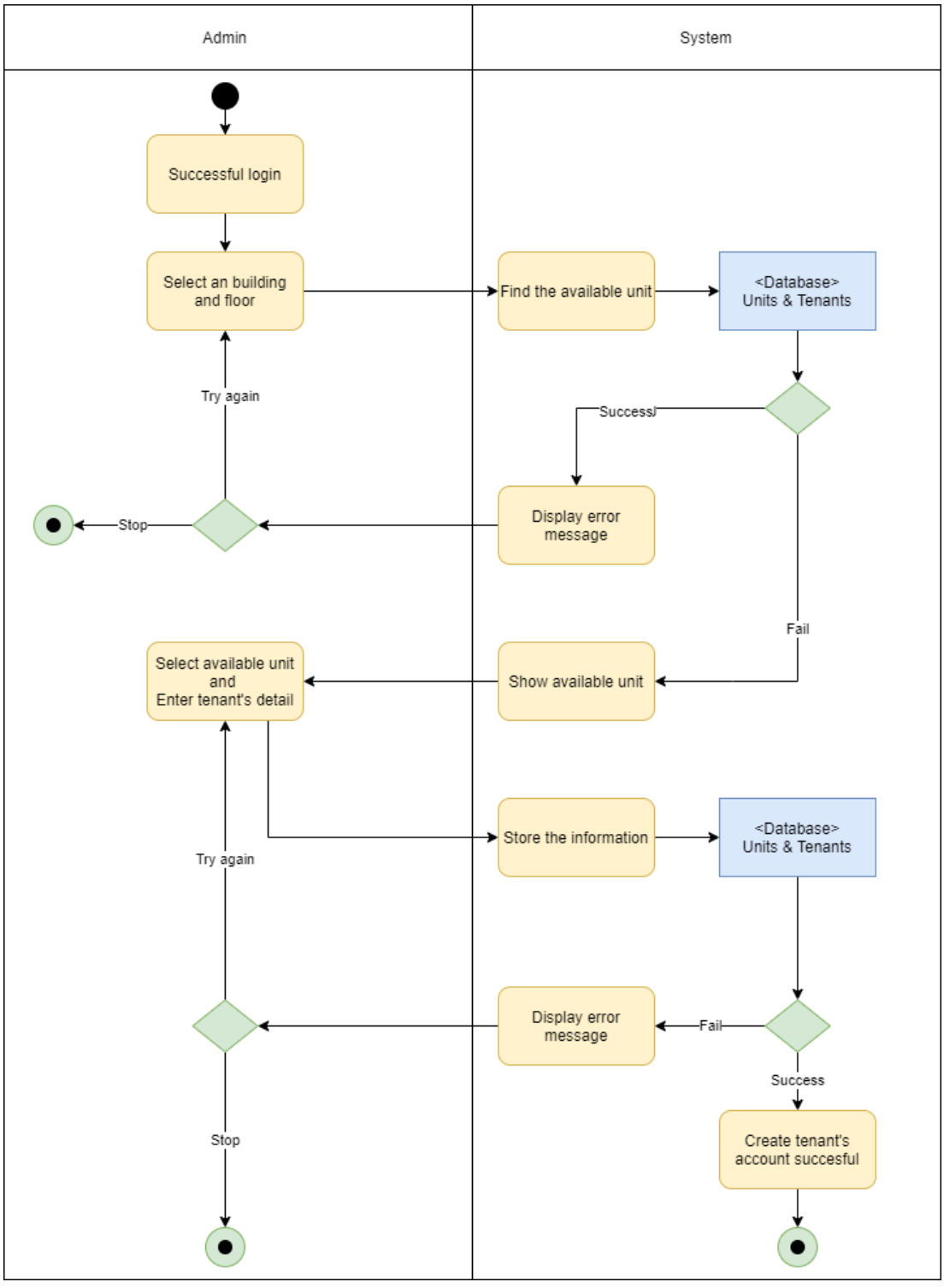


Figure 5.7: Activity diagram for admin(create tenant account)

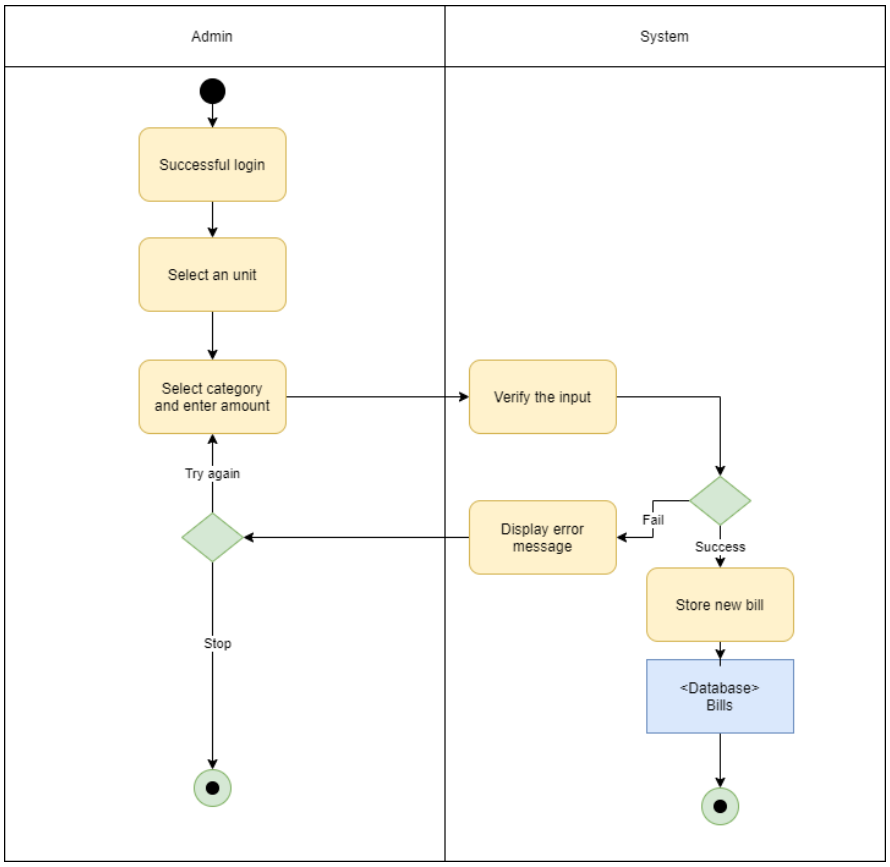


Figure 5.8: Activity diagram for admin(add new bill)

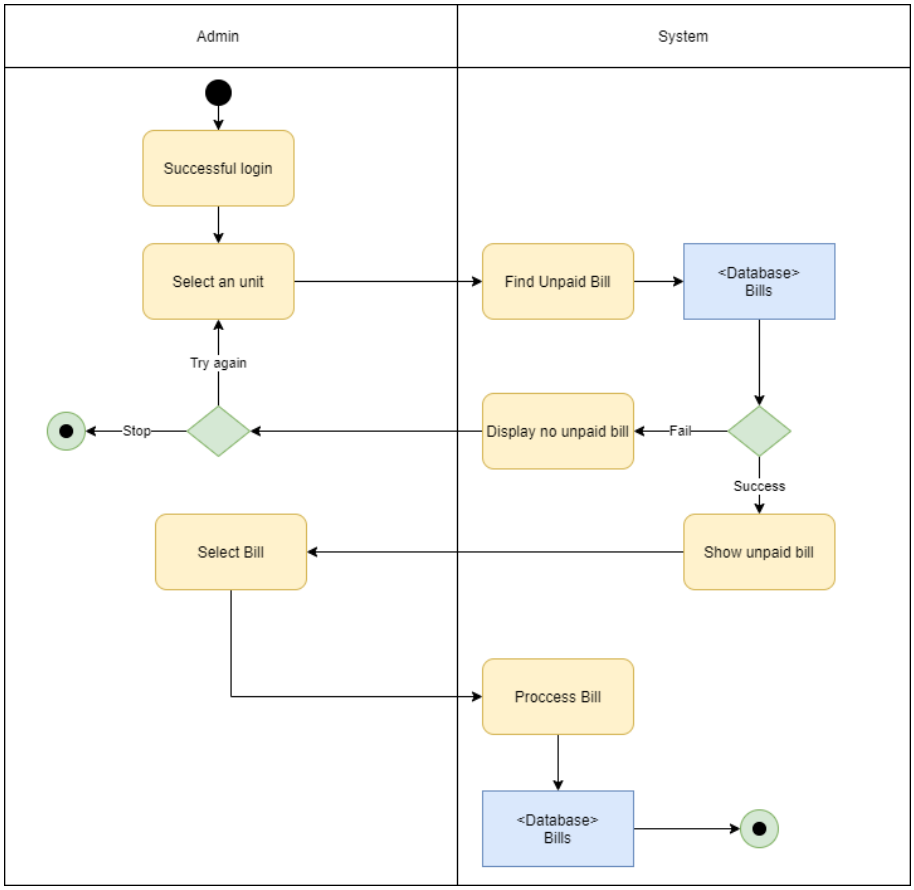


Figure 5.9: Activity diagram for admin(record payment)

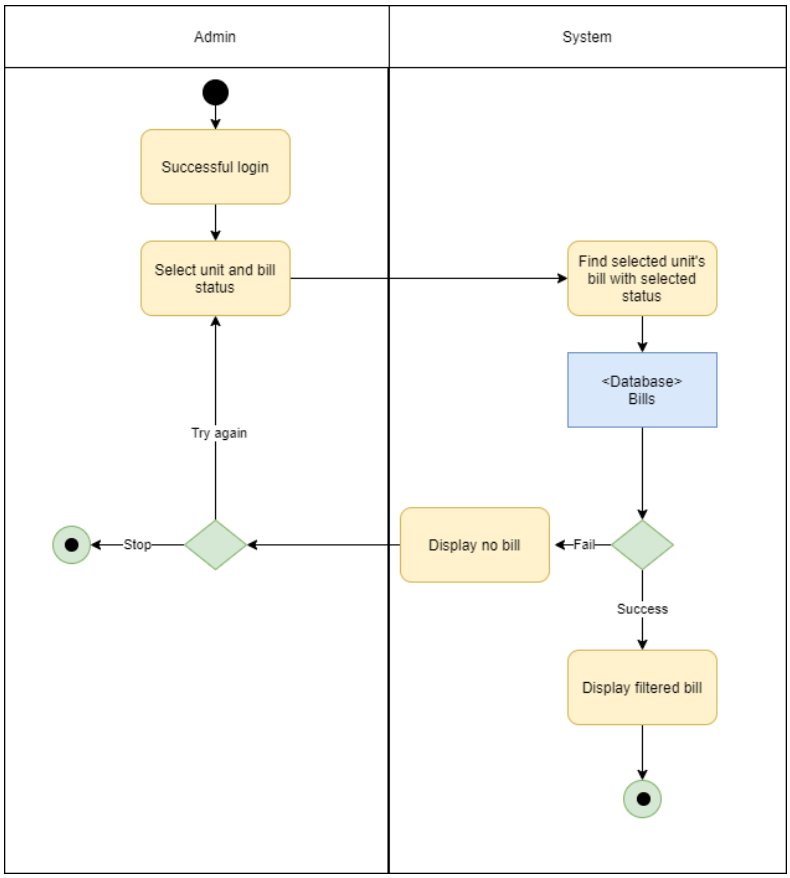


Figure 5.10: Activity diagram for admin(view bill record)

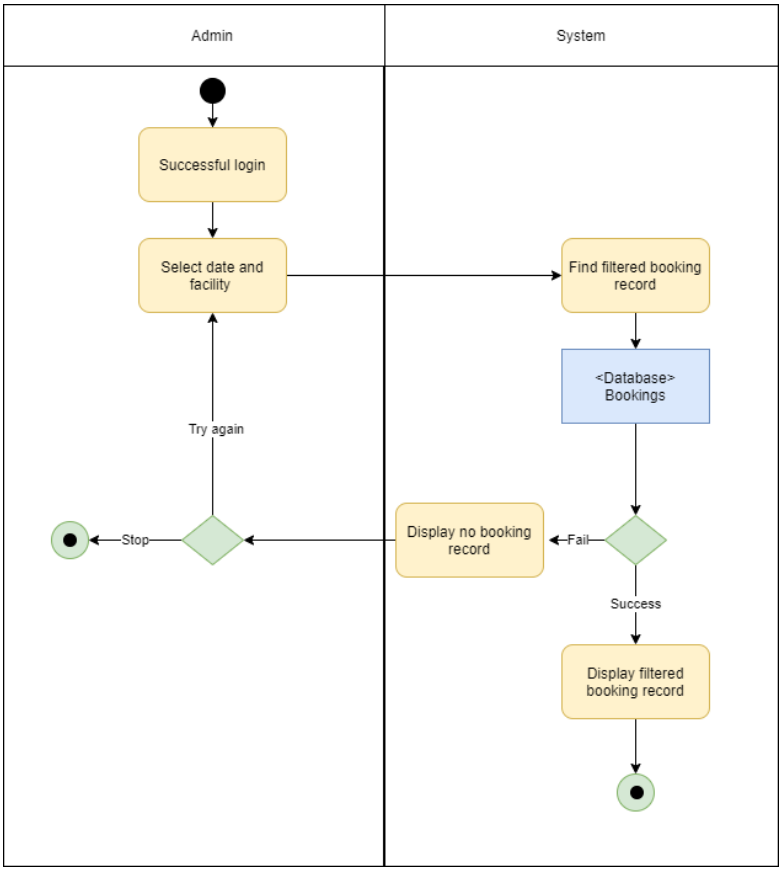


Figure 5.11: Activity diagram for admin(view booking record)

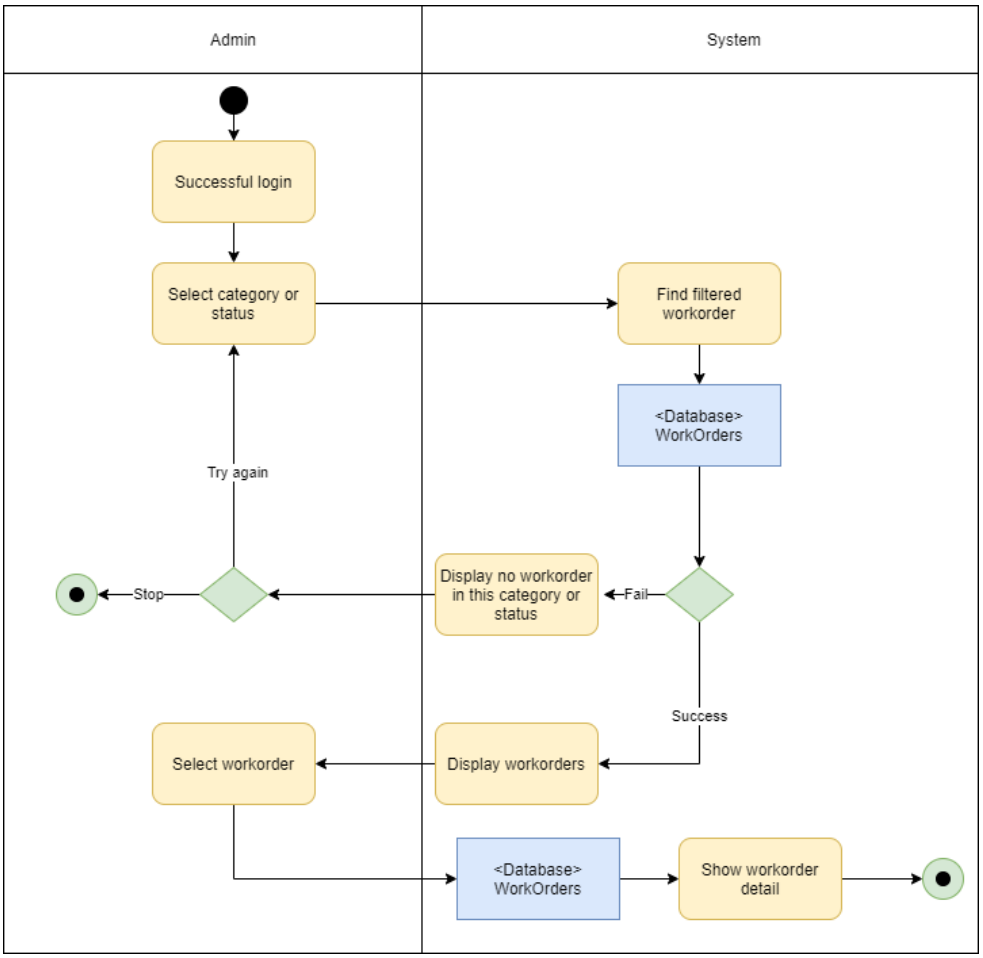


Figure 5.12: Activity diagram for admin(view workorder)

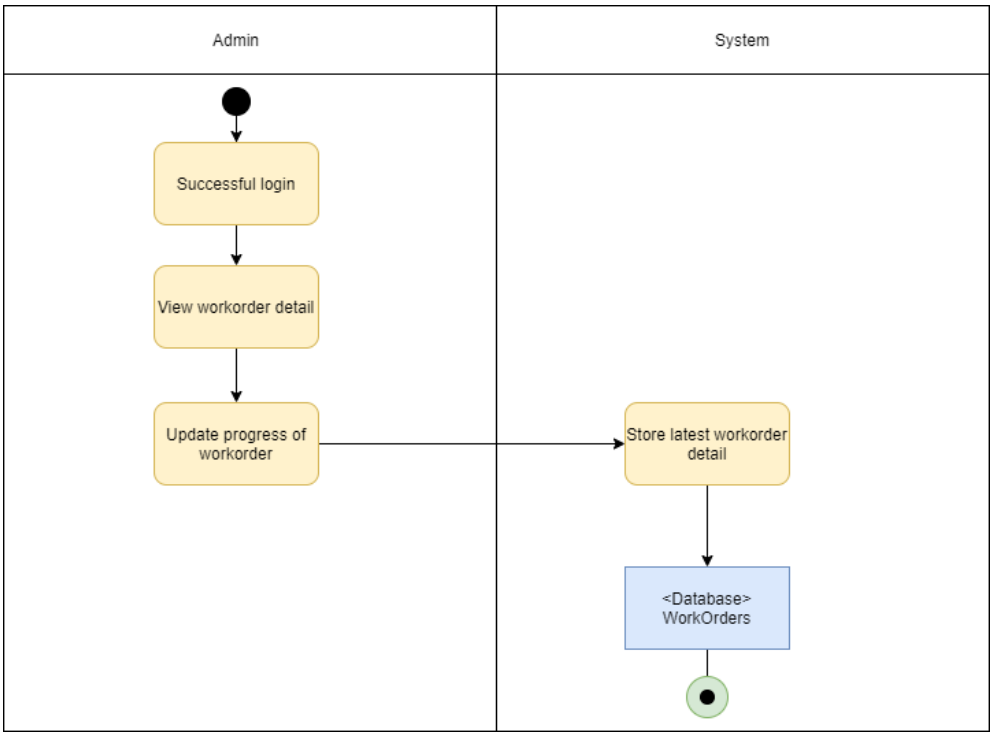


Figure 5.13: Activity diagram for admin(update workorder)

