



**UTeM**

اُنیورسیتی تکنیک ملیسا ملاک  
UNIVERSITI TEKNIKL MALAYSIA MELAKA

**FTMK**

Fakulti Teknologi Maklumat dan Komunikasi

**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**BITU3923  
WORKSHOP 2**

**SEMESTER 1**

**SESSION 2022/2023**

<b>GROUP</b>	2	
<b>PREPARED BY</b>	<b>NAME</b>	<b>MATRIC NUMBER</b>
	TAN WEI MING	B032010060
	FIRZANA HUDA BINTI AZREEL	B032010236
	INTAN FARRA LIYANNA BINTI SUHAIMI	B032010474
	KHALID ALI FARAH	B032020051
<b>SUPERVISOR</b>	TS. MASLITA BINTI ABD AZIZ	
<b>PROJECT TITLE</b>	<b>FINAL REPORT: SMART PARKING SYSTEM</b>	

## **ACKNOWLEDGEMENT**

We are grateful and thankful to the Creator for the time, energy, idea and patience that He gave to us to complete our project of Workshop 2 which is Smart Parking System. The deepest gratitude also to our supervisor, Ts. Maslita binti Abd Aziz for her guidance and helps throughout the project until we are completing our project. We also want to express our appreciation to each of our parents, family, friends and loved ones for the emotional support that they gave to us every time we need it. Thank you to everyone who had gave their support, inspiration and outstanding suggestions. Last but not least, we would like to give the warmest thanks to ourselves for the extraordinary support and cooperation during completing the project. Each one of us has give their best and we are very appreciating everyone's efforts.

## **ABSTRACT**

The number of vehicles is increasing day by day. It is hard to find parking space in workplace or even shopping malls. This has become disadvantages to people especially if the people is in rush. This problem has inspired the team to developed a system called Smart Parking System. Smart Parking System will give pleasure to the people to find the parking space easily. The technology that has been used also increasing the security aspects for their parked vehicles.

## TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION .....	1
1.1 Introduction.....	1
1.2 Problem Statement.....	1
1.3 Objectives .....	1
1.4 Scopes .....	2
1.4.1 Module to be developed.....	2
1.4.2 Target User.....	3
1.5 Project Significance .....	3
1.6 Hardware and Software.....	4
1.7 Conclusion .....	4
CHAPTER 2: SYSTEM METHODOLOGY AND PLANNING .....	5
2.1 Introduction.....	5
2.2 System Development Methodology.....	5
2.3 System Development Techniques.....	8
2.4 Conclusion .....	9
CHAPTER 3: SYSTEM ANALYSIS.....	10
3.1 Introduction.....	10
3.2 Analysis of Current Problem .....	10
3.3 Analysis of Proposed System.....	11
3.3.1 Background.....	11
3.3.2 Functional Requirements .....	13
3.3.3 Non-functional Requirements .....	14
3.4 Structured Chart of Proposed System .....	15
3.5 Work Breakdown .....	16
3.5.1 Task Distribution .....	16
3.6 Gantt Chart.....	17
3.7 Conclusion .....	17
CHAPTER 4: SYSTEM DESIGN.....	18
4.1 Introduction.....	18
4.2 Database Design.....	18
4.2.1 Entity Relationship Diagram (ERD).....	18
4.2.2 Data Dictionary .....	19

4.3 Module Integration and Interface Design .....	25
4.3.1 Smart Parking System Mobile Application .....	25
4.3.2 Smart Parking System Scanning.....	34
4.3.3 Smart Parking System Admin Website.....	35
4.4 Conclusion .....	37
CHAPTER 5: SYSTEM IMPLEMENTATION.....	38
5.1 Introduction.....	38
5.2 System Development Environment .....	38
5.2.1 Android Studio.....	38
5.2.2 Visual Studio Code .....	38
5.2.3 XAMPP.....	38
5.3 System Configuration Management .....	39
5.3.1 Installation and Setup of Android Studio.....	39
5.3.2 Installation and Setup of Visual Studio Code .....	45
5.3.3 Installation and Setup of XAMPP.....	51
5.4 Security Characteristics .....	53
5.5 Output of Implementation.....	54
5.5.1 Smart Parking System Mobile Application .....	54
5.5.2 Admin Website .....	69
5.6 Conclusion .....	70
CHAPTER 6: SYSTEM TESTING.....	71
6.1 Introduction.....	71
6.1.1 Black Box Testing.....	71
6.1.2 White Box Testing .....	71
6.2 Test Result Analysis .....	72
6.2.1 Test Case 1: Successful User Login.....	72
6.2.2 Test Case 2: Unsuccessful User Login .....	72
6.2.3 Test Case 3: Successful Account Registration.....	73
6.2.4 Test Case 4: Unsuccessful Account Registration .....	74
6.2.5 Test Case 5: Create Profile .....	75
6.2.6 Test Case 6: Booking Location Validation.....	76
6.2.7 Test Case 7: Booking Success .....	77
6.2.8 Test Case 8: Successful Vehicle Plate Number First Scanning .....	78