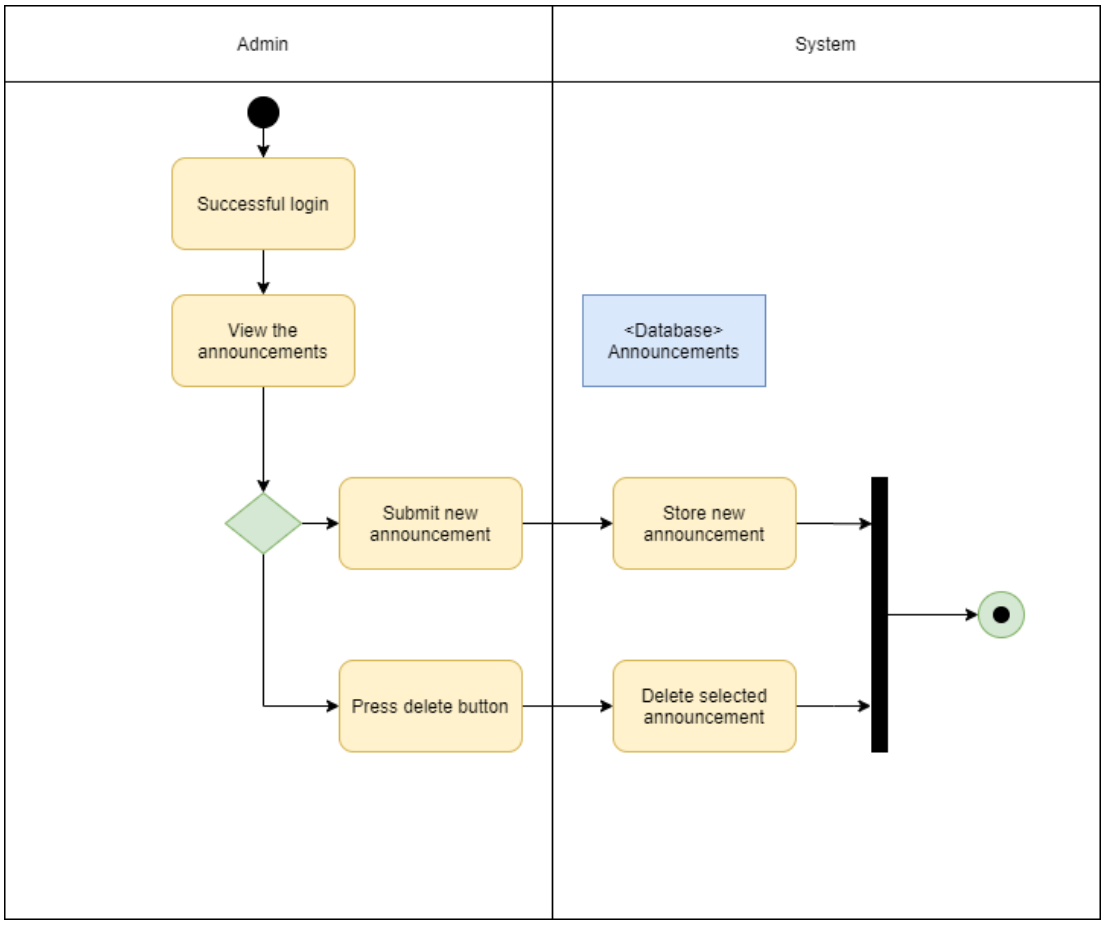


Figure 5.14: Activity diagram for admin(handle announcement)

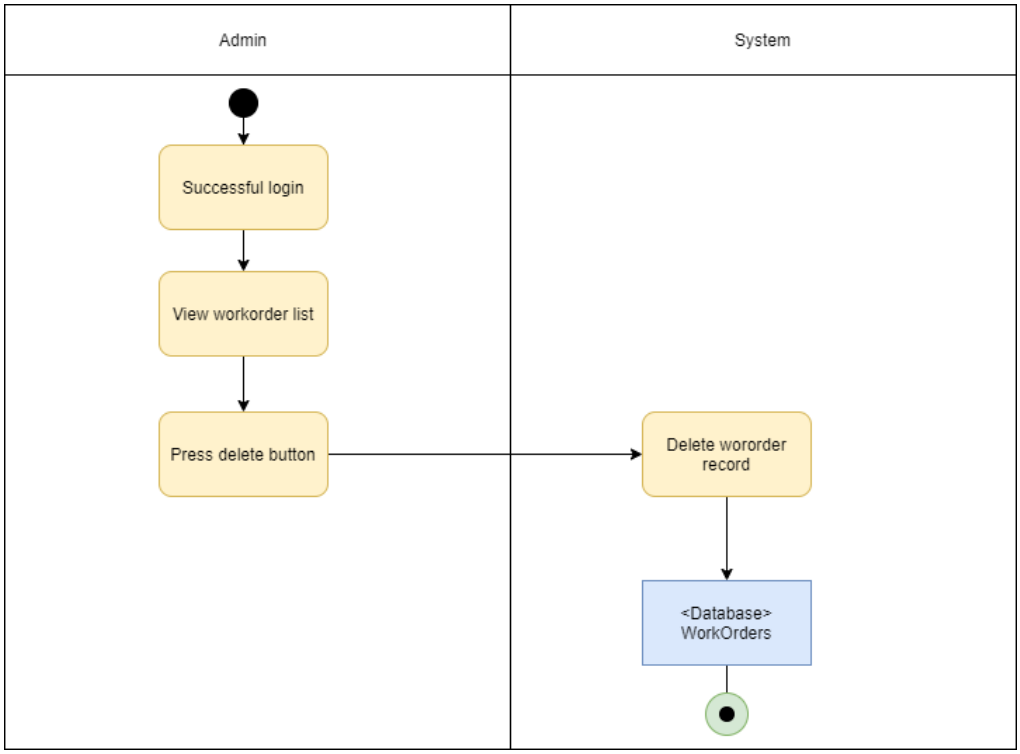


Figure 5.15: Activity diagram for admin(delete workorder)

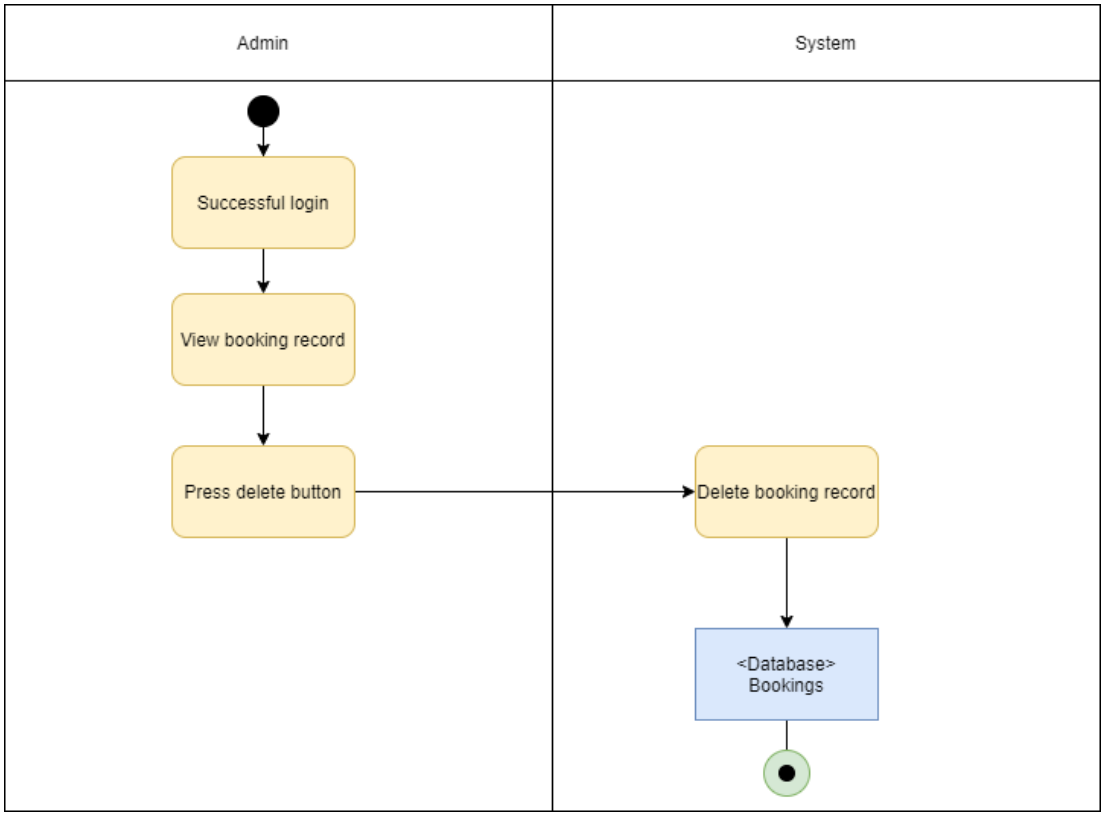


Figure 5.16: Activity diagram for admin(delete booking record)

5.4.1.2 Mobile Application

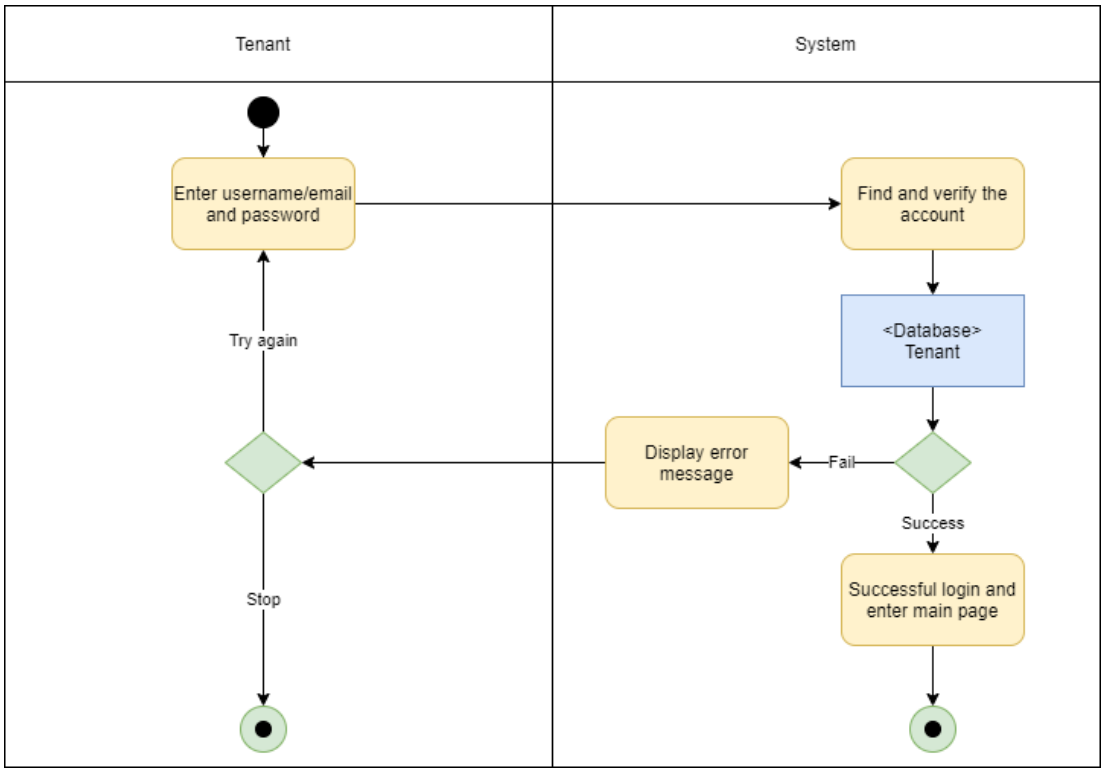


Figure 5.17: Activity diagram for tenant (log in)

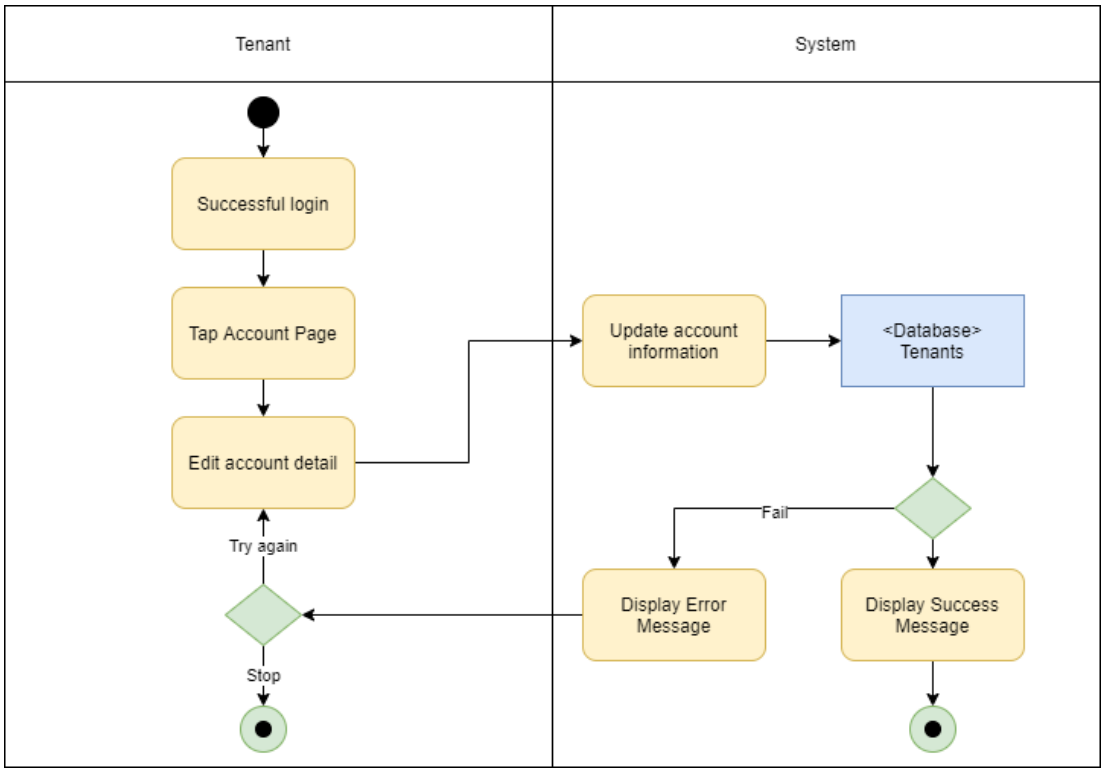


Figure 5.18: Activity diagram for tenant (edit profile)

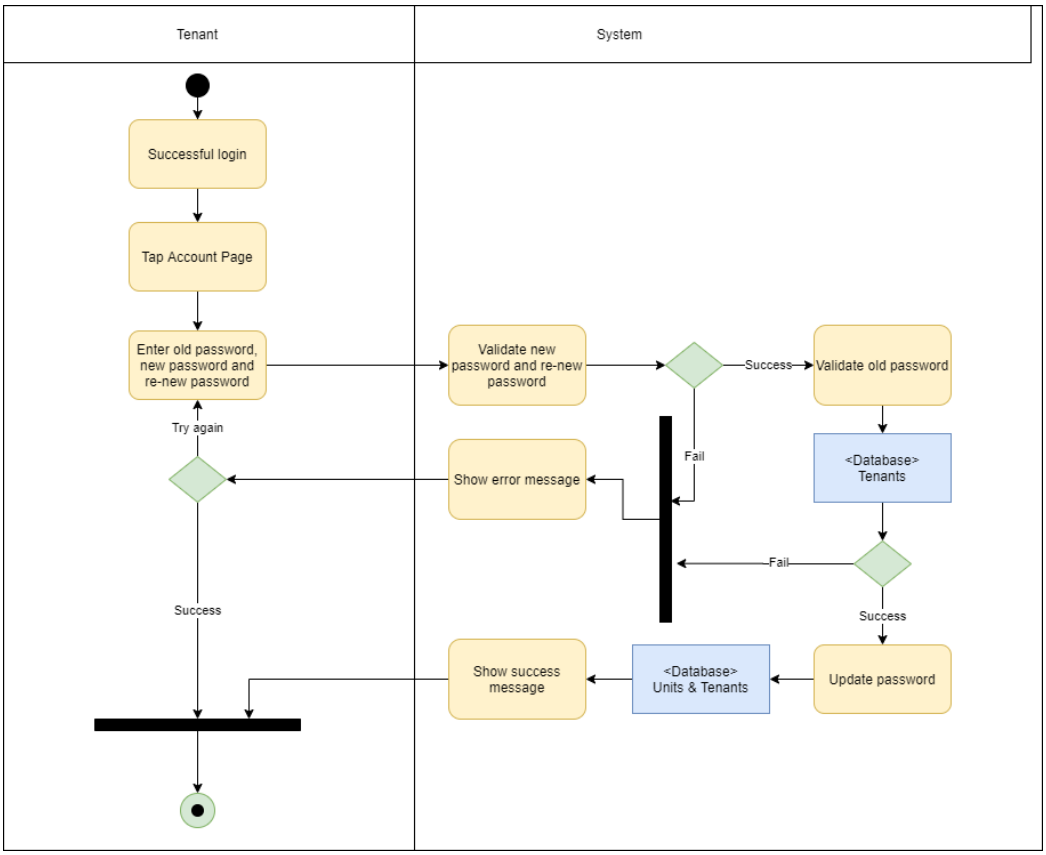


Figure 5.19: Activity diagram for tenant (change password)

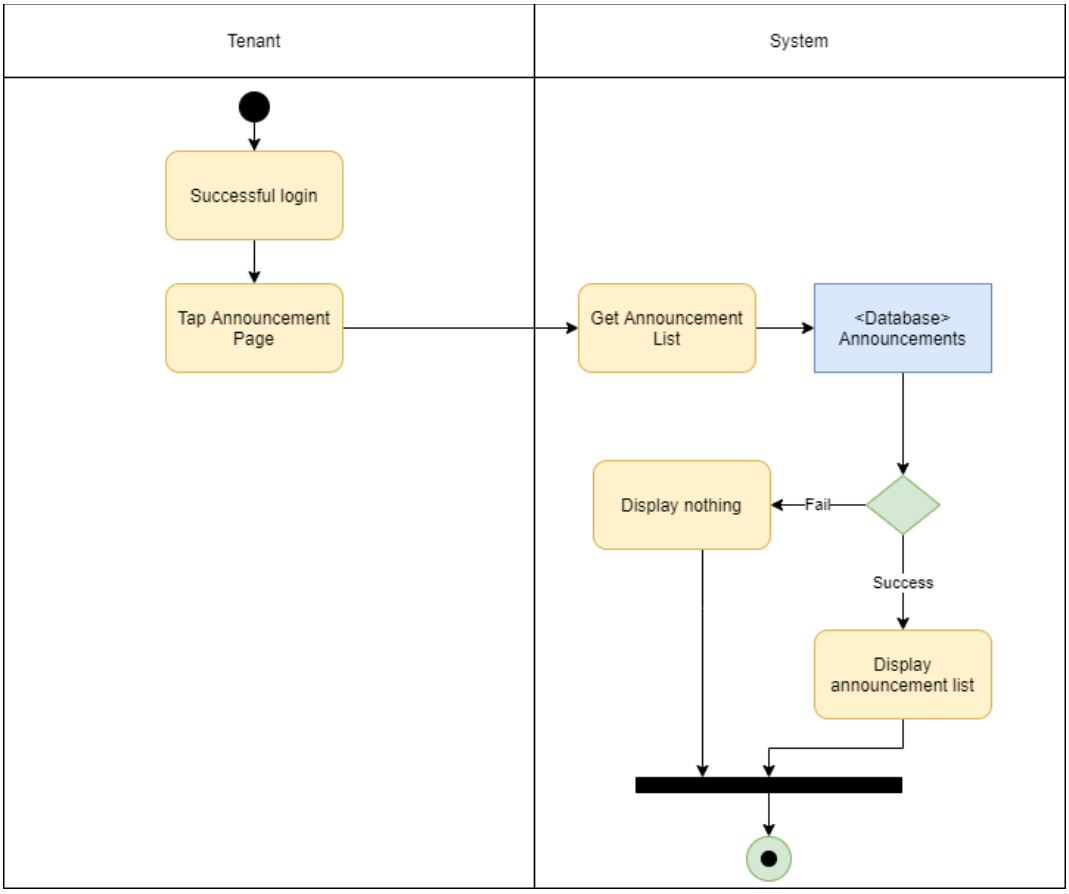


Figure 5.20: Activity diagram for tenant (view announcement)

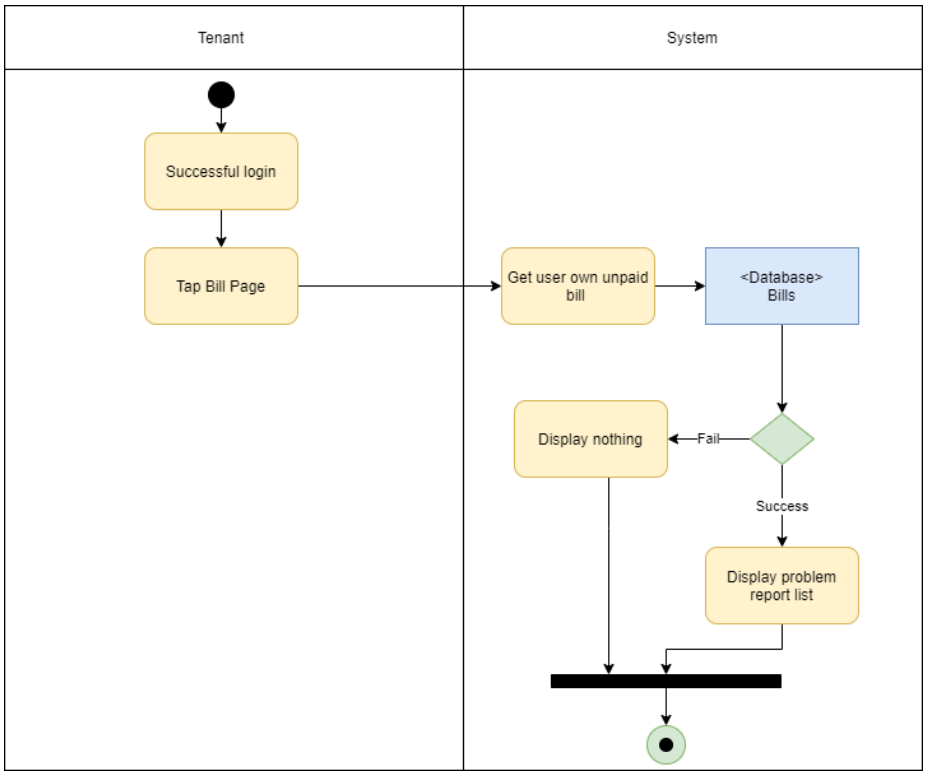


Figure 5.21: Activity diagram for tenant (view bill)

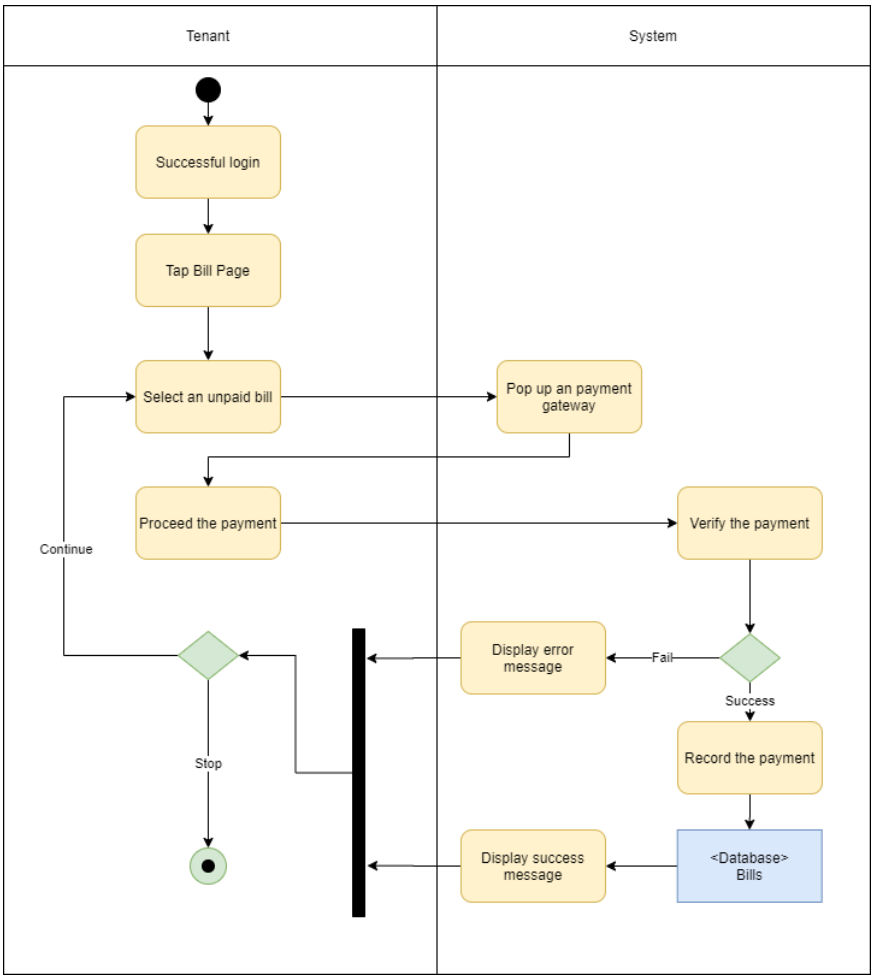


Figure 5.22: Activity diagram for tenant (pay bill)

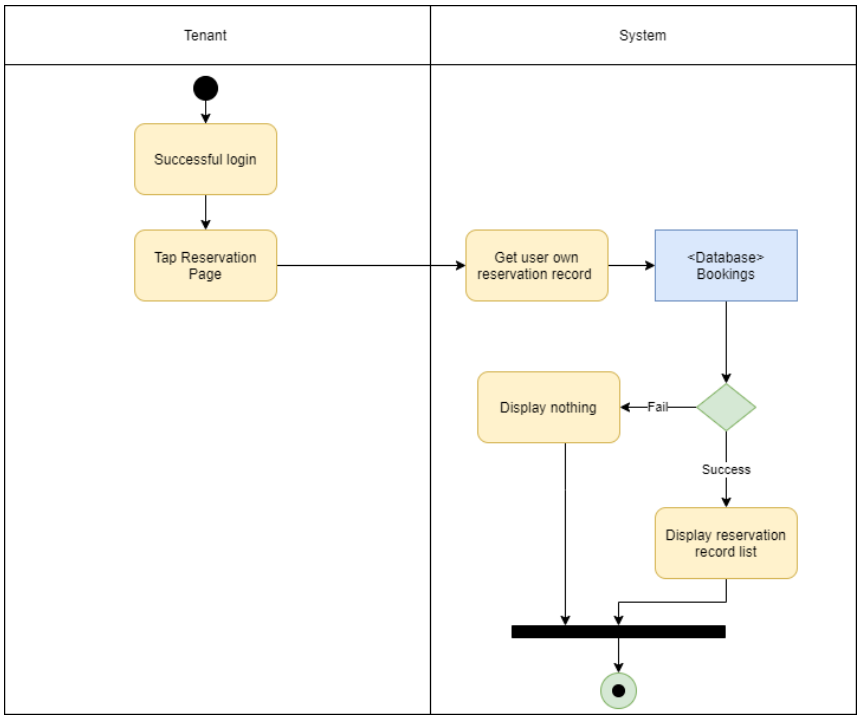


Figure 5.23: Activity diagram for tenant (view reservation)

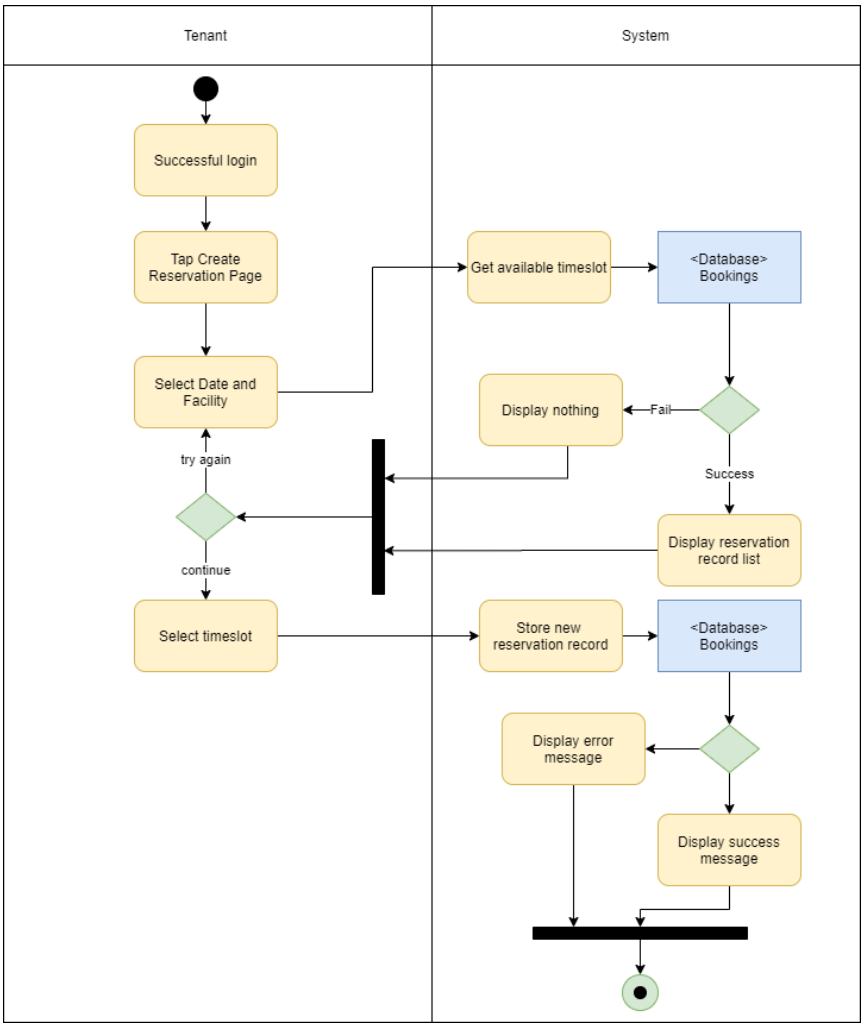


Figure 5.24: Activity diagram for tenant (create new reservation)

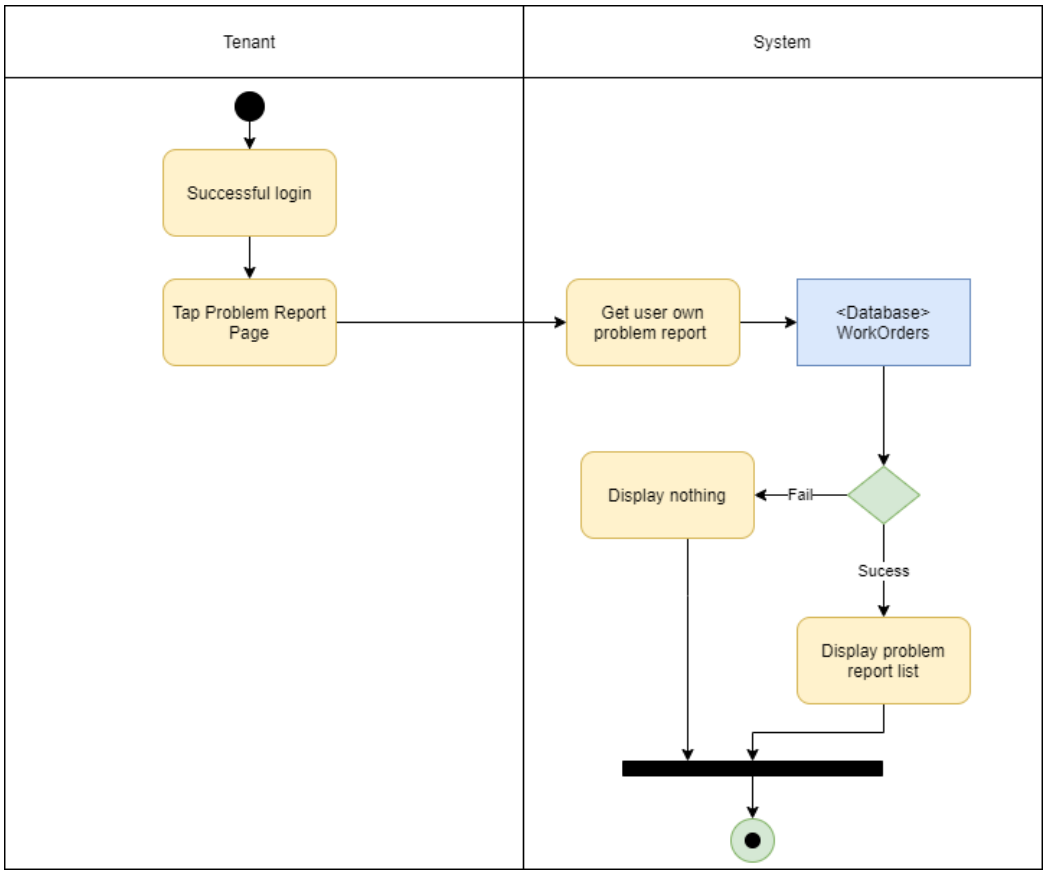


Figure 5.25: Activity diagram for tenant (view problem report)

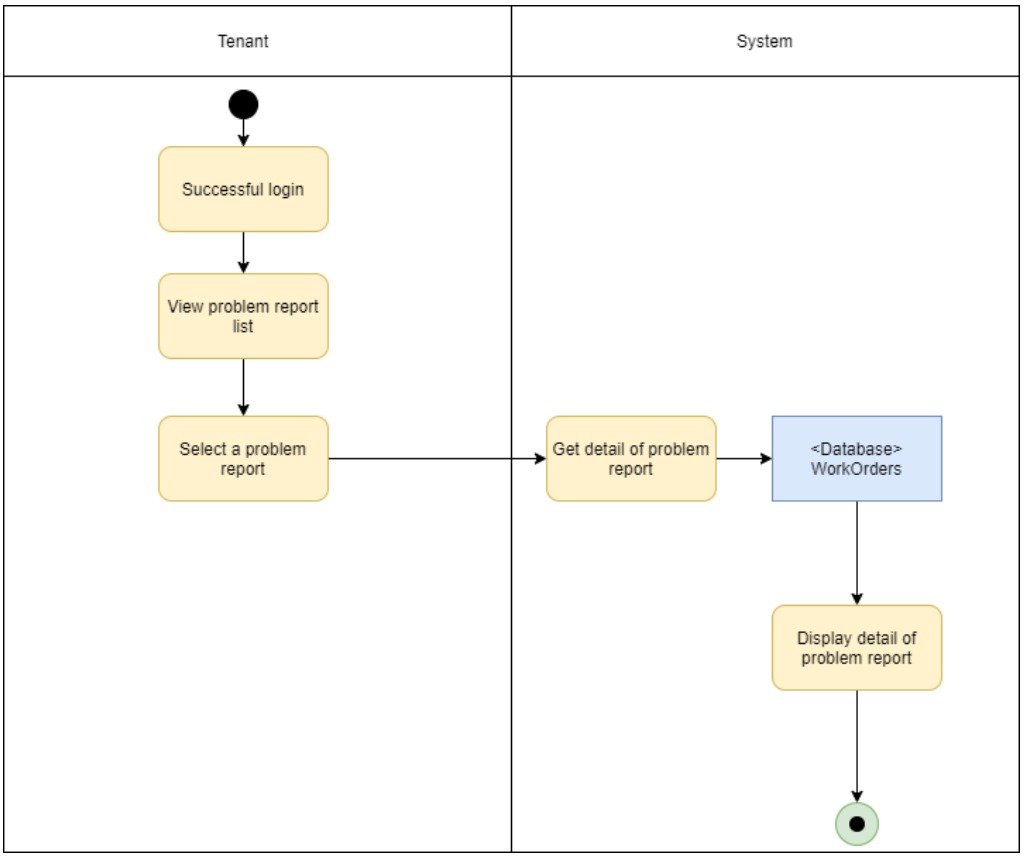


Figure 5.26: Activity diagram for tenant (view detail of problem report)

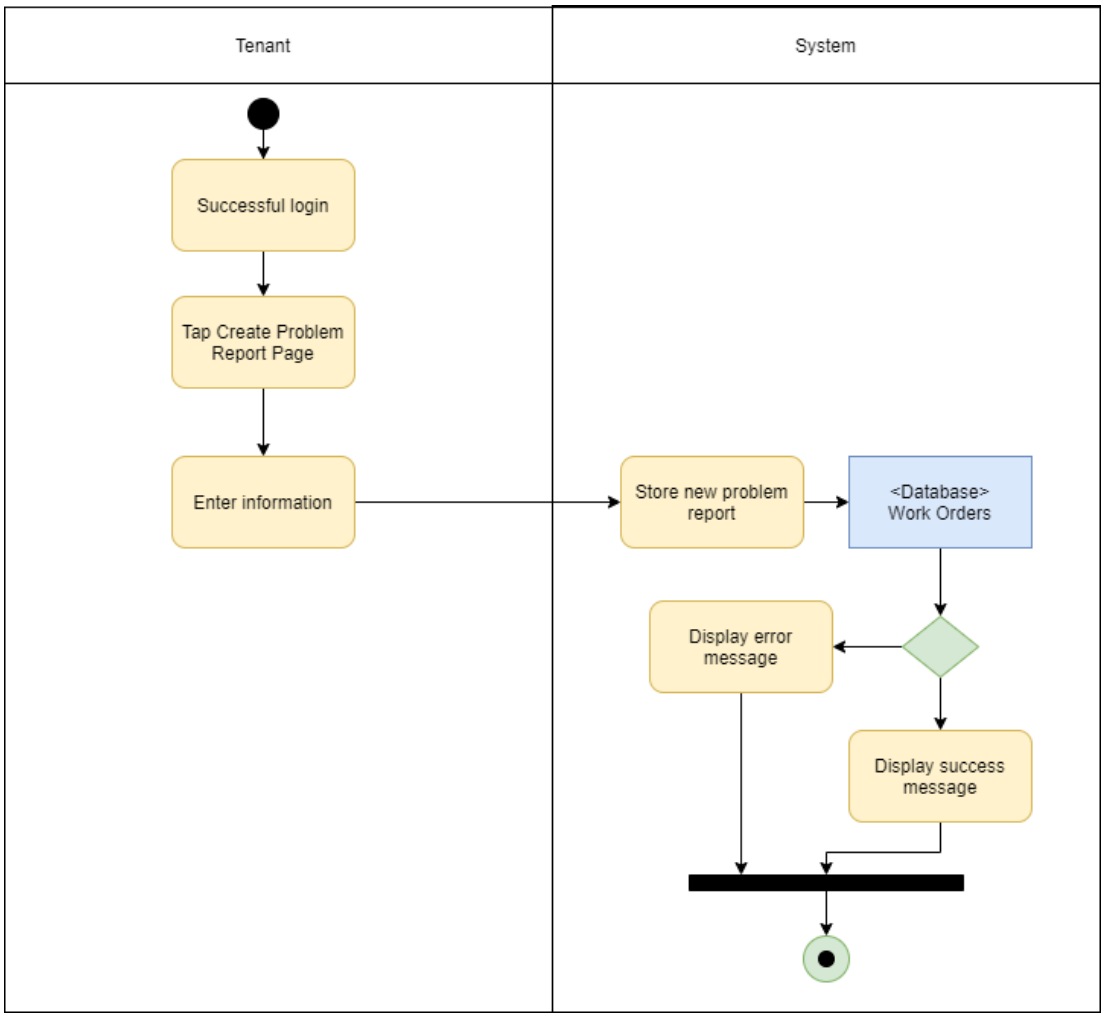


Figure 5.27: Activity diagram for tenant (create problem report)

5.4.2 Data Flow Diagram

The processes and data function of the system can be illustrated by the data flow diagram, while it can define how the data transferred from the input of users.