6.2.9 Test Case 9: Unsuccessful Vehicle Number Plate First Scanning.....	79
6.2.10 Test Case 10: Vehicle Plate Number First Scanning Validation .....	80
6.2.11 Test Case 11: Successful Vehicle Plate Number Second Scanning .....	81
6.2.12 Test Case 12: Unsuccessful Vehicle Plate Number Second Scanning .....	81
6.2.13 Test Case 13: Vehicle Plate Number Second Scanning Validation.....	82
6.2.14 Test Case 14: Successful Upcoming QR Code Scanning .....	83
6.2.15 Test Case 15: Unsuccessful Upcoming QR Code Scanning.....	85
6.2.16 Test Case 16: Upcoming QR Code Scanning Validation .....	85
6.2.17 Test Case 17: Current Page (No vehicle is parked) .....	86
6.2.18 Test Case 18: Current Page(Vehicle is parked) .....	86
6.2.19 Test Case 19: Penalty Function.....	87
6.2.20 Test Case 20: Successful Payment.....	88
6.2.21 Test Case 21: Successful Retrieve QR Code Scanning .....	89
6.2.22 Test Case 22: Unsuccessful Retrieve QR Code Scanning .....	90
6.2.23 Test Case 23: Successful Vehicle Plate Number Third Scanning .....	90
6.2.24 Test Case 24: Vehicle Plate Number Third Scanning Validation.....	91
6.2.25 Test Case 25: History Page (No Vehicle is Retrieved).....	92
6.2.26 Test Case 26 History Page(Vehicle is retrieved) .....	93
6.2.27 Test Case 27: Unsuccessful Website Admin Login.....	94
6.2.28 Test Case 28: Successful Website Admin Login .....	95
6.2.29 Test Case 29: Update Price Table .....	95
6.2.30 Test Case 30: History Review (Filter Dates) .....	96
6.2.31 Test Case 31: History Review Table (Payment Date Filter).....	97
6.2.32 Test Case 32: History Review Table (Total filter).....	98
6.2.33 Test Case 33: History Review Table (Status filter) .....	99
6.3 Test Result Analysis .....	100
6.3.1 Performance .....	100
6.3.2 Availability .....	100
6.3.3 Scalability .....	101
6.3.4 Portability.....	101
6.3.5 Maintainability .....	101
6.3.6 Security .....	101
6.4 Conclusion .....	101
CHAPTER 7: CONCLUSION .....	102

7.1 Introduction.....	102
7.2 Achievement .....	102
7.3 Project Limitations.....	102
7.4 Suggestions for Improvement .....	103
7.5 Potential Commercialisation.....	103
7.6 Conclusion .....	103
REFERENCES .....	104

## LIST OF FIGURES

Figure 2.2 Agile Methodology.....	5
Figure 3.2 The collision between two cars while finding a parking spot .....	10
Figure 3.3.1 Flowchart of the whole system.....	11
Figure 3.4 Structured Chart of proposed system .....	15
Figure 3.5.1 Organizational chart .....	16
Figure 3.6 Gantt Chart .....	17
Figure 4.2.1 Entity Relationship Diagram (ERD) .....	18
Figure 4.3.1.1 Account registration page.....	25
Figure 4.3.1.2 Login page .....	26
Figure 4.3.1.3 Create profile page .....	26
Figure 4.3.1.4 Current page .....	27
Figure 4.3.1.5 Current Page with Data .....	27
Figure 4.3.1.6 Upcoming page.....	28
Figure 4.3.1.7 Upcoming page with data.....	28
Figure 4.3.1.8 History page.....	29
Figure 4.3.1.9 History page with data.....	29
Figure 4.3.1.10 Google Maps page.....	30
Figure 4.3.1.11 Booking page.....	30
Figure 4.3.1.12 Booking requirements page .....	31
Figure 4.3.1.13 Booking details page .....	31
Figure 4.3.1.14 Retrieve page .....	32
Figure 4.3.1.15 Retrieve page with data .....	32
Figure 4.3.1.16 Payment details.....	33
Figure 4.3.1.17 Payment page.....	33
Figure 4.3.1.18 My Account page .....	34
Figure 4.3.2.1 Scanning car plate text .....	34
Figure 4.3.3.1 Admin login page .....	35
Figure 4.3.3.2 Admin dashboard page .....	35
Figure 4.3.3.3 Price update table page .....	36
Figure 4.3.3.4 History review page.....	36
Figure 4.3.3.5 History review with data page .....	37
Figure 5.3.1a JDK installation interface .....	39

Figure 5.3.1b Android Studio installation interface.....	39
Figure 5.3.1c Android Studio setup interface .....	40
Figure 5.3.1d Android Studio setup interface .....	40
Figure 5.3.1e Android Studio setup interface .....	41
Figure 5.3.1f Android Studio setup interface.....	41
Figure 5.3.1g Android Studio setup interface .....	42
Figure 5.3.1h Android Studio setup interface.....	42
Figure 5.3.1i Android Studio interface .....	43
Figure 5.3.1j Android Studio interface .....	43
Figure 5.3.1k Android Studio interface .....	44
Figure 5.3.1l Android Studio interface .....	44
Figure 5.3.1m Android Studio interface .....	45
Figure 5.3.2a Download Visual Studio Code interface .....	45
Figure 5.3.2b Setup Visual Studio Code interface.....	46
Figure 5.3.2c Setup Visual Studio Code interface .....	46
Figure 5.3.2d Setup Visual Studio Code interface.....	47
Figure 5.3.2e Setup Visual Studio Code interface .....	47
Figure 5.3.2f Setup Visual Studio Code interface .....	48
Figure 5.3.2g VS Code interface.....	48
Figure 5.3.2h VS Code interface.....	49
Figure 5.3.2i VS Code interface .....	49
Figure 5.3.2j VS Code interface .....	49
Figure 5.3.2k VS Code interface.....	50
Figure 5.3.2l VS Code interface .....	50
Figure 5.3.3a Setup XAMPP interface .....	51
Figure 5.3.3b List of folders in chosen drive .....	51
Figure 5.3.3c XAMPP interface.....	52
Figure 5.3.3d phpMyAdmin interface .....	52
Figure 5.5.1.1 Account Registration Interface.....	54
Figure 5.5.1.2 Login Interface .....	55
Figure 5.5.1.3 Update Profile Dialogue .....	55
Figure 5.5.1.4 Update Profile Interface .....	56
Figure 5.5.1.5 My Account Profile .....	57
Figure 5.5.1.6 Google Maps .....	57

Figure 5.5.1.7 Booking Interface .....	58
Figure 5.5.1.8 Booking Requirements .....	59
Figure 5.5.1.9 Booking Details .....	59
Figure 5.5.1.10 Upcoming Page .....	60
Figure 5.5.1.11 First and Second Scanning .....	61
Figure 5.5.1.12 Scanning Parked .....	62
Figure 5.5.1.13 Upcoming Page after Parking.....	62
Figure 5.5.1.14 QR Code Scanning .....	63
Figure 5.5.1.15 Current Page after Parking .....	63
Figure 5.5.1.16 Retrieve Page after Parking .....	64
Figure 5.5.1.17 Payment Details.....	65
Figure 5.5.1.18 Payment Activity .....	65
Figure 5.5.1.19 Scanning QR Code at Retrieve.....	66
Figure 5.5.1.20 History after Retrieve .....	67
Figure 5.5.1.21 Penalty Current Page .....	67
Figure 5.5.1.22 Penalty Payment .....	68
Figure 5.5.2.1 Admin Login Page.....	69
Figure 5.5.2.2 Price Update Table .....	69
Figure 5.5.2.3 History Review Page .....	70

## LIST OF TABLES

Table 3.3.2 Functional Requirements .....	13
Table 3.3.3 Non-functional Requirements.....	14
Table 3.5.1 Table of tasks .....	16
Table 4.2.2a Data dictionary for table Administrator .....	19
Table 4.2.2b Data dictionary for table Customer.....	19
Table 4.2.2c Data dictionary for table Customer_Vehicle .....	20
Table 4.2.2d Data dictionary for table Parking_Lot .....	20
Table 4.2.2e Data dictionary for table Booking.....	21
Table 4.2.2f Data dictionary for table Booking_Parking_Lot .....	21
Table 4.2.2g Data dictionary for table Parking_Station .....	22
Table 4.2.2h Data dictionary for table Payment .....	22
Table 4.2.2i Data dictionary for table Overtime_Vehicle.....	22
Table 4.2.2j Data dictionary for table Backup_Record .....	23
Table 4.2.2k Data dictionary for table Scanning .....	23
Table 4.2.2l Data dictionary for table Price_Update .....	24
Table 4.2.2m Data dictionary for Report .....	24
Table 5.4 Security Characteristics .....	53
Table 6.2.1 Test Case of Successful User Login .....	72
Table 6.2.2 Test Case of Unsuccessful User Login .....	72
Table 6.2.3 Test Case of Successful Account Registration .....	73
Table 6.2.4 Test Case of Unsuccessful User Login .....	74
Table 6.2.5 Test Case of Create Profile .....	75
Table 6.2.6 Test Case of Booking Location Validation.....	76
Table 6.2.7 Test Case of Booking Success .....	77
Table 6.2.8 Test Case of Successful Vehicle Plate Number First Scanning.....	78
Table 6.2.9 Test Case of Unsuccessful Vehicle Plate Number First Scanning .....	79
Table 6.2.10 Test Case of Vehicle Plate Number First Scanning Validation.....	80
Table 6.2.11 Test Case of Successful Vehicle Plate Number Second Scanning .....	81
Table 6.2.12 Test Case of Unsuccessful Vehicle Plate Number Second Scanning .....	81
Table 6.2.13 Test Case of Vehicle Plate Number Second Scanning Validation .....	82
Table 6.2.14 Test Case of Successful Upcoming QR Code Scanning.....	83
Table 6.2.15 Test Case of Unsuccessful Upcoming QR Code Scanning .....	84