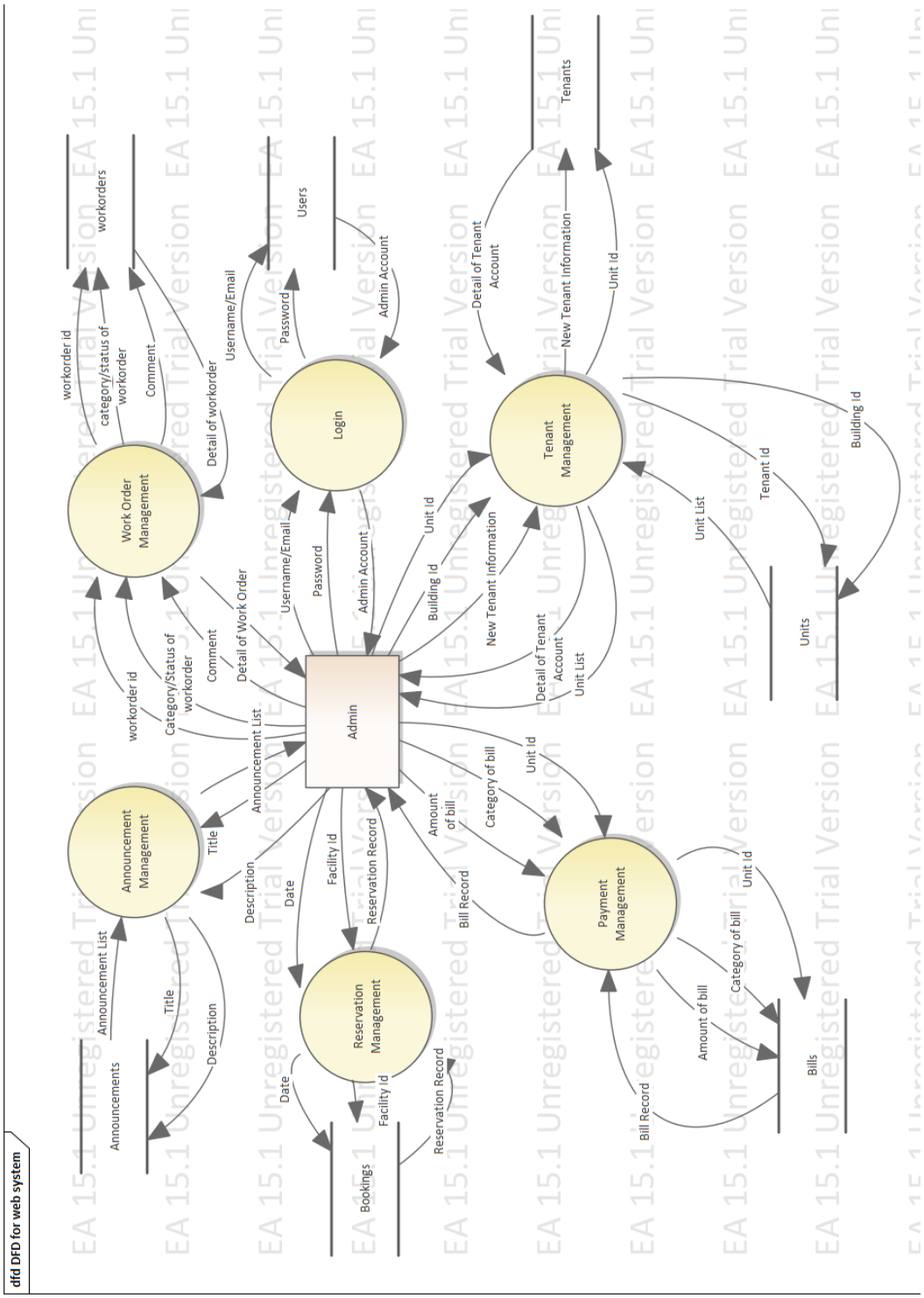


Figure 5.28: Data Flow Diagram for web system

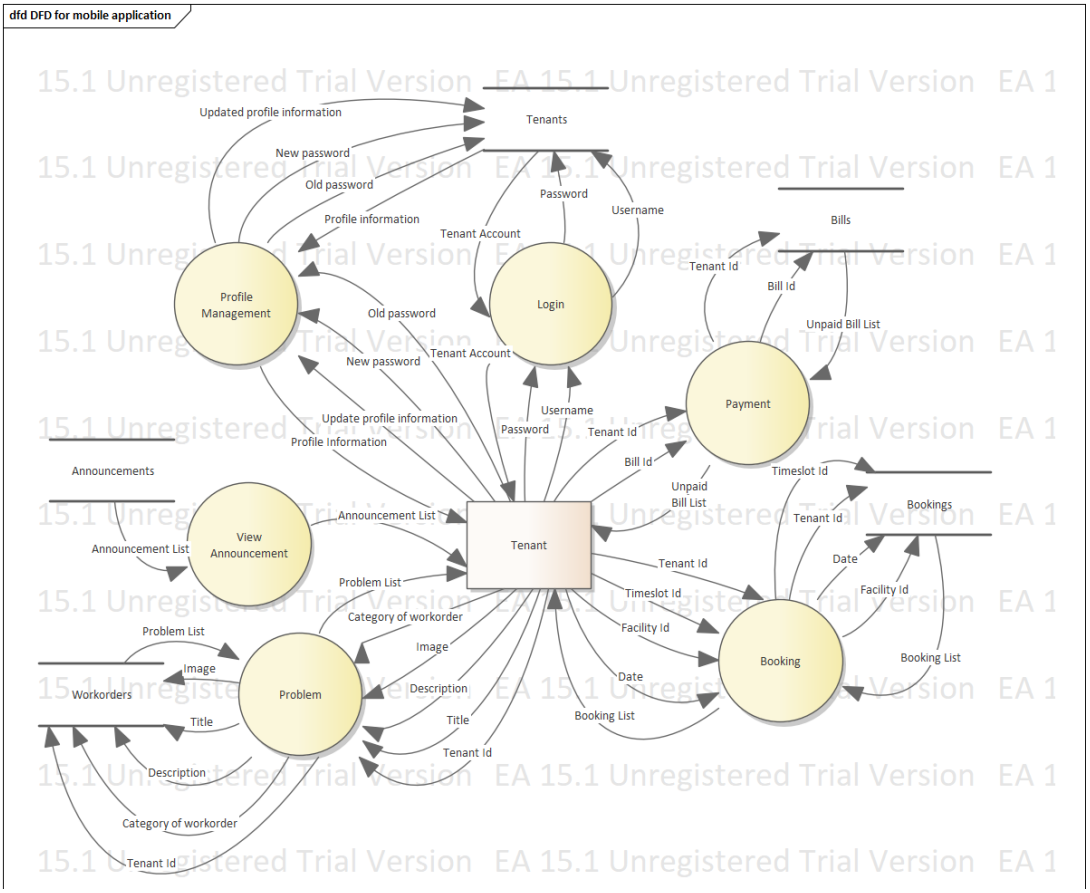


Figure 5.29: Data Flow Diagram for mobile application

5.5 User Interface Design

All software engineer should be attached the UI Design as it is the way that help the system to connect and interact with the user. In this project, there will be two different interactive graphical layouts of system, which are the Web System and Mobile Application.

5.5.1 Web System

All of the web system will be using two types of UI layout design, likely Single Section Design and Multiple Section Design. All of the main module of the system will implement on the navigation bar, the below information will be changed by current page.

Single Section Design will only implement when the contents of the page are simple, for an example, the Announcement Page will only display announcement without any other condition. Besides, all of the create page and login page also use the Single Section Design

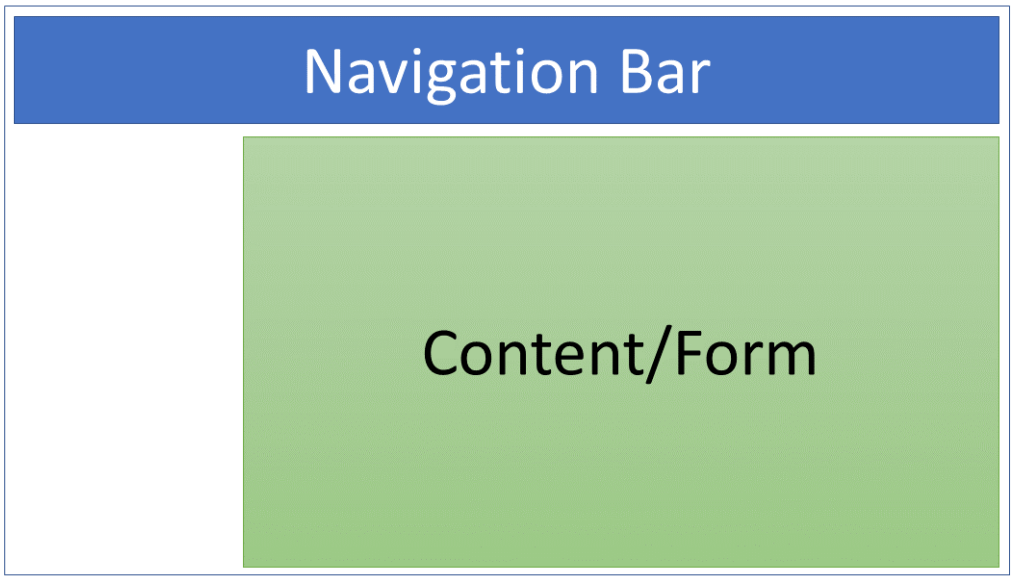


Figure 5.30: Single Section Design

If the contents will be too complex with a lot of options, the page will adopt the Multiple Section Design. The content list can be filtered by selecting the option in filter list. Admin can click the selected button to call and display other dialog box to proceed other actions such as add bill, view information and update progress.

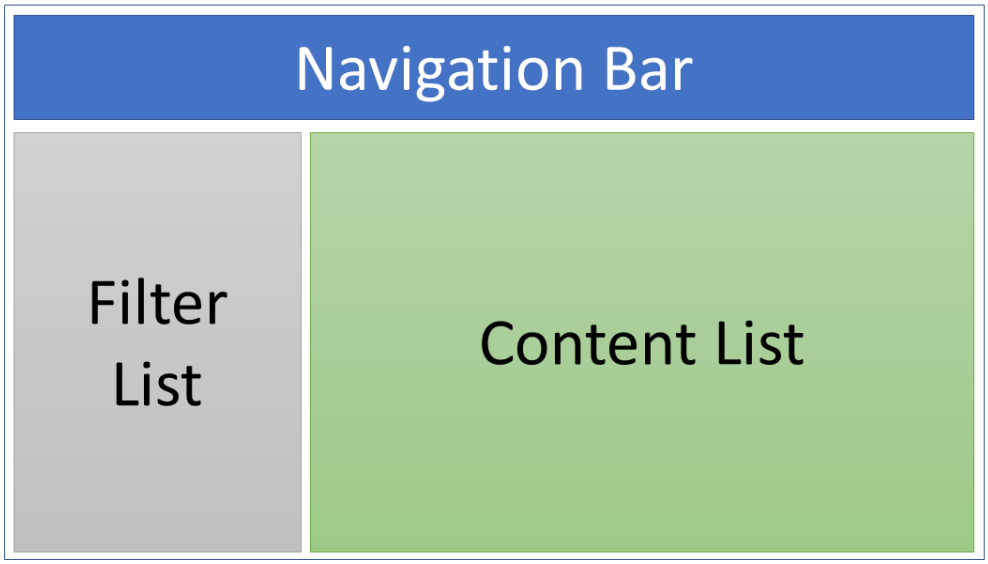


Figure 5.31: Multiple Section Design

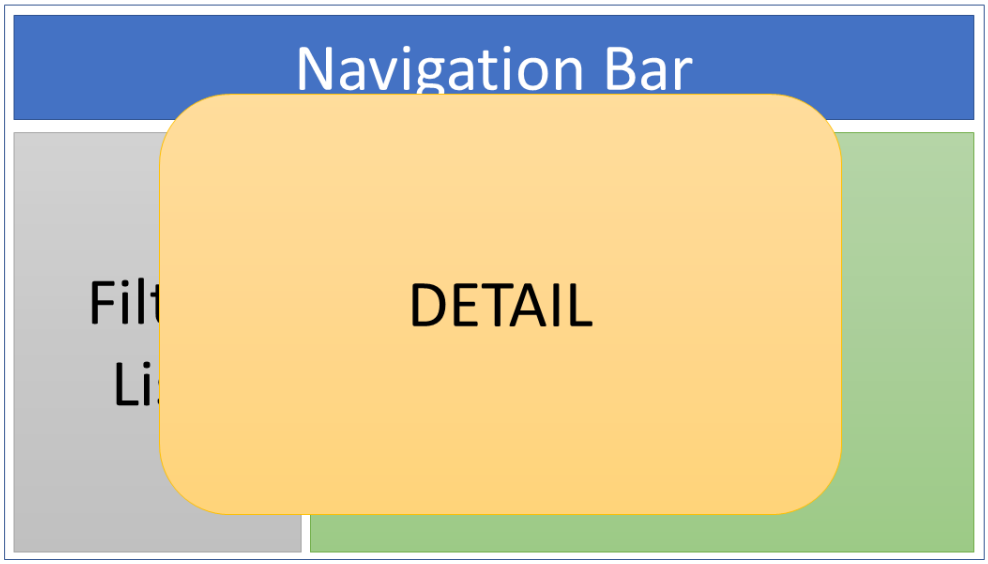


Figure 5.32: Multiple Section Design with pop up dialog box

5.5.2 Mobile Application

In the mobile app, the bottom navigation bar will be adopted in the main page. It is used to navigate to different functionality page. The page might display the floating action button when some pages allow the tenant to create or submit some data. When user redirect the next page, the navigation bar will be hidden and the back button will be shown.

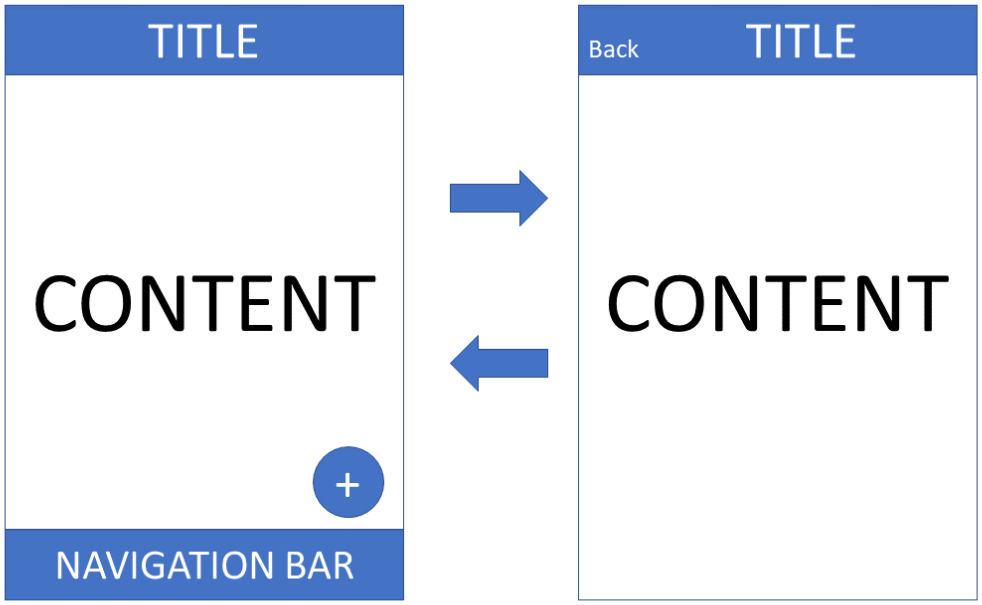


Figure 5.33: Layout Design of Mobile Application

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

There are 2 types of users will interact with the system, likely Apartment Admin and Tenants. Those users have different modules for operating respective features that provided by this system. The modules for each user will be shown as below:

User	Modules
Admin	Login Tenant Management Announcement Management Work Order Management Bill Management Reservation Management
Tenants	Login View Announcement Payment Booking Problem Report Profile Management

Table 6.1: Module list for each user

6.2 Module for Admin

6.2.1 Login

Before accessing the system, the admin must log in by entering the username/email and password. If the credential data is matched with the data in database, the system will allow the user to access the system, or else an error message will be displayed by system.

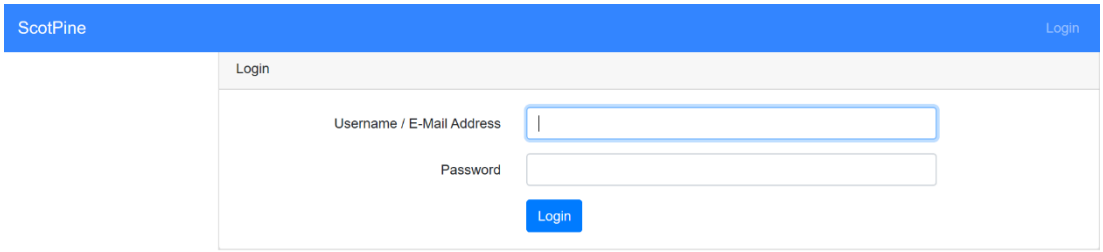


Figure 6.1: Login Page for Admin

6.2.2 Tenant Management

In this system, only admin is allowed to create the tenant account. In the create tenant account page, the system will show the unit that haven't create account after selecting the building and floor. There are few input validations are implemented in this page, such as the username must be unique in the database and the email must follow the email format. After passing the input validation, the system will store the data in database.

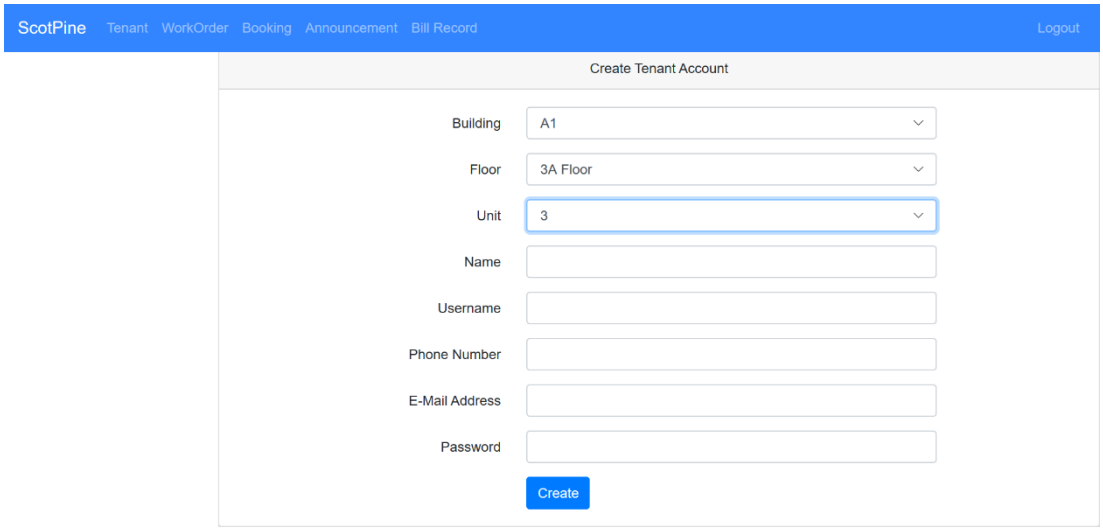


Figure 6.2: Create Tenant Account Page

The unit list will be displayed after the admin select an building.

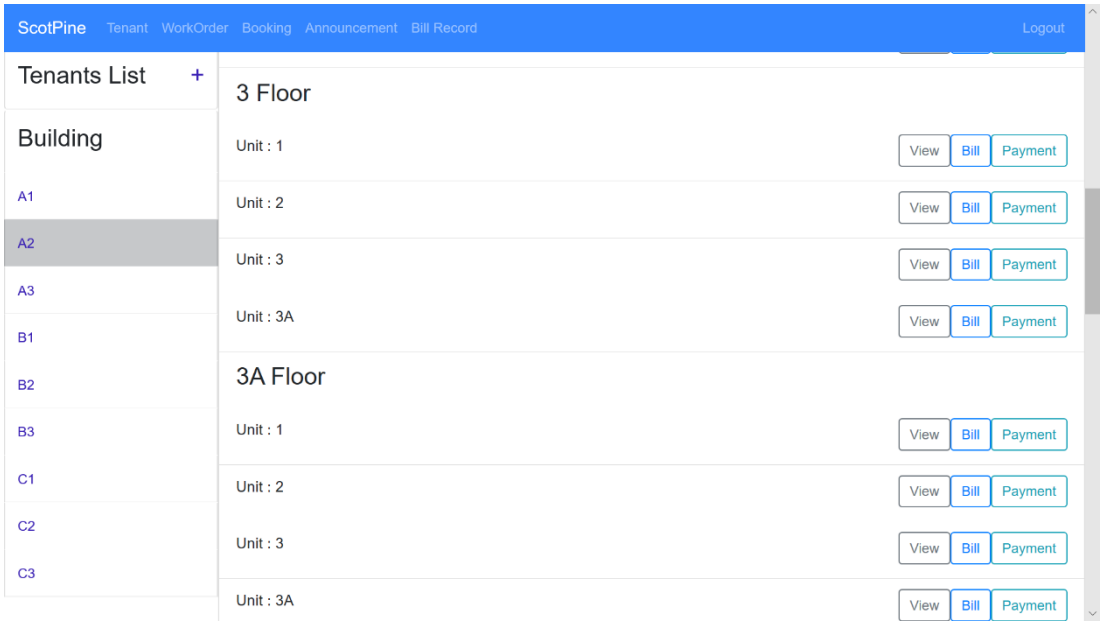


Figure 6.3: View Unit List

If the selected unit which have register an account, the account detail will be displayed when admin click the view button, or else system will display a message. Admin also can delete the tenant account.