Table 6.2.16 Test Case of Upcoming QR Code Scanning Validation.....	85
Table 6.2.17 Test Case of Current Page (No Vehicle Parked) .....	86
Table 6.2.18 Test Case of Current Page (Vehicle Parked) .....	86
Table 6.2.19 Test Case of Penalty Function .....	87
Table 6.2.20 Test Case of Successful Payment .....	88
Table 6.2.21 Test Case of Successful Retrieve QR Code Scanning.....	89
Table 6.2.22 Test Case of Unsuccessful Retrieve QR Code Scanning.....	90
Table 6.2.23 Test Case of Successful Vehicle Plate Number Third Scanning .....	90
Table 6.2.24 Test Case of Vehicle Plate Number Third Scanning Validation .....	91
Table 6.2.25 Test Case of History Page (No Vehicle is Retrieved) .....	92
Table 6.2.26 Test Case of History Page (Vehicle is Retrieved) .....	93
Table 6.2.27 Test Case of Unsuccessful Website Admin Login .....	94
Table 6.2.28 Test Case of Successful Website Admin Login.....	95
Table 6.2.29 Test Case of Update Price Table.....	95
Table 6.2.30 Test Case of History Review ( Date Filter) .....	96
Table 6.2.31 Test Case of History Review (Payment Date Filter) .....	97
Table 6.2.32 Test Case of History Review ( Total Filter) .....	98
Table 6.2.33 Test Case of History Review ( Status Filter) .....	99

## **CHAPTER 1: INTRODUCTION**

### **1.1 Introduction**

Smart parking system is a system that allows the user to park their vehicle without the efforts of looking for an available parking lot physically. The system allows the user to register their vehicles' car plate number, which will be used as validation when parking and retrieving the vehicle. The system consists of a booking system that allows the user to book the time and duration with respective car plate numbers. The system is embedded with OCR (Optical Character Recognition) ML model which allows the camera sensor to identify and verify the car plate whether it is the one registered by the user in the system. The OCR ML model can identify the car plate number better than usual algorithm which can prevent any detection error.

### **1.2 Problem Statement**

Nowadays, the number of vehicles is increasing. This is caused by problems such as large urban parking lots becoming inefficient, difficulty to find space in busy parking lots, and also the increasing need to offer larger areas for extra parking spaces. Major problems that the increasing number of vehicles and the decreasing efficiency of modern busy parking face are valuable time wasted from inconvenient and inefficient parking lots. Other than that is possible accidents caused by the abundance of moving vehicles in unorganized parking lots. Last but not least is consuming more fuel while idling or driving around parking lots, which results in more emissions of carbon dioxide (CO<sub>2</sub>).

### **1.3 Objectives**

Objectives in developing the Smart Parking System are to create a user-friendly and adaptable system that can be implemented in large, multi-level parking garages in order to alleviate parking hassles. Furthermore, it is to save more time for the drivers as it is time-consuming to look for parking space. Last but not least is to prevent traffic problems caused by the illegal parking of irresponsible drivers.

## 1.4 Scopes

### 1.4.1 Module to be developed

#### a) Login Module

Customers can login to their account by using their email address and password.

Administrators can login to their account by using their ID and password.

#### b) Register Module

Customers can register a new account with an email address and password.

#### c) Payment Module

Customers will be charged with the price based on the total days, hours or minutes. Prices will be different for weekends and holidays.

#### d) Booking Module

Customers can book the time and duration for the parking slot.

#### e) OCR Module

The camera can recognise the car plate text from the camera image.

#### f) Due Time Module

The user will receive notification 10 minutes before the due parking time; will receive warning if the user vehicle reaches due time; will give penalty warning to the user if the parked vehicle is 3 days past the due time.

#### g) Blacklist Module

The user will be blacklisted if he didn't clear the payment, and not allowed to park any vehicle until he clears the payment.

#### h) Price Update Module

Administrators can make updates to the price of the parking duration.

### i) Report Module

Administrators can generate monthly, multiple months, yearly or multiple years reports.

## 1.4.2 Target User

### a) Vehicle Owner

- The vehicle owner can register their car plate number
- Vehicle owners can book the duration for the parking slot, either in minutes, hours or days.
- Vehicle owners can extend their parking duration.
- Vehicle owner will be provided the notification email 10 minutes before the due time.
- Vehicle owners will receive a warning email if the due time passes.
- Vehicle owners will be provided QR code to retrieve their vehicle.
- Vehicle owners can cancel the booking anytime.

### b) Administrator

- The administrator can manage the system by using their administrator's ID and password.
- The administrator can trace back the customers' data such as the duration of the parking.
- The administrator can mark the vehicle if the duration is passed for more than 3 days.
- The administrator can update the base price for the parking duration.

## 1.5 Project Significance

The inspiration to develop the system is because of the shortage of space to park the vehicles around the country. Normally, there is traffic congestion in the parking area in places, for instance shopping malls, housing areas such as apartments and more. The traffic congestion increases the level of stress of the drivers because sometimes the parking space is not enough and

inconvenient for the numerous vehicles. Besides, the time it takes just to park the car is much longer than usual and it will interrupt the daily activities. Furthermore, by using the system, it will optimize parking space usage and help the flow of the traffic around the country while introducing an efficient, convenient and smart lifestyle.

## **1.6 Hardware and Software**

In developing the Smart Parking System, the team certainly has used hardware and software service to complete the system. The hardware which has been used are laptops and Android smartphones. Furthermore, Android Studio, Visual Studio Code, XAMPP, PhpMyAdmin are among the software which have been used.

## **1.7 Conclusion**

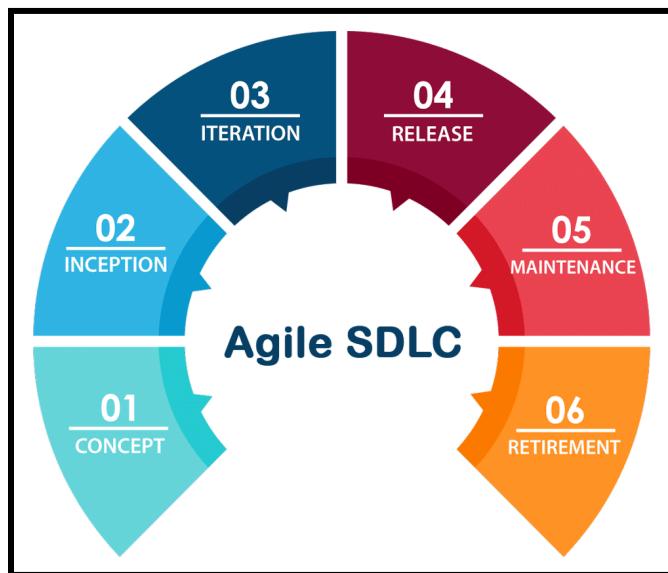
In conclusion, Smart Parking System gives various benefits to the user. As an user-friendly and adaptable system, it surely can be implemented in larger scale parking lots. This system has the potential to stand out and a lot of ideas and innovations can be added in order to improve the system to another excellent level.

## CHAPTER 2: SYSTEM METHODOLOGY AND PLANNING

### 2.1 Introduction

A consistent procedure used in an organization to carry out all the life cycles required which are planning and analyzing, design, development and implementation, testing and maintaining the software system is called System Development Methodology (SDM). A systems development methodology should be adopted by enterprises while developing systems. There are various types of models of System Development Methodology that can be used for the system such as Agile Development Methodology, DevOps Deployment Methodology, Waterfall Development Method, Rapid Application Development etc. However, for this Smart Parking System, the best system development methodology that can be implemented is Agile Development Methodology.

### 2.2 System Development Methodology



**Figure 2.2 Agile Methodology**

A systematic succession of phases that a system goes through as it develops from start to finish is known as the Agile Software Development Life Cycle. Concept, Inception, Iteration,

Release, Maintenance, and Retirement are its six phases. The project management methodology decided by the project team will have a small impact on the Agile life cycle.

- **Concept Phase**

In this stage, the project management team will decide the scope of their project. Generally, the team manager will prioritize the most important phase of the life cycle of the project to be developed first before others if there are numerous projects that need to be done. The project management team will have a short meeting with the client at the initiation phase of the development to ensure the project requirement, the objectives of the project, the functional and nonfunctional requirements including what features of the system will be supported are well understood and documented. The requirements are advisable to be kept to a minimum as the requirements can be expanded or to be added in later in the next phases. The estimation of the due time and the cost of the potential project should be done in this concept phase with the client to avoid any misunderstanding of communication between these two parties. Thus, the project management team can use and study this documentation to determine whether the project is possible before they start working on it.

- **Inception Phase**

This phase will happen once the concept is outlined. The main objective of this phase is to build the software development team. The Team Manager that is responsible for the project will pick the best team member based on their colleagues' availability, skills and technique needed following the requirement of the project. The necessary tools and resources will be provided by team management to ease the team to do their own designated role efficiently. After all the setups have been done accordingly, the design process can start and continue with other processes such as building the project architecture and designing the user interface. To completely develop the requirements on a diagram and establish the functionality of the product, further input from stakeholders is required at this inception stage. Check-ins at regular intervals will help to guarantee that the design process incorporates all needs.