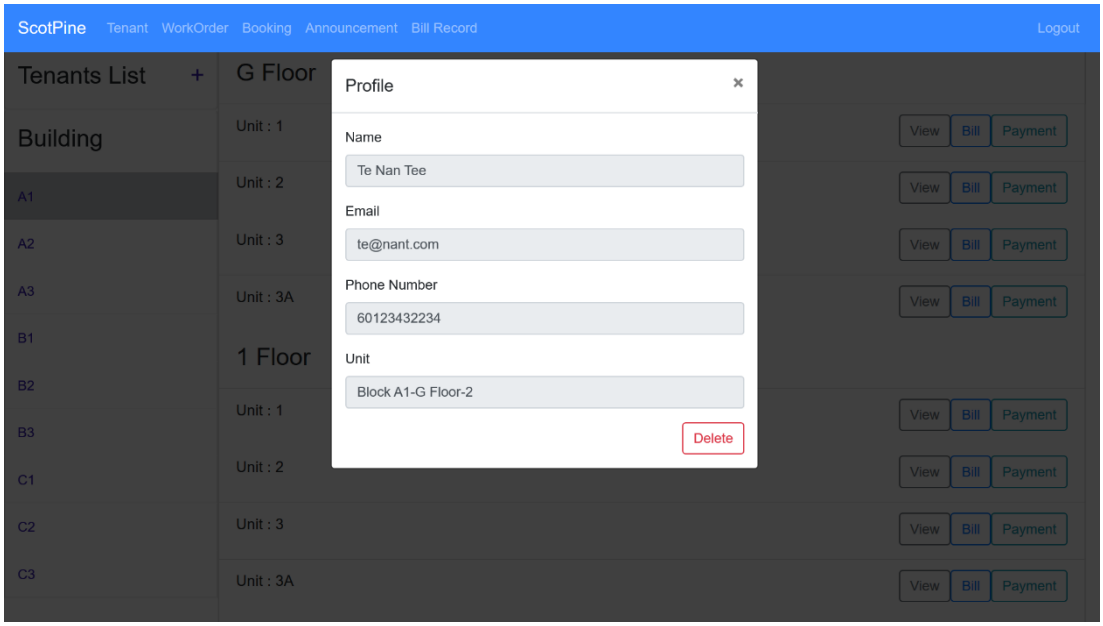


Figure 6.4: View Tenant Account Page

6.2.3 Announcement Management

The admin is able to view all the announcement released. The admin also allowed to release new announcement and delete old announcement.

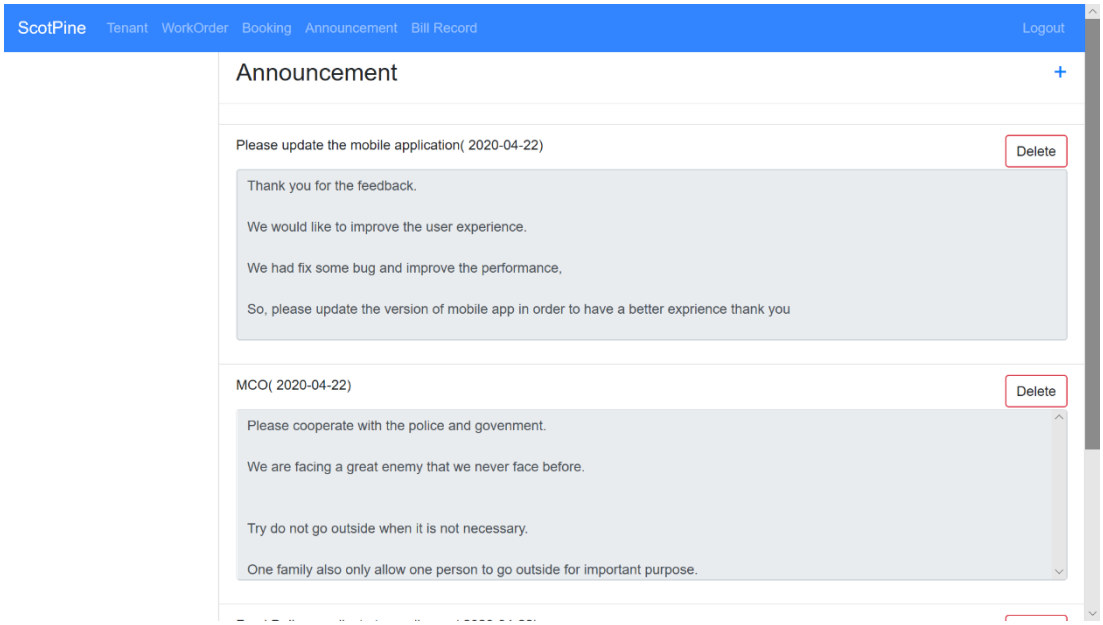


Figure 6.5: View Announcement List Page

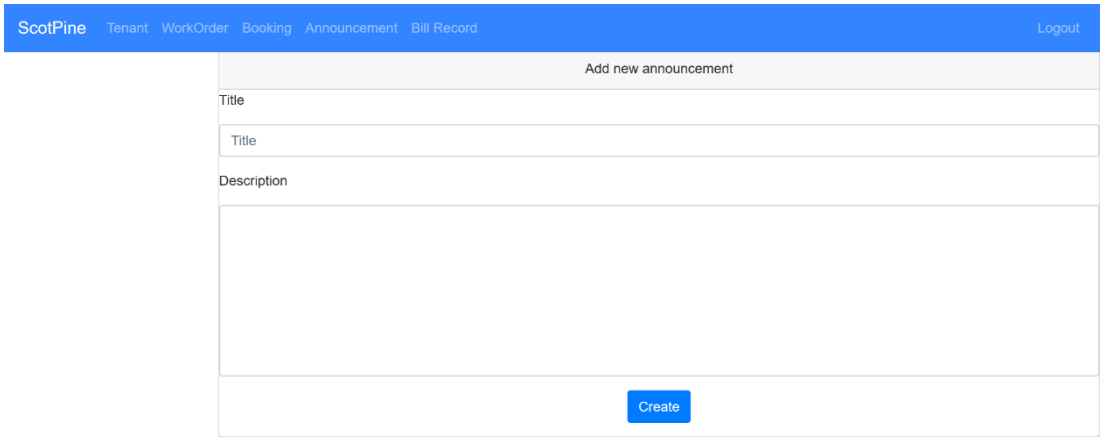


Figure 6.6: Create Announcement Page

6.2.4 Work Order Management

Admin can view the work order by select the category or status, or else the system will show all work order without any data filtering. Besides, the admin also can delete the work order.

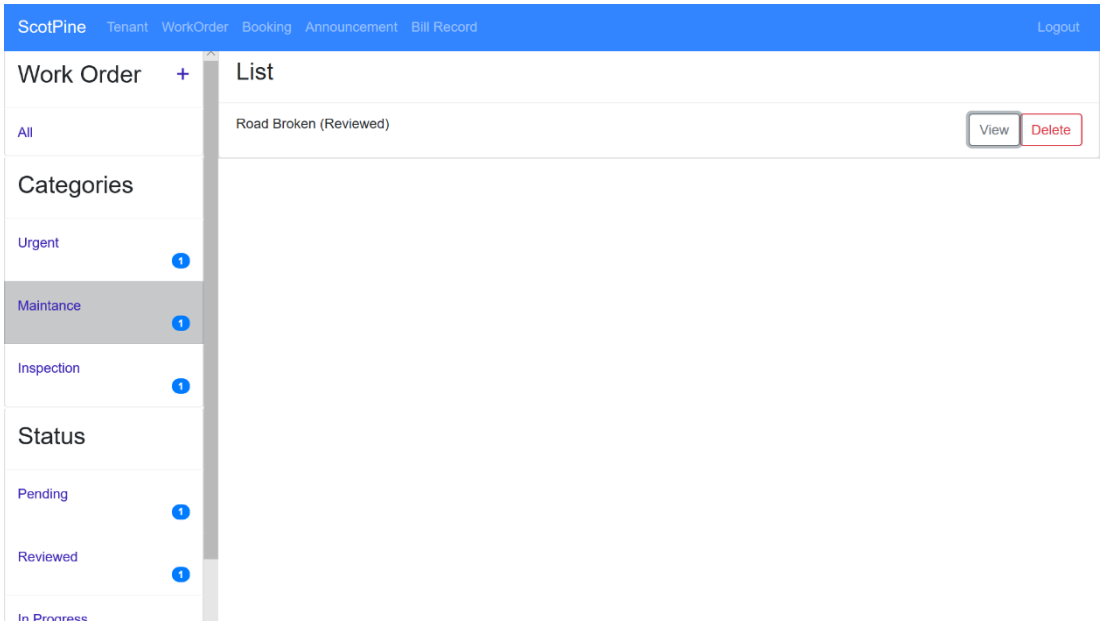


Figure 6.7: View WorkOrder List Page

After pressing the view button, admin can view the detail of work order. Admin also can update the work order by changing the status and leaving an comment.

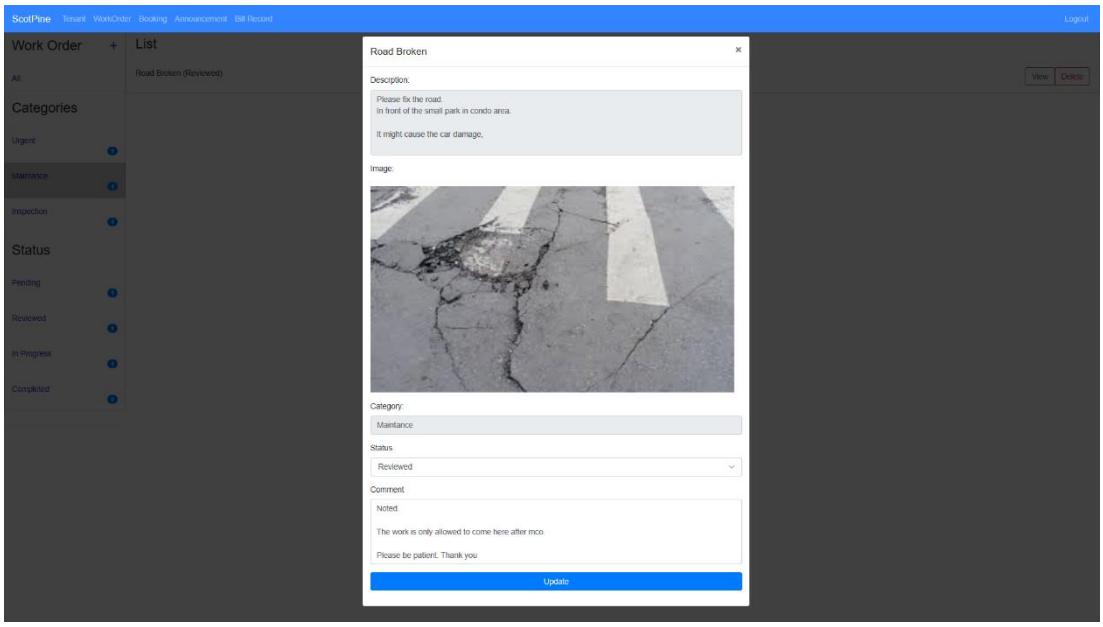


Figure 6.8: View WorkOrder Detail and Update WorkOrder

6.2.5 Bill Management

Admin also can add new bill for any tenant by pressing the bill button in the tenant list page. There will be 3 categories of bill can be selected.

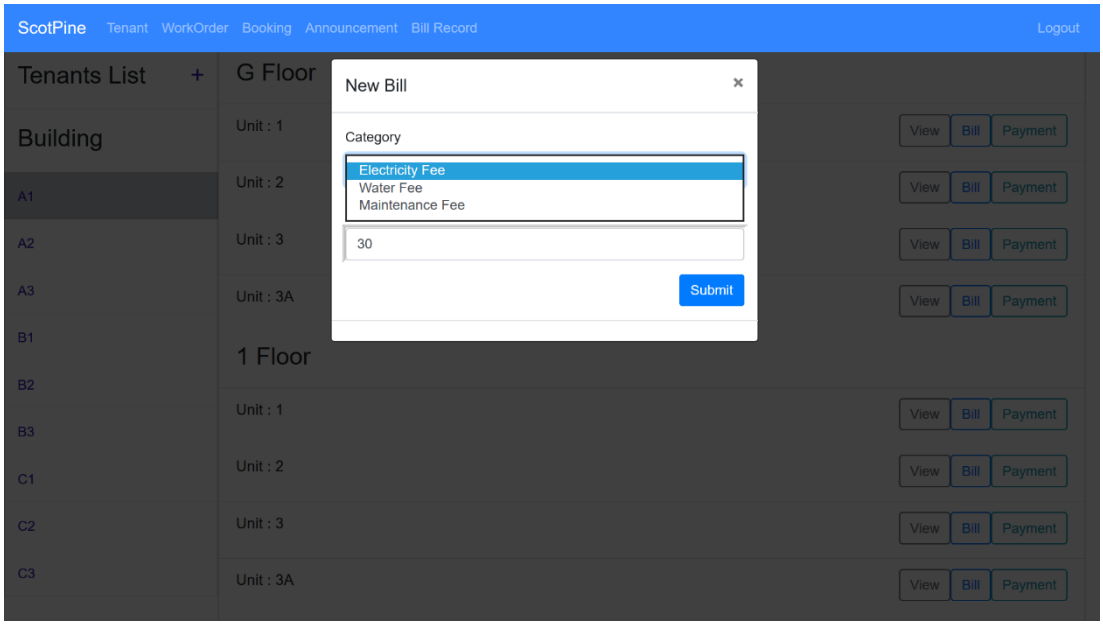


Figure 6.9: Add new Bill

By pressing the payment button in tenant list page, admin can record the payment when the tenant chooses to walk-in payment. After select the category of bill, the unpaid bill of the category will be displayed.

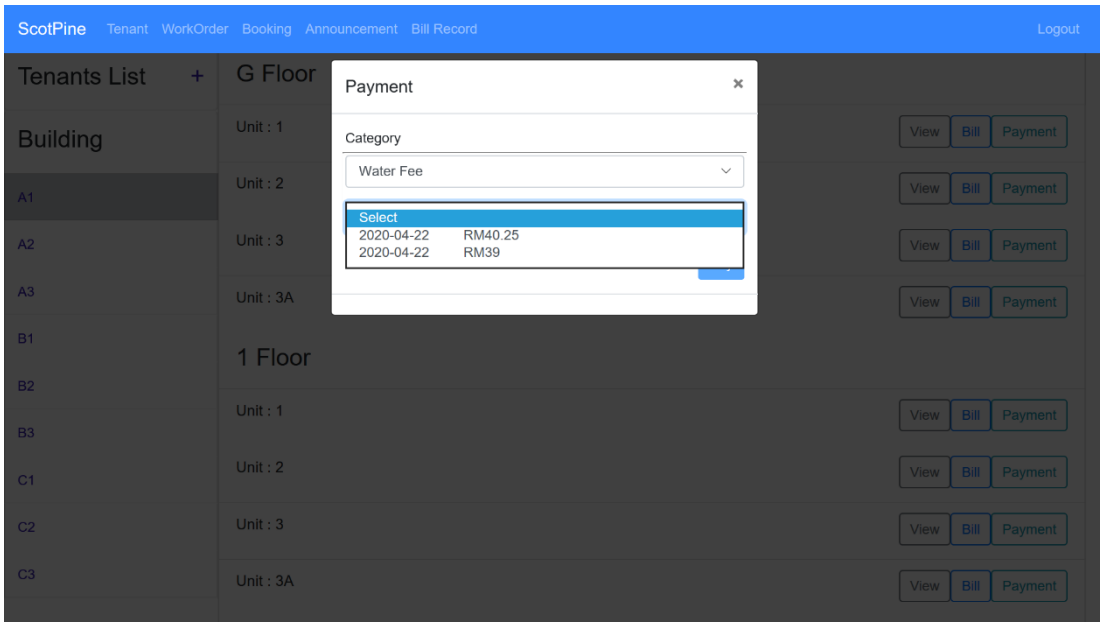


Figure 6.10: Record Payment

Admin also can view the bill record (/history) by clicking the bill record button on the navigation bar. After selecting a unit, the unit's bill record will be displayed.

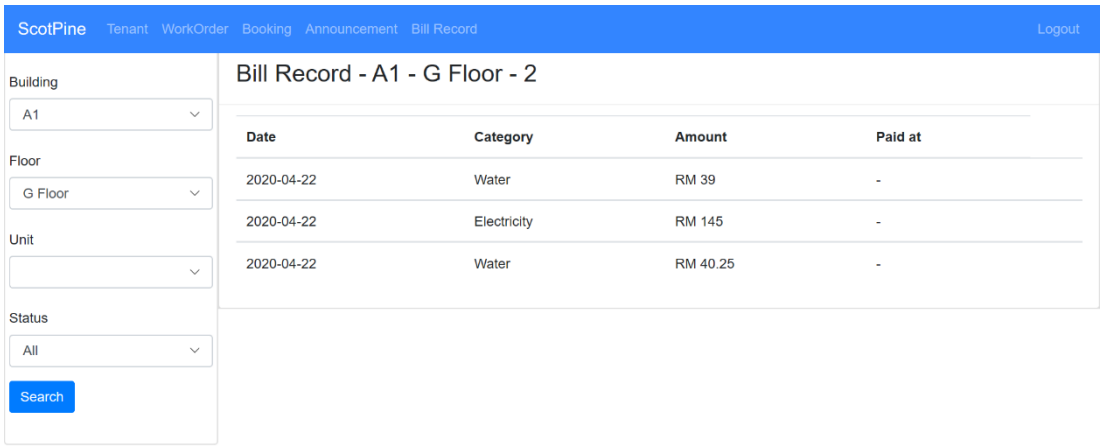


Figure 6.11: View Bill Record Page

6.2.6 Reservation Management

The reservation record will be displayed by clicking the booking button on the navigation bar. The booking record will be shown after selecting the date and facility. The booking can be deleted by admin.

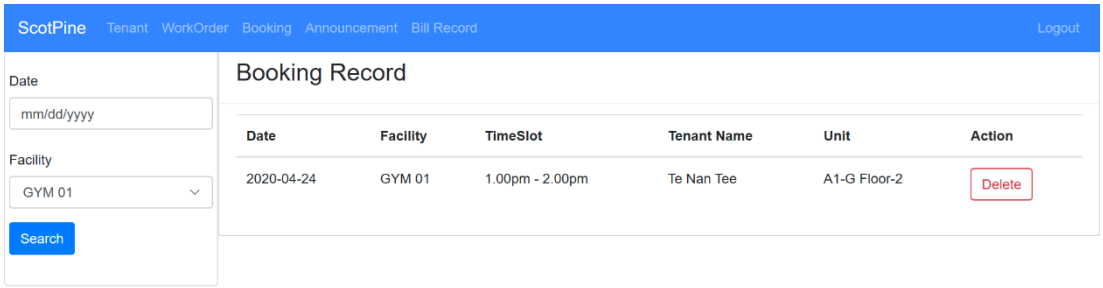


Figure 6.12: Reservation Page

6.3 Module for Tenant

6.3.1 Login

Before using the application, the tenant must log in by entering the username and password. If the credential data is matched with the data in database, the API will send data to allow the user to access the application, or else an error message will be pop up.

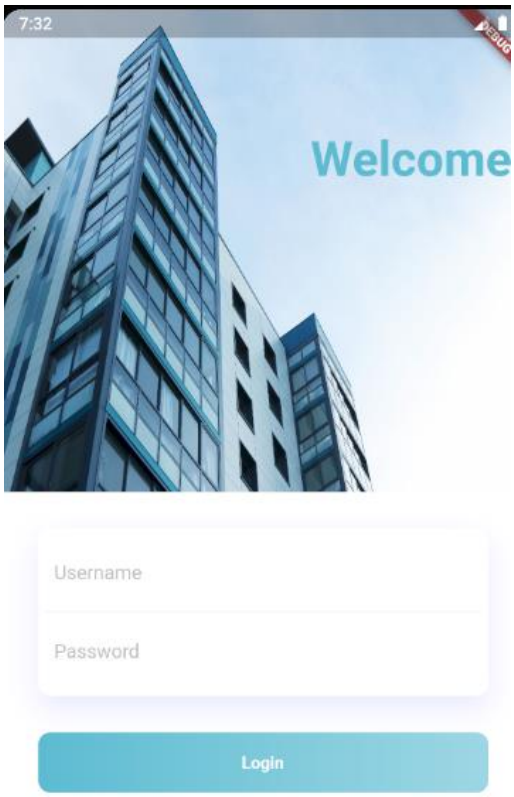


Figure 6.13: Login Page for tenant

6.3.2 View Announcement

Tenant can view all announcement that released by the admin. In the announcement list, there will show the title and date only, but user can tap the selected announcement to expand the detail.