- **Iteration Phase**

It is also known as the construction phase. This is the longest phase of the development process as a huge process will be done throughout this phase. To translate the design into code, the developers will collaborate with UX designers to incorporate all product requirements and user feedback. By the end of the first iteration or sprint, the product should have the bare minimum functionality. Later revisions can include more features and refinements. This stage is crucial to Agile software development since it enables programmers to produce functional software fast and make changes to please the client. This phase should be done precisely to ensure there is no overdue work as it can increase the cost and budgets of the project

- **Release Phase**

After all the design and development phase has been done, the final product is almost ready to release to the intended client. However, there is one phase where a needed testing should be done by the quality assurance team to ensure the software has no errors and fully functional according to the requirements of the project. Basically, this assurance team needs to make sure the coding of the implementation is clean so that there are no potential bugs or defects on the code. If there are bugs or syntax errors detected, the developers will quickly take care of them. In this phase, user training also will be done before releasing the final product to the client to ensure the client or intended user knows how to use the software. Last but not least, after all of this process has been done perfectly, the final product iteration can be released to the client or intended user to be used.

- **Maintenance Phase**

Now that the program has been installed completely where users can access it, then it will enter the maintenance phase. The software development team will continue to offer help throughout this stage to keep the system operating efficiently and fix any new bugs. This phase basically is a phase where the maintenance team will keep up to date with the software to ensure there are no problems. They will also be available to provide clients or users with further training and to make sure they are familiar with how to

utilize the product. Iterations can be made to the current product over time to add improvements and new features.

- **Retirement Phase**

A product might reach the retirement phase for one of two reasons: either it is being replaced with new software or the system itself has grown dated or incompatible with the company over time. Users will initially receive notification from the program development team that the software is being decommissioned. The users will be transferred to the new system if there is a replacement. The developers will then complete any outstanding end-of-life tasks and stop providing support for the current product.

## **2.3 System Development Techniques**

For this Smart Parking System, there are two types of System Development Technique that will be implemented throughout the development cycle. First is Modelling Technique. Systems developers can examine, test, and adjust a graphical representation of a notion or process created through modeling. A system analyst can use a collection of business, data, object, network, and process models to characterize and streamline the system. For this Smart Parking System, Data Flow Diagramming (DFD) and the flowchart of the system will be provided in later part of documentation. Basically, reasons that programmers employ when creating code modules is described by a process model. The models overlap at first glance, but they actually function together to describe the same environment from several angles.

Second technique of the Smart Parking System that will be implemented is Structured Analysis. Traditional systems development methods like structured analysis are tried-and-true and simple to comprehend. The systems development cycle (SDLC), a set of steps used by structured analysis, is used to plan, analyze, design, implement, and support the software system. Structured analysis has been a mainstay of systems development since mainframe computing became the standard for most systems. In structured analysis, a system is graphically described using a collection of process models. Structured analysis is referred to as a process-centered technique because it places a strong emphasis on the procedures that convert data into valuable

knowledge. Structured analysis encompasses issues with data organization and structure, relational database design, and user interfaces in addition to modeling the processes. The data entering a process, the business rules transforming the input, and the output data flow are all identified through process modeling.

## **2.4 Conclusion**

In conclusion, we choose Agile Software Development Life Cycle because Agile has flexibility in developing the system. The team can effortlessly respond to any changes that may need to be made. In other words, it makes the development process much more adaptable. Last but not least, the system is developed using Agile because it is easier to work on small deliverables rather than the whole work. Thus, it saves a significant amount of loss in terms of work effort because each agile sprint might be anywhere between one and four weeks or even just one day.

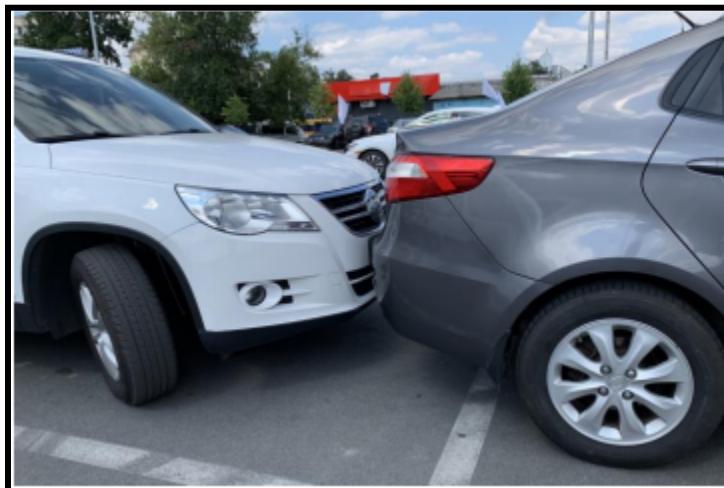
## CHAPTER 3: SYSTEM ANALYSIS

### 3.1 Introduction

In this chapter, the analysis of current problems, analysis of proposed system, structured chart of proposed system and work breakdown structure will be described.

### 3.2 Analysis of Current Problem

In Malaysia, the number of cars on the road is increasing. The expert also says the number of registered cars and vehicles in the country has surpassed the total number of human population. This leads to issues such as inefficient huge urban parking lots. The problem is still the same because even though the people are provided with huge parking lots, the people are still wasting time to find parking space by themselves. Therefore, it is still inconvenient and inefficient especially to people who are rushing in the morning to go to class or work. Next, there are a lot of possible accidents caused by the abundance of moving vehicles in unorganized parking lots. The collision between the vehicles usually happens when the vehicles are queuing up to find a parking spot. The drivers tend to become impatient to grab the parking spaces that are available until they did not realize that they hit the car in front of them.

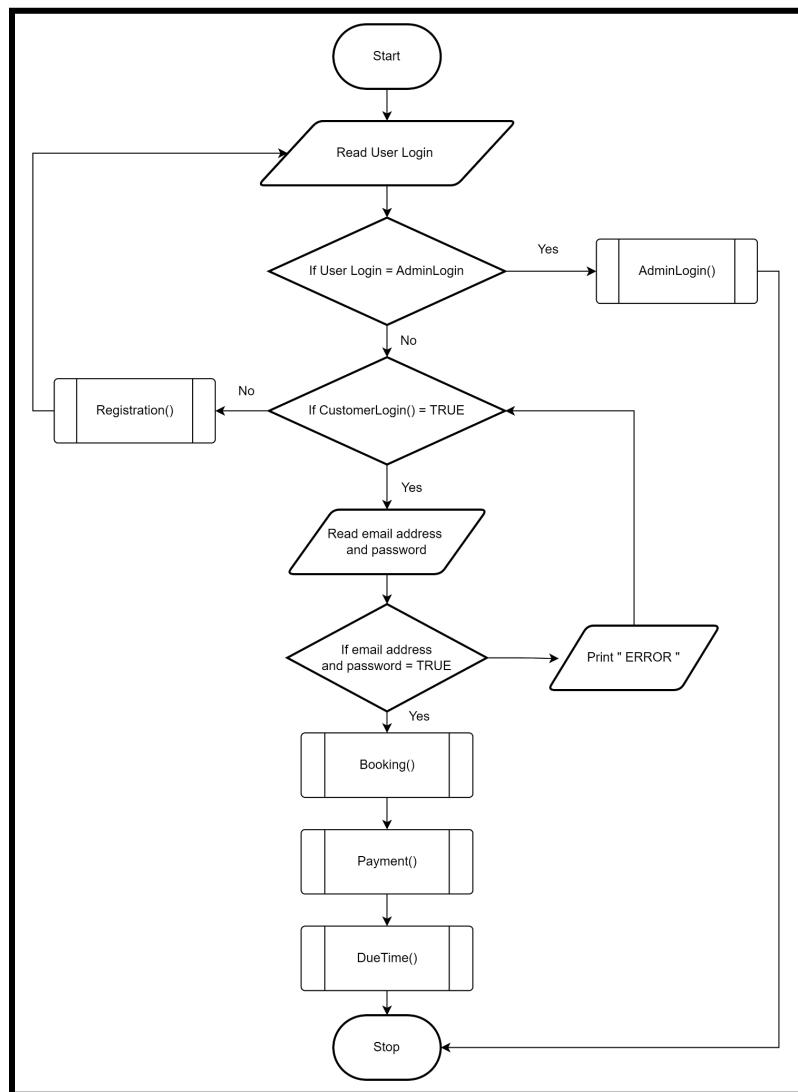


**Figure 3.2 The collision between two cars while finding a parking spot**

In addition, we currently experienced human-made climate change that interchanges the environment that we lived in. We, the people, are responsible to reduce anything that will worsen the climate change. The people can start by reducing usage of fuel. But, when people are idling or driving around parking lots to find the space for their vehicles, it will consume a lot of fuel which will result in more emissions of carbon dioxide which is not good for our environment.

### 3.3 Analysis of Proposed System

#### 3.3.1 Background



**Figure 3.3.1 Flowchart of the whole system**

With the overloading number of vehicles registered compared to the Malaysia residents, it would cause difficulties in finding parking lots. This situation not only leads to time wasting of every driver while waiting or searching for parking spaces but also increases the rate of accidents. The Smart Parking System is able to solve the problem stated and provide conveniences to the drivers as it would help the users park their car automatically. Users just need to place their vehicle in the elevator, then it will be placed at the arranged parking lot. What drivers used to do is to drive slowly along the road and search for empty parking slots. This may not only waste the drivers time, but also prevent any accident from happening. As drivers queuing up for parking slots might get frustrated or annoyed and crash into the vehicle ahead of them.

The Smart Parking System consists of 5 main modules which are the User Profile module, Booking module, OCR module, Payment module and the Management module. For the User Profile module, users are able to register an account with their email, username and password which is an account to keep track of the user's booking details, history and also registered vehicle plate number. Users are able to keep track of all their past booking or parking details.

Next is the Booking module. Under the Booking module, users are able to insert the location and search for the nearest smart parking station available, after choosing then insert the booking details. The booking details include the dates, duration, and also the selecting the registered vehicle plate number. Notification will be sent to the users' mobile device 10 minutes before reaching the booking time. Besides that, the user is also able to make any changes to the booking prior to the booking time. After the vehicle is parked, users could also extend the parking duration.