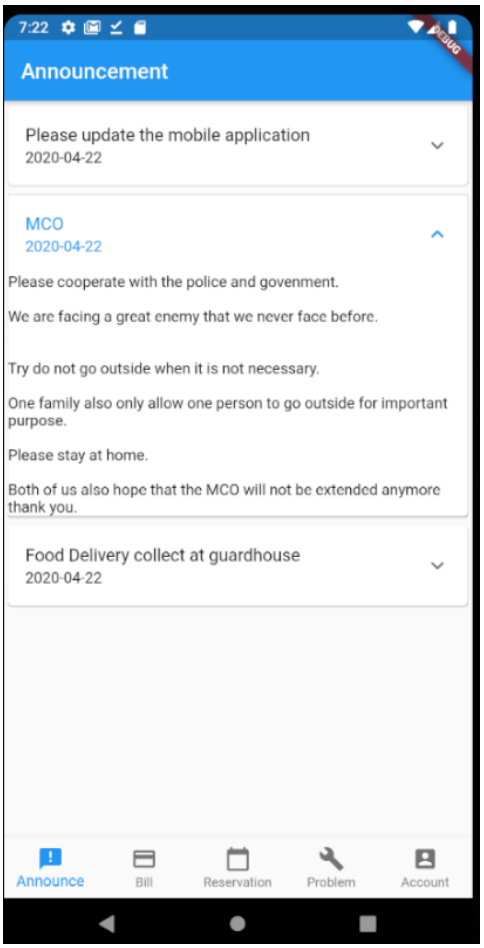


Figure 6.14:Announcement Page

6.3.3 Payment

The app will only display the unpaid bill in bill page. After clicking any bill, it will pop up a third party payment gateway. If payment success, the payment will be recorded in system, else app will display an error message.

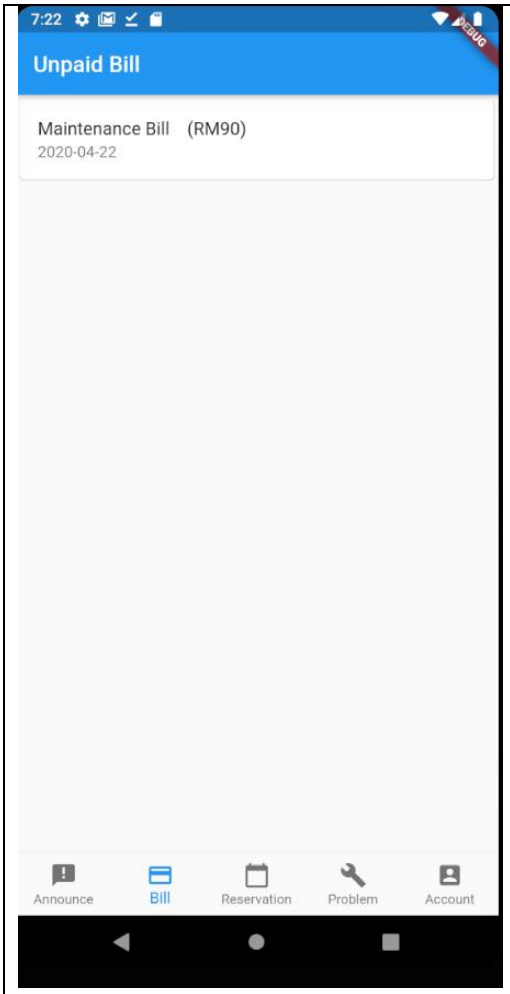


Figure 6.15:Bill Page

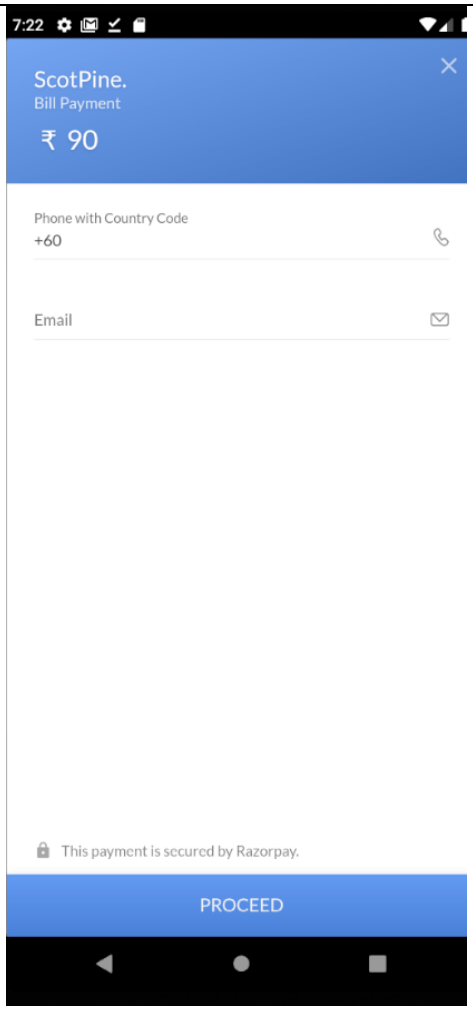


Figure 6.16:Payment page

6.3.4 Booking

Tenant can view the reservation order in the reservation page, but it will not show the past reservation order. Tenant also can book new reservation by submitting the date, facility and timeslot. After select date and facility, the available timeslot will be shown by clicking the check timeslot button.

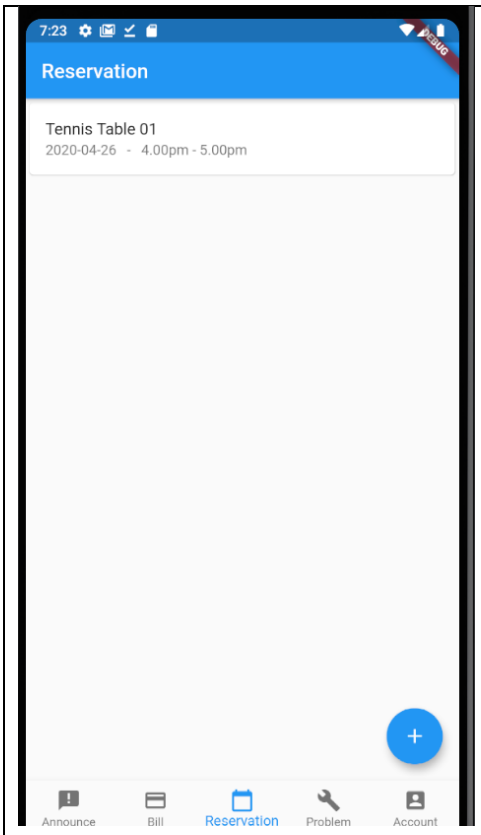


Figure 6.17:Reservation Page

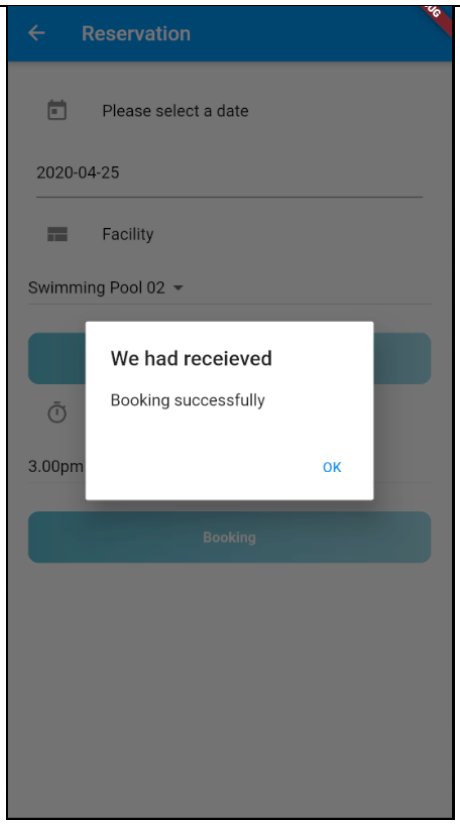


Figure 6.18:Create new reservation Page

6.3.5 Problem Report

Tenants can only view their own problem report. Tenant is allowed to submit new problem report via the application.

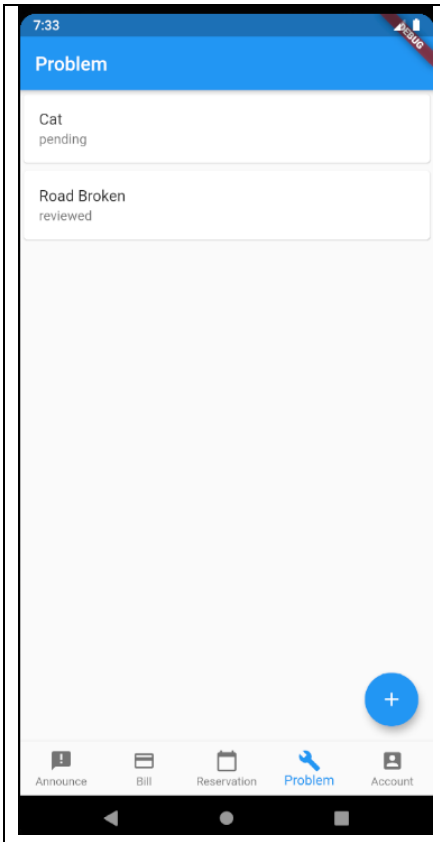


Figure 6.19:Problem Page

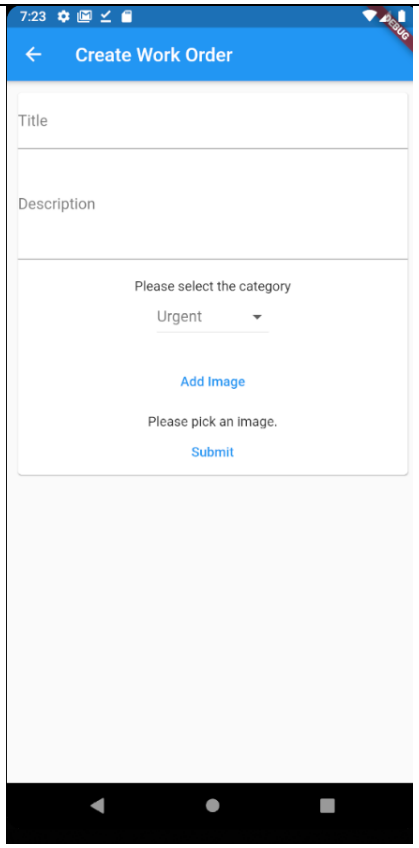


Figure 6.20:Report problem page

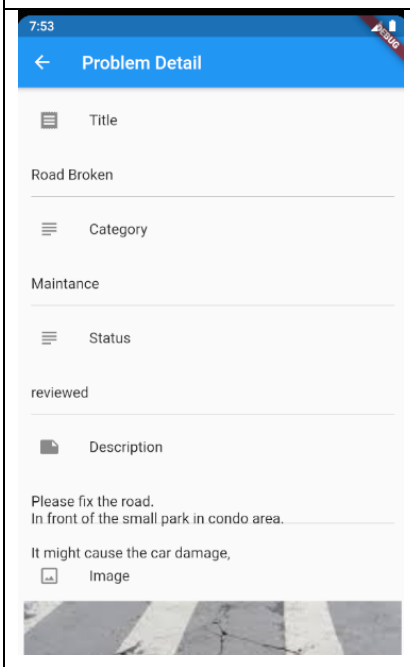


Figure 6.21:View Problem Page 01

Figure 6.22:View Problem Page 02

6.3.6 Profile Management

Tenant can change the information of account in account page. User also can change the password, and it is encourage to change the password after first time login.

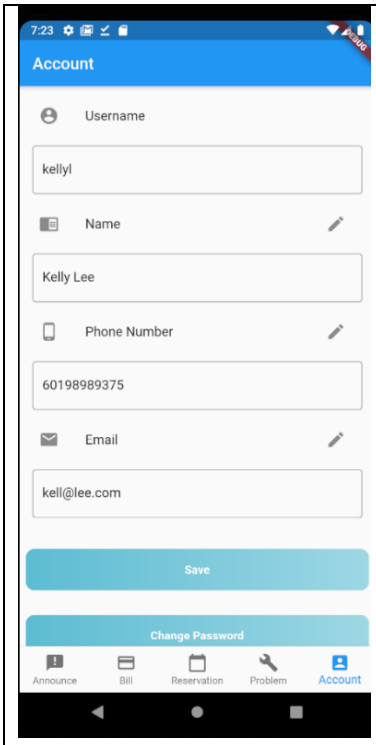


Figure 6.23:Account Page

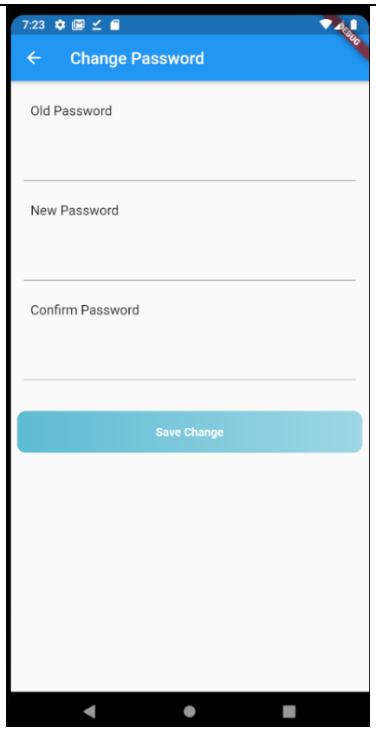


Figure 6.24:Change Password Page

CHAPTER 7

SOFTWARE TESTING

7.1 Introduction

After design and implement the system, the project's system is needed to conduct the software testing in order to ensure the quality of system and the scope of system. In this chapter, there are 3 type of testing are involved, such as unit test, user acceptance test and usability test.

7.2 Unit Test

All of the modules will split into small and individual unit, so the unit testing can validate each function of system is able to perform as the requirement.

Test Case No	01	**User**	Admin
Module	Login		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
Login without any input	Click Login button	Login fail with an error message.	**(pass)**
Login with incorrect username/email and correct password	Enter username/email Enter password Click Login button	Login fail with an error message.	**(pass)**
Login with correct username/email and incorrect password	Enter username/email Enter password Click Login button	Login fail with an error message.	**(pass)**
Login with correct username/email and correct password	Enter username/email Enter password Click Login button	Login successful and redirect to tenant page.	**(pass)**

Table 7.1: Unit Test Case for Admin (Login)

Test Case No	02	**User**	Admin
Module	Tenant Management		
Pre-condition	The admin log in successful.		
	The admin clicks the tenant button on the navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View existing tenant account by selecting the unit	Select building	Pop up the information of the tenant account.	**(pass)**
	Find the selected unit		
	Click view button		
View not existing tenant account by selecting the unit	Find the selected unit	Pop up a message show that the select unit haven't create an account.	**(pass)**
	Click view button		
	Click Login button		
Show the unit that haven't create account in tenant create page	Select building	The unit dropdown list will only display the unit that haven't create account.	**(pass)**
	Select floor		
Create a tenant account with an existing username	Selected building, floor and unit	Pop up a message show that the username is existing.	**(pass)**
	Enter existing username		
	Fill in other needed information		
	Click the create button		
Create a tenant account with new username	Selected building, floor and unit	The tenant account is created successful.	**(pass)**
	Enter existing username		
	Fill in other needed information		
	Click the create button		

Delete a tenant account	Select building	The tenant account is deleted successful.	**(pass)**
	Find the selected unit		
	Click view button		
	Click delete button		
	Click confirm button		
Delete a tenant account without confirmation	Select building	Everything remains the same.	**(pass)**
	Find the selected unit		
	Click view button		
	Click delete button		
	Click cancel button		

Table 7.2: Unit Test Case for Admin (Tenant Management)

Test Case No	03	**User**	Admin
Module	Announcement Management		
Pre-condition	The admin log in successful.		
	The admin clicks the announcement button on the navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View announcement	-	The announcement list is displayed	**(pass)**
Delete announcement without confirmation		Everything remains the same.	**(pass)**
	Click delete button		
	Click cancel button		
Delete announcement	View announcement list	The selected announcement is deleted successful.	**(pass)**
	Click delete button		
	Click confirm button		
Create new announcement	View announcement list	New announcement is released and redirect back to announcement list page.	**(pass)**
	Click “+” button		

	Fill in the information		
	Click create button		

Table 7.3: Unit Test Case for Admin (Announcement Management)

Test Case No	04	**User**	Admin
Module	Work Order Management		
Pre-condition	The admin log in successful.		
	The admin clicks work order button on the navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
Display work order with the selected category	Select a category	The system displays the work order that under the selected category.	**(pass)**
Display work order with the selected status	Select a status	The system displays the work order that under the selected status.	**(pass)**
View the detail of work order	View work order list	Pop up the detail of work order.	**(pass)**
	Click view button		
View the detail of work order	View work order list	The selected work order is updated successful.	**(pass)**
	Click view button		
	Change the status and fill comment		
	Click update button		
Delete work order without confirmation	View work order list	Everything remains the same.	**(pass)**
	Click delete button		
	Click cancel button		
Delete work order	View work order list	The selected work order is deleted successful.	**(pass)**
	Click delete button		
	Click confirm button		

Table 7.4: Unit Test Case for Admin (Work Order Management)

Test Case No	05	**User**	Admin
Module	Bill Management		
Pre-condition	The admin log in successful.		
	The admin clicks tenant button on the navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
Add new bill	Select a unit Click bill button Select category of bill Enter the amount Click submit button	The bill is created successful	**(pass)**
Add new bill with invalid amount (RM0 or negative)	Select a unit Click bill button Select category of bill Enter the invalid bill amount Click submit button	Error Message is displayed	**(pass)**
Show unpaid bill	Select a unit Click payment button Select category of bill	The unpaid bill is show in the dropdown list.	**(pass)**
Record payment bill	Select a unit Click payment button Select category of bill Select the bill Click pay button	The selected bill is record as paid.	**(pass)**
View bill record	Click bill record button on the navigation bar. Select building, floor and unit. Select status of bill Click search button	The selected unit's bill record is displayed with correct status	**(pass)**

Table 7.5: Unit Test Case for Admin (Bill Management)

Test Case No	06	**User**	Admin
Module	Reservation Management		
Pre-condition	The admin log in successful.		
	The admin clicks booking button on the navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View reservation record	Select a date	The selected facility's reservation record is displayed on the selected date	**(pass)**
	Select a facility		
Delete reservation record without confirmation	View reservation record list	Everything remains the same	**(pass)**
	Click delete button		
	Click cancel button		
Delete reservation record	View reservation record list	The selected reservation record is deleted successful.	**(pass)**
	Click delete button		
	Click confirm button		

Table 7.6: Unit Test Case for Admin (Reservation Management)

Test Case No	07	**User**	Tenant
Module	Login		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
Login without any input	Click Login button	Login fail with an error message.	**(pass)**
Login with incorrect username/email and correct password	Enter username/email Enter password Click Login button	Login fail with an error message.	**(pass)**
Login with correct username/email and incorrect password	Enter username/email Enter password Click Login button	Login fail with an error message.	**(pass)**
Login with correct username/email and correct password	Enter username/email Enter password Click Login button	Login successful and redirect to tenant page.	**(pass)**

Table 7.7: Unit Test Case for Tenant (Login)

Test Case No	08	**User**	Tenant
Module	View Announcement		
Pre-condition	The tenant log in successful. The tenant taps announcement button on the bottom navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View announcement list	-	The announcement list is displayed.	**(pass)**
View announcement detail.	Tap selected announcement	The selected announcement will be expanded and show the detail.	**(pass)**

Table 7.8: Unit Test Case for Tenant (View announcement)

Test Case No	09	**User**	Tenant
Module	Payment		
Pre-condition	The tenant log in successful.		
	The tenant taps bill button on the bottom navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View unpaid bill list	-	The unpaid bill list is displayed.	**(pass)**
Pay the bill correctly	Tap selected unpaid bill	The payment is recorded in the system	**(pass)**
	Follow the third-party payment gateway's instruction		
Pay the bill incorrectly	Tap selected unpaid bill	An error message is displayed, the payment is not recorded in the system	**(pass)**
	Did not follow the third-party payment gateway's instruction		
Cancel the payment process	Tap selected unpaid bill	An error message is displayed, the payment is not recorded in the system	**(pass)**
	Cancel the process		

Table 7.9: Unit Test Case for Tenant (Payment)

Test Case No	10	**User**	Tenant
Module	Booking		
Pre-condition	The tenant log in successful.		
	The tenant taps reservation button on the bottom navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View the booking record	-	The booking record list is displayed.	**(pass)**
Get the available time	Tap the floating add button	The available timeslot is listed on the dropdown	**(pass)**

slot	Select a date and facility Tap the check timeslot button	list	
Make new reservation	Tap the floating add button Select a date, facility and available timeslot Tap booking button	The booking is recorded successful	**(pass)**

Table 7.10: Unit Test Case for Tenant (Booking announcement)

Test Case No	11	**User**	Tenant
Module	Problem Report		
Pre-condition	The tenant log in successful. The tenant taps problem button on the bottom navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View the problem report list	-	The problem report list is displayed.	**(pass)**
View detail of selected problem report	Tap a problem record Select a date and facility	Redirect to new page with the detail of problem report	**(pass)**
Create new problem report	Tap the floating add button Enter the title, description and upload an image. Click submit button	The problem report is created successful	**(pass)**

Submit new report without title or description	Tap the floating add button Upload an image Click submit button	An error message is displayed	**(pass)**
Submit new report without uploading an image	Tap the floating add button Enter title and description Click submit button	An error message is displayed	**(pass)**

Table 7.11: Unit Test Case for Tenant (Problem report)

Test Case No	12	**User**	Tenant
Module	Profile Management		
Pre-condition	The tenant log in successful.		
	The tenant taps account button on the bottom navigation bar.		
Test Case	**Test Steps**	**Expected Result**	**Status(P/F)**
View profile information	-	The information of tenant account is displayed	**(pass)**
Old password validation before change password	Click the change password button Enter wrong old pass Enter new password and confirm password with same value Click save change button	Login fail with an error message.	**(pass)**
New password validation before change password	Click the change password button Enter correct old password	Login fail with an error message.	**(pass)**

	Enter new password and confirm password with different value		
	Click save change button		
Change password	Click the change password button	The password of the account is updated successful.	**(pass)**
	Enter correct old password		
	Enter new password and confirm password with same value		
	Click save change button		

Table 7.12: Unit Test Case for Tenant (Profile management)

7.3 Backend Testing (API)

Postman software is used to ensure the API is work as expectation.

API route	/api/login
Description	Send username and password to verify the tenant account.
Type	post
Body	username: Tenant password: 12341234
Expected Result	return response true and allow user to login
Status (Pass/Fail)	Pass

Table 7.13: Backend Test Case (Login)

API route	/api/getAnnouncement
Description	Get the announcement list
Type	Get
Body	-
Expected Result	Return announcement list in json format.
Status (Pass/Fail)	Pass

Table 7.14: Backend Test Case (Get Announcement)

API route	/api/getBill
Description	Get the tenants' unpaid bill list
Type	post
Body	tenant_id:1
Expected Result	Return the selected user's unpaid bills.
Status (Pass/Fail)	Pass

Table 7.15: Backend Test Case (Get Bill)

API route	/api/payBill
Description	After completing the payment in third party api, record the payment in system.
Type	post
Body	id:12 (bill id)
Expected Result	The payment is recorded.
Status (Pass/Fail)	Pass

Table 7.16: Backend Test Case (Pay Bill)

API route	/api/getBooking
Description	Get tenant's booking record
Type	post
Body	tenant_id: 1
Expected Result	Return the selected user's booking record.
Status (Pass/Fail)	Pass

Table 7.17: Backend Test Case (Get Booking)

API route	/api/storeBooking
Description	Record new booking request in system
Type	Post
Body	Facility: 1 (id) Timeslot: 2 (id) Tenant_id: 1 Thedate: 30-05-2020
Expected Result	The booking request is recorded successful.
Status (Pass/Fail)	Pass

Table 7.18: Backend Test Case (Store Booking)

API route	/api/getFacilityList
Description	Get facility list
Type	Get
Body	-
Expected Result	Return the available facility list
Status (Pass/Fail)	Pass

Table 7.19: Backend Test Case (Get Facility List)

API route	/api/getAvailableTimeSlot
Description	After selecting facility and date, return the available timeslot
Type	Post
Body	Date: 30-05-2020 Facility: 1 (id)
Expected Result	Return available timeslot in selected condition
Status (Pass/Fail)	Pass

Table 7.20: Backend Test Case (Get Available Timeslot)

API route	/api/getWorkOrder
Description	Get the tenant's work order list.
Type	Post
Body	Tenant_id:1
Expected Result	Return the tenant's work order list
Status (Pass/Fail)	Pass

Table 7.21: Backend Test Case (Get WorkOrder)

API route	/api/storeWorkOrder
Description	Save the ne work order from tenant.
Type	post
Body	Title: Testing Description : Testing for description Category_id: 1 Tenant_id: 1 Image: [base64 format code]
Expected Result	The work order is recorded successful.
Status (Pass/Fail)	Pass

Table 7.22: Backend Test Case (Store WorkOrder)

API route	/api/getWorkOrderDetail
Description	After selecting a workorder, get the detail of the workorder.
Type	post
Body	Id:2 (workorder id)
Expected Result	Return the information of the selected workorder.
Status (Pass/Fail)	Pass

Table 7.23: Backend Test Case (get WorkOrder's detail)

API route	/api/getTenantDetail
Description	Get the information of tenant.
Type	post
Body	Tenant_id:1
Expected Result	Return the selected tenant's information.
Status (Pass/Fail)	Pass

Table 7.24: Backend Test Case (Get Tenant's Detail)

API route	/api/updateTenantDetail
Description	Update the information of the tenant.
Type	Post
Body	Tenant_id: 1 Name: Te Nan Email: Te@nan.com Phone_number: 60123456789
Expected Result	The information of the selected tenant is updated successful.
Status (Pass/Fail)	Pass

Table 7.25: Backend Test Case (Update Tenant's Detail)

API route	/api/updateTenantPassword
Description	Update the password of the tenant
Type	post
Body	Tenant_id :1 Old: 12341234 New: 123412341
Expected Result	The password of the selected tenant is updated successful.
Status (Pass/Fail)	Pass

Table 7.26: Backend Test Case (Update Tenant's Password)

7.4 Integration Testing

After completing the unit testing, all modules will be integrated and tested.

Test Case No	I-01
User	Admin
Module	Login
Step	1) Go to Login Page 2) Fill username/email and password 3) Click Login button
Expected Result	Login successfully
Status (Pass/Fail)	Pass

Table 7.27: Integration Test Case for Admin (Login)

Test Case No	I-02
User	Admin
Module	Tenant Management
Step	1) Go to Tenant Page and click + button. 2) Fill the information and tenant. 3) Click create button. 4) Select unit and click view button. 5) Click delete button. 6) Select the same unit and click view button.
Expected Result	Tenant created successfully. Able to view the tenant account. Tenant account deleted successfully.
Status (Pass/Fail)	Pass

Table 7.28: Integration Test Case for Admin (Tenant Management)

Test Case No	I-03
User	Admin
Module	Announcement Management
Step	1) Go to Tenant Page and click + button. 2) Fill the information of new announcement. 3) Click Submit button
Expected Result	Announcement created successfully. All of the announcement are displayed.
Status (Pass/Fail)	Pass

Table 7.29: Integration Test Case for Admin (Announcement Management)

Test Case No	I-04
User	Admin
Module	Work Order Management
Step	1) Go to work order page. 2) Select a category/ status. 3) Click a work order and click view button. 4) Change status and fill some comment. 5) Click update button. 6) Select the same workorder and click view button.
Expected Result	The information of work order is displayed. The work order is updated successfully.
Status (Pass/Fail)	Pass

Table 7.30: Integration Test Case for Admin (Work Order Management)

Test Case No	I-05
User	Admin
Module	Bill Management
Step	1) Go to Tenant Page. 2) Select a building and unit. 3) Click bill button. 4) Add new bill. 5) Click payment button.

	6) Select the category and bill. 7) Click pay button. 8) Go to Bill record Page.
Expected Result	Bill is added successfully. Payment is recorded successfully. The bill record is able to display correctly.
Status (Pass/Fail)	Pass

Table 7.31: Integration Test Case for Admin (Bill Management)

Test Case No	I-06
User	Admin
Module	Reservation Management
Step	1) Go to booking page. 2) Select date and facility. 3) Click view button.
Expected Result	Booking record is displayed correctly.
Status (Pass/Fail)	Pass

Table 7.32: Integration Test Case for Admin (Reservation Management)

Test Case No	I-07
User	Tenant
Module	Login
Step	1) Open application. 2) Fill username and password 3) Click Login button
Expected Result	Login successfully
Status (Pass/Fail)	Pass

Table 7.33: Integration Test Case for Tenant (Login)

Test Case No	I-08
User	Tenant
Module	Announcement
Step	1) Go to Announcement Page.

	2) Tap one announcement.
Expected Result	Announcement list is displayed. Detail of selected announcement is displayed.
Status (Pass/Fail)	Pass

Table 7.34: Integration Test Case for Tenant (Announcement)

Test Case No	I-09
User	Tenant
Module	Bill
Step	1) Go to Bill Page. 2) Select a bill 3) Proceed payment.
Expected Result	Bill list is displayed. After completing payment, the payment is recorded successfully.
Status (Pass/Fail)	Pass

Table 7.35: Integration Test Case for Tenant (Bill)

Test Case No	I-10
User	Tenant
Module	Booking
Step	1) Go to Booking Page. 2) Tap floating add button. 3) Select date, facility timeslot. 4) Tap booking button.
Expected Result	Booking list is displayed. Booking is recorded successfully.
Status (Pass/Fail)	Pass

Table 7.36: Integration Test Case for Tenant (Booking)

Test Case No	I-11
User	Tenant
Module	Problem Report
Step	1) Go to Problem Page.

	2) Tap floating add button. 3) Fill in the information. 4) Tap Submit button. 5) Tap the new added problem.
Expected Result	Problem list is displayed. Problem is recorded successfully. Information of selected problem is displayed.
Status (Pass/Fail)	Pass

Table 7.37: Integration Test Case for Tenant (Problem Report)

Test Case No	I-12
User	Tenant
Module	Profile
Step	1) Go to account Page. 2) Tap edit icon button 3) Edit the information. 4) Tap save button. 5) Tap change password button. 6) Fill in new password, old password and re-enter password 7) Tap save change button.
Expected Result	Information of the account is displayed. Information of the account is updated successful. Password of account is updated successful.
Status (Pass/Fail)	Pass

Table 7.38: Integration Test Case for Tenant (Profile)

7.5 User Acceptance Test

There are 7 UTAR students are invited to run through the project's system to verify the workflow of the system. Due to the limitation of movement control order (MCO) by the Malaysia Government, the management team in Evergreen Scot Pine cannot involve this testing unfortunately.

7.5.1 UAT Test Case

7.5.1.1 Web System

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	
Comment (if any)	

Table 7.393: Test Case for Web System (Login and Logout)

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	
Comment (if any)	

Table 7.40: Test Case for Web System (Tenant and Unit)

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	
Comment (if any)	

Table 7.41: Test Case for Web System (Bill)

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	
Comment (if any)	

Table 7.42: Test Case for Web System (Work order)

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	
Comment (if any)	

Table 7.43: Test Case for Web System (Announcement)

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	
Comment (if any)	

Table 7.44: Test Case for Web System (Booking)

7.5.1.2 Mobile Appilcation

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	
Comment (if any)	

Table 7.45: Test Case for mobile application (Authentication)

Scenario	1. You would like to view the announcement.
Result	
Comment (if any)	

Table 7.46: Test Case for mobile application (Announcement)

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	
Comment (if any)	

Table 7.47: Test Case for mobile application (Bill)

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	
Comment (if any)	

Table 7.48: Test Case for mobile application (Reservation)

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	
Comment (if any)	

Table 7.49: Test Case for mobile application (Problem)

7.5.2 UAT Test Result

All end users are able to understand the UI of system and run through the scenario. Hence, the result proves that most of the features of the system are workable and achieved the requirement.

7.5.2.1 Web System

Test Functionality	Number of test conduct	Number of pass result	Comment
Login and Logout	7	7	
Tenant and Unit	7	7	- suggest to add header/title after user select to let user know user is selecting the correct unit
Bill	7	7	-view bill record can be integrated in same page
Work Order	7	7	
Announcement	7	7	
Booking	7	7	

Table 7.50: UAT Result for web system

7.5.2.2 Mobile Application

Test Functionality	Number of test conduct	Number of pass result	Comment
Authentication	7	7	
Announcement	7	7	
Bill	7	7	
Reservation	7	7	-The time slot checking could be interactive. For instance, the word “time slot available” turns green if is it available and turn red for opposite availability.
Problem	7	7	

Table 7.51: UAT Result for mobile application

7.6 Usability Test

After conducting UAT, the testers are requested to fill another survey form (usability test form) as it is also important to evaluate the end user experience with the system.

7.6.1 Usability Test Form

7.6.1.1 Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?					
2. Did you think that this system is unnecessarily complex?					
3. Did you think that this system is easy to use?					
4. Did you need someone to assist when using this system?					
5. Did you think that this system is easy to navigate?					
6. Did you think that this system has too much inconsistency?					
7. Did you think that most of the users can learn to use this system very quickly?					
8. Did you think that this system is very cumbersome/awkward to use?					
9. Did you felt very confident using this system?					
10. Did you need to learn a lot of things before you start to use this system?					
Comment (if any):					

Table 7.52: Usability Test Form for Web System

7.6.1.2 Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?					
2. Did you think that this mobile application is unnecessarily complex?					
3. Did you think that this mobile application is easy to use?					
4. Did you need someone to assist when using this mobile application?					
5. Did you think that this mobile application is easy to navigate?					
6. Did you think that this mobile application has too much inconsistency?					
7. Did you think that most of the users can learn to use this mobile application very quickly?					
8. Did you think this mobile application is very cumbersome/awkward to use?					
9. Did you felt very confident using this mobile application?					
10. Did you need to learn a lot of things before you start to use this mobile application?					
Comment (if any):					

Table 7.53: Usability Test From for Mobile Application

7.6.2 Usability Test Result

The System Usability Scale (SUS) is referenced as the scoring system for usability test result. The formula of SUS will be shown as below:

1. Odd question (1,3,5….): Response - 1
2. Even question (2,4,6…): 5 –Response
3. Calculate total score and multiply by 2.5 to get range of values from 0 – 100.

7.6.2.1 Web System

Question	Tester							
	1	2	3	4	5	6	7	Average
1	3	2	3	4	2	3	3	2.86
2	4	4	3	4	3	3	3	3.43
3	3	2	3	3	3	3	3	2.86
4	2	2	2	2	3	3	3	2.43
5	4	2	3	4	3	3	3	3.14
6	4	4	3	4	2	4	2	3.29
7	3	3	4	4	3	3	3	3.29
8	3	3	3	4	3	3	2	3.00
9	3	2	3	4	3	3	3	3.00
10	3	3	4	4	4	3	3	3.43
Total	32	27	31	37	29	31	28	30.71
SUS score	80	67.5	77.5	92.5	72.5	77.5	70	76.79

Table 7.54: Usability Test Result for Web System

7.6.2.2 Mobile Application

Question	Tester							
	1	2	3	4	5	6	7	Average
1	3	2	3	4	2	3	3	2.86
2	4	3	3	4	3	3	4	3.43
3	3	3	4	4	3	3	3	3.29
4	3	2	3	4	3	3	3	3.00
5	4	3	4	4	3	3	3	3.43
6	4	4	3	4	4	4	4	3.86
7	4	2	4	4	4	3	3	3.43
8	4	3	4	4	3	3	3	3.43
9	3	2	4	4	3	3	3	3.14
10	3	2	3	4	4	3	3	3.14
Total	35	26	35	40	32	31	32	33.00
SUS score	87.5	65	87.5	100	80	77.5	80	82.50

Table 7.55: Usability Test Result for Mobile Application

7.6.2.3 Overall

Both of web system and mobile application get a very good score in the sus. According to the result, the grade of web system is B while the grade of mobile application is A. The referenced standard grade of sus will be shown as figure below:

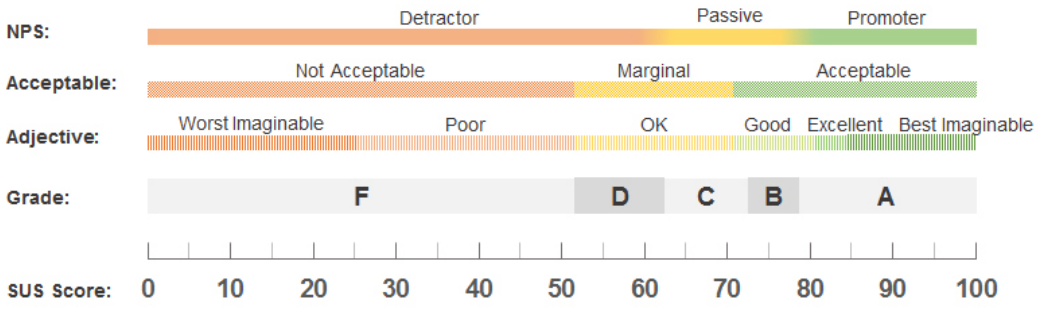


Figure 7.1: Evaluation with the SUS scores. (Sauro, 2018)

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

This project is proposed due the traditional way of managing condo property are not effective. In order to solve the problem by using the Information Technology, this project has been started more than half year, from planning to software testing.

It is important to prepare a good planning and design of system, as it can lower the development time and cost. Besides, the Rapid Application Development software methodology also provides this project a better risk control.

There are two adopted frameworks in this project, which is Laravel and Flutter. By using the Laravel, the backend of system and API are able to develop and write in PHP Language rapidly. On the other hand, the bootstrap library and Flutter are used in developing the front end of web system and mobile app respectively.

After conducting a series of software testing, it can ensure that the system can help condo management team to maintain condo and communicate with the tenants in more convenient way. The tenant also can get the latest information, book a reservation, report problem and pay the bill via the mobile application.

All of the maintenance problem can be upload via the mobile application, while the web system can update the progress. Both platforms are able to view the maintenance problems anytime. Besides, the web system can assist management team to add new bill, the mobile application allows the tenant to pay the bill. In the mobile application, user also can view the announcement or other new notification.

In conclusion, the web system and mobile application also are delivered at the end phase of the SDLC. The project also achieves all of the objective of the project:

- To provide an efficient solution to follow up on the progress of the maintenance problem without any paperwork.

- To provide an integrated solution to accept and respond to the maintenance requests and facility bookings.
- To allow property the tenants to view the billing statement and pay the payment via mobile application.
- To allow property owners and tenants to receive instant notification and status updates.

.

8.2 Recommendations for future work

Although the project is fulfilled all the requirements and get a lot of positive feedback from the end user, the system still a lot of room for improvement. There are some suggestion can be considers as future enhancement to improve the quality of the web system and mobile application：

- Implement push notification system with real-time database.
- Implement visitor management feature.
- Implement document system to generate e-bill statement.
- Implement more interactive animation in the mobile app.
- Allow tenant to upload more image for a work order.

REFERENCES

Anderson, K., 2017. What is Rapid Application Development and When Should You Use It? Capterra blog, [blog] 4 October 2017. Available at: <https://blog.capterra.com/ what-is-rapid-application-development/> [Accessed on 30 June 2019].

Borowski, C., 2015. Become a Paperless Office With Document Management Software Industry View. Software Advice, [online] 4 September 2015. Available at: <https://www.softwareadvice.com/cms/industryview/go-paperless-with-document-management-software/> [Accessed on 2 July 2019].

Geambasu, C. V., Jianu, I., Jianu, I., Gavrila, A., 2011. INFLUENCE FACTORS FOR THE CHOICE OF A SOFTWARE DEVELOPMENT METHODOLOGY. The Bucharest Academy of Economic Studies, Romania. Accounting and Management Information Systems, 10(4), pp.479–494.

Hanna, K. (2007). Adsorption of aromatic carboxylate compounds on the surface of synthesized iron oxide-coated sands. Applied Geochemistry, 22, 2045-2053.

IncludeHelp (n.t.). Difference between DBMS and Traditional File System. [online] Includehelp.com. Available at: <https://www.includehelp.com/dbms/dbms-vs-traditional-file-system.aspx/> [Accessed 26 Jul. 2019].

Johnston, G.P. and Bowen, D. V., 2005. The benefits of electronic records management systems: A general review of published and some unpublished cases. Records Management Journal, 15(3), pp.131–140.

Lotz, M. (2018). Waterfall vs. Agile: Which Methodology is Right for Your Project?. [online] Segue Technologies. Available at: https://www.seguetech.com/waterfall-vs-agile-methodology/ [Accessed 28 Jul. 2019].

Otwell, T., 2019. Introduction - Laravel - The PHP Framework For Web Artisans. [online] Available at <https://laravel.com/docs/4.2/introduction> [Accessed on 12 July 2019]

Sauro, J., 2018. Measuringu: 5 Ways To Interpret A SUS Score. [online] Measuringu.com. Available at: <https://measuringu.com/interpret-sus-score/> [Accessed 30 February 2020].

Shaydulin, R. and Sybrandt, J., 2017. To Agile, or not to Agile: A Comparison of Software Development Methodologies. [online] pp.1–11. Available at: <http://arxiv.org/abs/1704.07469>.

Soni, D. and Kohli, P.J., 2017. Cost Estimation Model for Web Applications using Agile Software Development Methodology. Pertanika Journal of Science and Technology, 25(3), pp.931–938.
Wallen, J., 2019. What is MySQL. Lifewire, [online] 19 March 2019. Available at: <https://www.lifewire.com/what-is-mysql-4582965> [Accessed on 14 July 2019]

Zhang, B., Cohen, J., Ferrence, R., &Rehm, J. (2006). The impact of tobacco tax cuts on smoking initiation among Canadian young adults. *American Journal of Preventive Medicine, 30*, 474-479.

APPENDICES

APPENDIX A: Interview Result

Interview Form

Name of Management Team: Ms. Lalla
Location: Evergreen Park Scot Pine
Objective: To enquire about how the condo management team handle the tasks
Feedbacks:

1. Can tenant request any work order without being in the office?

   The tenants will be added in a WhatsApp group. So, they can ask the management team to fix or solute problem via Whatsapp.

2. How do management team send an announcement to all tenants?

   We will post the announcement letter on all the board in the condo. However, the announcement is too easy to be ignored by the tenants.

3. Can tenant pay bill without being in the office?

   No, although they can online banking transfer to our management team bank account, they still need to come to office to show the invoice

APPENDIX B: User Acceptance Test Form

User Acceptance Test

Participant: CHAN KANG NENG

Testing Date: 21/4/2020

Web System

1. Test Functionality: Login and Logout

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	**No issue**
Comment (if any)	

2. Test Functionality: Tenant and Unit

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	**No issue**
Comment (if any)	

6. Test Functionality: Booking

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	**No issue**
Comment (if any)	

Mobile Application

1. Test Functionality: Authentication

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	**No issue**
Comment (if any)	

2. Test Functionality: Announcement

Scenario	1. You would like to view the announcement.
Result	**No issue**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	**No issue**
Comment (if any)	

4. Test Functionality: Work order

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	**No issue**
Comment (if any)	

5. Test Functionality: Announcement

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	**No issue**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	**No issue**
Comment (if any)	

4. Test Functionality: Reservation

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	**Little confusion**
Comment (if any)	**-The time slot checking could be interactive. For instance, the word "time slot available" turns green if is it available and turn red for opposite availability.**

5. Test Functionality: Problem

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	**No issue**
Comment (if any)	

User Acceptance Test

Participant: CHOW YUET MEI

Testing Date: 21/4/2020

Web System

1. Test Functionality: Login and Logout

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	**working**
Comment (if any)	

2. Test Functionality: Tenant and Unit

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	**working**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	**working**
Comment (if any)	

4. Test Functionality: Work order

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	**working**
Comment (if any)	

5. Test Functionality: Announcement

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	**working**
Comment (if any)	

6. Test Functionality: Booking

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	**working**
Comment (if any)	

Mobile Application

1. Test Functionality: Authentication

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	**working**
Comment (if any)	

2. Test Functionality: Announcement

Scenario	1. You would like to view the announcement.
Result	**working**
Comment (if any)	

User Acceptance Test

Participant: EDWARD GO CHEE ERN

Testing Date: 21/4/2020

Web System

1. Test Functionality: Login and Logout

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	**working**
Comment (if any)	

2. Test Functionality: Tenant and Unit

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	**working**
Comment (if any)	**suggest to add header/title after user select to let user know user is selecting the correct unit**

6. Test Functionality: Booking

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	**working**
Comment (if any)	

Mobile Application

1. Test Functionality: Authentication

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	**working**
Comment (if any)	

2. Test Functionality: Announcement

Scenario	1. You would like to view the announcement.
Result	**working**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	**working**
Comment (if any)	

4. Test Functionality: Reservation

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	**working**
Comment (if any)	

5. Test Functionality: Problem

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	**working**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	**working**
Comment (if any)	

4. Test Functionality: Work order

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	**working**
Comment (if any)	

5. Test Functionality: Announcement

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	**working**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	**working**
Comment (if any)	

4. Test Functionality: Reservation

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	**working**
Comment (if any)	

5. Test Functionality: Problem

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	**working**
Comment (if any)	

User Acceptance Test

Participant: TAN JIASHENG

Testing Date: 22/4/2020

Web System

1. **Test Functionality: Login and Logout**

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	**working**
Comment (if any)	

2. **Test Functionality: Tenant and Unit**

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	**working**
Comment (if any)	

6. **Test Functionality: Booking**

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	**working**
Comment (if any)	

Mobile Application

1. **Test Functionality: Authentication**

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	**working**
Comment (if any)	

2. **Test Functionality: Announcement**

Scenario	1. You would like to view the announcement.
Result	**working**
Comment (if any)	

User Acceptance Test

Participant:Eric Low Zhao Lun

Testing Date:22/04/2020

Web System

1. **Test Functionality: Login and Logout**

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	**ok**
Comment (if any)	

2. **Test Functionality: Tenant and Unit**

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	**ok**
Comment (if any)	

3. **Test Functionality: Bill**

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	**working**
Comment (if any)	

4. **Test Functionality: Work order**

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	**working**
Comment (if any)	

5. **Test Functionality: Announcement**

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	**workiing**
Comment (if any)	

3. **Test Functionality: Bill**

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	**working**
Comment (if any)	

4. **Test Functionality: Reservation**

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	**working**
Comment (if any)	

5. **Test Functionality: Report**

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	**working**
Comment (if any)	

3. **Test Functionality: Bill**

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	**ok**
Comment (if any)	**view bill record can be integrate in same page**

4. **Test Functionality: Work order**

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	**ok**
Comment (if any)	

5. **Test Functionality: Announcement**

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	**ok**
Comment (if any)	

6. Test Functionality: Booking

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	**ok**
Comment (if any)	

Mobile Application

1. Test Functionality: Authentication

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	**ok**
Comment (if any)	

2. Test Functionality: Announcement

Scenario	1. You would like to view the announcement.
Result	**ok**
Comment (if any)	

User Acceptance Test

Participant: LAI KAI WEN

Testing Date: 22/4/20

Web System

1. Test Functionality: Login and Logout

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	**No issue**
Comment (if any)	

2. Test Functionality: Tenant and Unit

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	**No issue**
Comment (if any)	

6. Test Functionality: Booking

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	**No issue**
Comment (if any)	

Mobile Application

1. Test Functionality: Authentication

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	**No issue**
Comment (if any)	

2. Test Functionality: Announcement

Scenario	1. You would like to view the announcement.
Result	**No issue**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	**ok**
Comment (if any)	

4. Test Functionality: Reservation

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	**ok**
Comment (if any)	

5. Test Functionality: Problem

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	**ok**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	**No issue**
Comment (if any)	

4. Test Functionality: Work order

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	**No issue**
Comment (if any)	

5. Test Functionality: Announcement

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	**No issue**
Comment (if any)	

3. Test Functionality: Bill

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	**No issue**
Comment (if any)	

4. Test Functionality: Reservation

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	**No issue**
Comment (if any)	

5. Test Functionality: Problem

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	**No issue**
Comment (if any)	

User Acceptance Test

Participant: Kum Wai Chung

Testing Date: 21/04/2020

Web System

1. **Test Functionality: Login and Logout**

Scenario	1. You would like to login to the system. Username : admin Password : testtest 2. You would like to log out the system
Result	**pass**
Comment (if any)	-

2. **Test Functionality: Tenant and Unit**

Scenario	1. Tenant (A1-1-2) did not have an account yet. He would like to register new account. a. Create a new tenant account for unit A1-1-2. 2. Tenant (A1-1-1) would like to view the account's information. a. View the information of the unit A1-1-1.
Result	**pass**
Comment (if any)	-

6. **Test Functionality: Booking**

Scenario	1. You would like view who booked the tennis table. Example: a. Date: 23/4/2020 b. Facility: Tennis Table 1
Result	**pass**
Comment (if any)	-

Mobile Application

1. **Test Functionality: Authentication**

Scenario	1. You would like to login to the system. Username : tenant Password : 12341234 2. You would like to change the profile. 3. You would like to change password. 4. You would like to log out the system.
Result	**pass**
Comment (if any)	-

2. **Test Functionality: Announcement**

Scenario	1. You would like to view the announcement.
Result	**pass**
Comment (if any)	-

3. **Test Functionality: Bill**

Scenario	1. New bill is released. a. Add new bill for unit C1-5-3. 2. Tenant (C1-5-3) walk-in to the office and pay the bill. a. Record the payment. 3. You would like to view the bill record of unit C1-5-3.
Result	**pass**
Comment (if any)	-

4. **Test Functionality: Work order**

Scenario	1. The work order is needed to update. a. Select a category or status. b. Find the work order and view the detail. c. Update the status and leave some comment.
Result	**pass**
Comment (if any)	-

5. **Test Functionality: Announcement**

Scenario	1. You would like to announce that food delivery is not allowed to go inside. 2. You would like to delete an announcement
Result	**pass**
Comment (if any)	-

3. **Test Functionality: Bill**

Scenario	1. You would like to view the unpaid bill. 2. You would like to pay the bill. a. Select card method. b. Phone number: +60123456789 c. Email: te@nant.com d. Card Number: 4111 1111 1111 1111 e. Expired Date : 12/23 f. Card Holder: Tenant g. CVV: 091
Result	**pass**
Comment (if any)	-

4. **Test Functionality: Reservation**

Scenario	1. You would like to view the reservation that you had requested. 2. You would like to make new reservation.
Result	**pass**
Comment (if any)	-

5. **Test Functionality: Problem**

Scenario	1. You would like to view the progress of the problem report. 2. You would like to submit a new report with image.
Result	**pass**
Comment (if any)	-

APPENDIX C: Usability Test Form

System Usability Scale (SUS)

Participant: Kum Wai Chung

Date: 21/04/2020

For each of the following statements, please mark **one** box that best describes your reactions.

Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?				y	
2. Did you think that this system is unnecessarily complex?	y				
3. Did you think that this system is easy to use?				y	
4. Did you need someone to assist when using this system?			y		
5. Did you think that this system is easy to navigate?					y
6. Did you think that this system has too much inconsistency?	y				
7. Did you think that most of the users can learn to use this system very quickly?				y	
8. Did you think that this system is very cumbersome/awkward to use?		y			
9. Did you felt very confident using this system?				y	
10. Did you need to learn a lot of things before you start to use this system?		y			
Comment (if any): No comment					

Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?				y	
2. Did you think that this mobile application is unnecessarily complex?	y				
3. Did you think that this mobile application is easy to use?				y	
4. Did you need someone to assist when using this mobile application?		y			
5. Did you think that this mobile application is easy to navigate?					y
6. Did you think that this mobile application has too much inconsistency?	y				
7. Did you think that most of the users can learn to use this mobile application very quickly?					y
8. Did you think this mobile application is very cumbersome/awkward to use?	y				
9. Did you felt very confident using this mobile application?				y	
10. Did you need to learn a lot of things before you start to use this mobile application?		y			
Comment (if any): Nice and clean interface					

System Usability Scale (SUS)

Participant: CHAN KANG NENG

Date: 21/4/2020

For each of the following statements, please mark **one** box that best describes your reactions.

Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?			/		
2. Did you think that this system is unnecessarily complex?	/				
3. Did you think that this system is easy to use?			/		
4. Did you need someone to assist when using this system?			/		
5. Did you think that this system is easy to navigate?			/		
6. Did you think that this system has too much inconsistency?	/				
7. Did you think that most of the users can learn to use this system very quickly?				/	
8. Did you think that this system is very cumbersome/awkward to use?		/			
9. Did you felt very confident using this system?			/		
10. Did you need to learn a lot of things before you start to use this system?		/			
Comment (if any):					

Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?			/		
2. Did you think that this mobile application is unnecessarily complex?		/			
3. Did you think that this mobile application is easy to use?				/	
4. Did you need someone to assist when using this mobile application?			/		
5. Did you think that this mobile application is easy to navigate?				/	
6. Did you think that this mobile application has too much inconsistency?	/				
7. Did you think that most of the users can learn to use this mobile application very quickly?			/		
8. Did you think this mobile application is very cumbersome/awkward to use?		/			
9. Did you felt very confident using this mobile application?			/		
10. Did you need to learn a lot of things before you start to use this mobile application?			/		
Comment (if any):					

System Usability Scale (SUS)

Participant: CHOW YUET MEI

Date: 21/4/2020

For each of the following statements, please mark **one** box that best describes your reactions.

Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?				√	
2. Did you think that this system is unnecessarily complex?		√			
3. Did you think that this system is easy to use?				√	
4. Did you need someone to assist when using this system?			√		
5. Did you think that this system is easy to navigate?				√	
6. Did you think that this system has too much inconsistency?		√			
7. Did you think that most of the users can learn to use this system very quickly?					√
8. Did you think that this system is very cumbersome/awkward to use?		√			
9. Did you felt very confident using this system?				√	
10. Did you need to learn a lot of things before you start to use this system?	√				
Comment (if any):					

Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?				√	
2. Did you think that this mobile application is unnecessarily complex?		√			
3. Did you think that this mobile application is easy to use?					√
4. Did you need someone to assist when using this mobile application?		√			
5. Did you think that this mobile application is easy to navigate?					√
6. Did you think that this mobile application has too much inconsistency?		√			
7. Did you think that most of the users can learn to use this mobile application very quickly?					√
8. Did you think this mobile application is very cumbersome/awkward to use?	√				
9. Did you felt very confident using this mobile application?					√
10. Did you need to learn a lot of things before you start to use this mobile application?		√			
Comment (if any):					

System Usability Scale (SUS)

Participant: EDWARD GO CHEE ERN

Date: 21/4/2020

For each of the following statements, please mark **one** box that best describes your reactions.

Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?					©
2. Did you think that this system is unnecessarily complex?	©				
3. Did you think that this system is easy to use?				©	
4. Did you need someone to assist when using this system?			©		
5. Did you think that this system is easy to navigate?					©
6. Did you think that this system has too much inconsistency?	©				
7. Did you think that most of the users can learn to use this system very quickly?					©
8. Did you think that this system is very cumbersome/awkward to use?	©				
9. Did you felt very confident using this system?					©
10. Did you need to learn a lot of things before you start to use this system?	©				
Comment (if any): user-friendly layout and easy to understand					

Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?					©
2. Did you think that this mobile application is unnecessarily complex?	©				
3. Did you think that this mobile application is easy to use?					©
4. Did you need someone to assist when using this mobile application?	©				
5. Did you think that this mobile application is easy to navigate?					©
6. Did you think that this mobile application has too much inconsistency?	©				
7. Did you think that most of the users can learn to use this mobile application very quickly?					©
8. Did you think this mobile application is very cumbersome/awkward to use?	©				
9. Did you felt very confident using this mobile application?					©
10. Did you need to learn a lot of things before you start to use this mobile application?	©				
Comment (if any): easy to use, well presented					

System Usability Scale (SUS)

Participant: TAN JIASHENG

Date:22/4/2020

For each of the following statements, please mark **one** box that best describes your reactions.

Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?			X		
2. Did you think that this system is unnecessarily complex?		X			
3. Did you think that this system is easy to use?				X	
4. Did you need someone to assist when using this system?		X			
5. Did you think that this system is easy to navigate?				X	
6. Did you think that this system has too much inconsistency?			X		
7. Did you think that most of the users can learn to use this system very quickly?				X	
8. Did you think that this system is very cumbersome/awkward to use?		X			
9. Did you felt very confident using this system?				X	
10. Did you need to learn a lot of things before you start to use this system?	X				
Comment (if any):					

Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?			X		
2. Did you think that this mobile application is unnecessarily complex?		X			
3. Did you think that this mobile application is easy to use?				X	
4. Did you need someone to assist when using this mobile application?		X			
5. Did you think that this mobile application is easy to navigate?				X	
6. Did you think that this mobile application has too much inconsistency?	X				
7. Did you think that most of the users can learn to use this mobile application very quickly?					X
8. Did you think this mobile application is very cumbersome/awkward to use?		X			
9. Did you felt very confident using this mobile application?				X	
10. Did you need to learn a lot of things before you start to use this mobile application?	X				
Comment (if any):					

System Usability Scale (SUS)

Participant: Eric Low Zhao Lun

Date: 22/04/2020

For each of the following statements, please mark **one** box that best describes your reactions.

Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?				/	
2. Did you think that this system is unnecessarily complex?		/			
3. Did you think that this system is easy to use?				/	
4. Did you need someone to assist when using this system?		/			
5. Did you think that this system is easy to navigate?				/	
6. Did you think that this system has too much inconsistency?	/				
7. Did you think that most of the users can learn to use this system very quickly?				/	
8. Did you think that this system is very cumbersome/awkward to use?		/			
9. Did you felt very confident using this system?				/	
10. Did you need to learn a lot of things before you start to use this system?		/			
Comment (if any):					

Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?				/	
2. Did you think that this mobile application is unnecessarily complex?		/			
3. Did you think that this mobile application is easy to use?				/	
4. Did you need someone to assist when using this mobile application?		/			
5. Did you think that this mobile application is easy to navigate?				/	
6. Did you think that this mobile application has too much inconsistency?	/				
7. Did you think that most of the users can learn to use this mobile application very quickly?				/	
8. Did you think this mobile application is very cumbersome/awkward to use?		/			
9. Did you felt very confident using this mobile application?				/	
10. Did you need to learn a lot of things before you start to use this mobile application?		/			
Comment (if any):					

System Usability Scale (SUS)

Participant: LAI KAI WEN

Date: 22/4/20

For each of the following statements, please mark **one** box that best describes your reactions.

Web System

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this system frequently?				✕	
2. Did you think that this system is unnecessarily complex?		✕			
3. Did you think that this system is easy to use?				✕	
4. Did you need someone to assist when using this system?		✕			
5. Did you think that this system is easy to navigate?				✕	
6. Did you think that this system has too much inconsistency?			✕		
7. Did you think that most of the users can learn to use this system very quickly?				✕	
8. Did you think that this system is very cumbersome/awkward to use?			✕		
9. Did you felt very confident using this system?				✕	
10. Did you need to learn a lot of things before you start to use this system?		✕			
Comment (if any):					

Mobile Application

	1 Strongly Disagree	2	3	4	5 Strongly Agree
1. Did you like to use this mobile application frequently?				✕	
2. Did you think that this mobile application is unnecessarily complex?	✕				
3. Did you think that this mobile application is easy to use?				✕	
4. Did you need someone to assist when using this mobile application?		✕			
5. Did you think that this mobile application is easy to navigate?				✕	
6. Did you think that this mobile application has too much inconsistency?	✕				
7. Did you think that most of the users can learn to use this mobile application very quickly?				✕	
8. Did you think this mobile application is very cumbersome/awkward to use?		✕			
9. Did you felt very confident using this mobile application?				✕	
10. Did you need to learn a lot of things before you start to use this mobile application?		✕			
Comment (if any):					