Before the users park their vehicle, their vehicle would need to undergo some form of validation to make sure that the vehicle is the one that is booked for the parking slot, here comes the OCR module. In the OCR module, the vehicle plate number will be scanned using a camera sensor and converted to text with the use of machine learning. If it is the booked vehicle plate number, the vehicle is passed and a QR code for vehicle

retrieval will be provided to the user's account. Else, the error sound would be provided from the scanning device to indicate that the vehicle plate number is not in the booking.

Under the Payment module, the user can scan the QR code through the camera sensor to retrieve their vehicle. Payment would need to be done first before the retrieving process. Users are allowed to pay using online banking or cards. Extra charging would be conducted for overtime parking. After payment, an invoice would be sent to the customer email.

The Management module is for the administrator site where the administrator is able to set the holiday through the calendar. Administrators can change the price for the weekday and weekend/holiday. Besides that, 3 types of reports can be generated from the users' booking database which are monthly, multiple months and yearly reports.

### 3.3.2 Functional Requirements

FR_ID	Requirements
FR01	Users should be able to enter email address, username and password to register an account.
FR02	Users should be able to enter username and password to login to the system.
FR03	Users should be able to log out of the system by pressing the log out button.
FR04	Users should be able to search for the smart parking location by using the search bar of the map.
FR05	Users should be able to extend the booking duration.
FR06	Users should be able to edit the booking details.
FR07	Users should be receiving the QR code once the registered vehicle car plate is scanned and approved.
FR08	Users should be receiving notification 10 minutes before the booking start time.

<b>FR09</b>	Users should be able to receive notification 30 minutes before the booking end time.
<b>FR10</b>	The system should produce error noise when the scanned vehicle plate number is not in the booking database list.
<b>FR11</b>	Administrator should be able to set the holiday date.
<b>FR12</b>	Administrator should be able to edit the normal and holiday price.
<b>FR13</b>	Administrators should be able to generate reports based on the report type chosen.

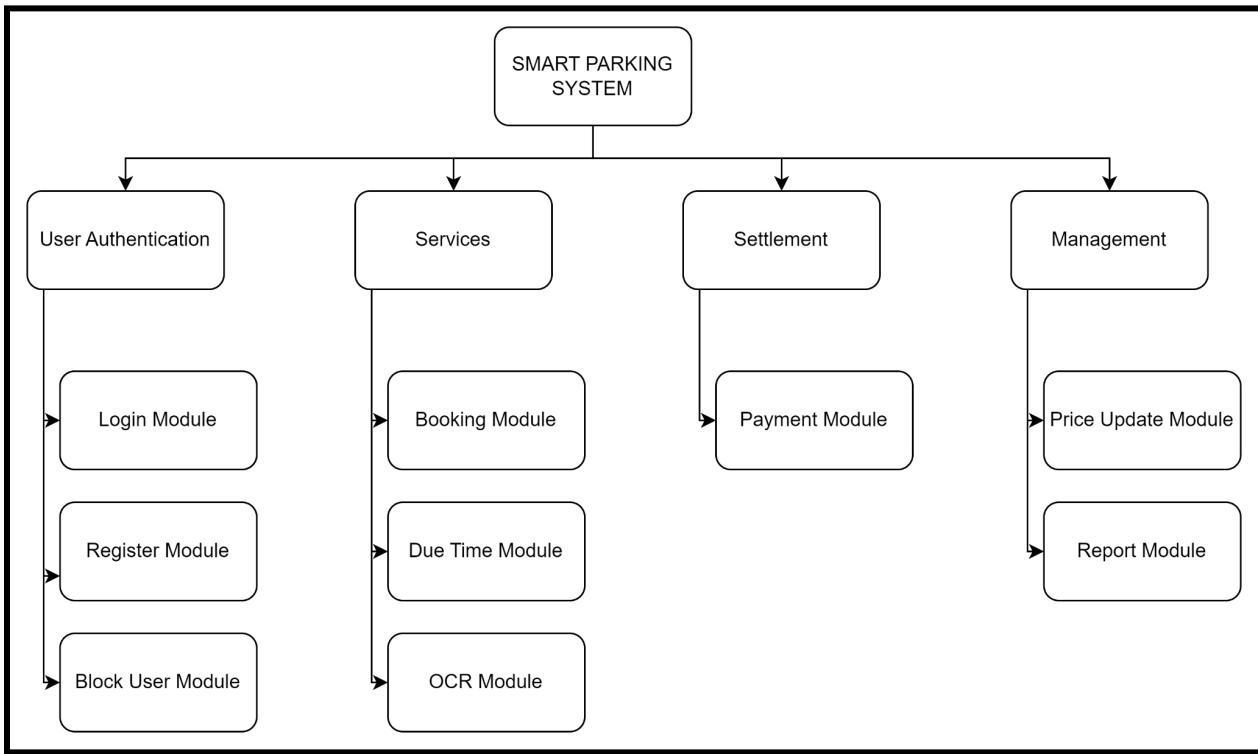
**Table 3.3.2 Functional Requirements**

### 3.3.3 Non-functional Requirements

<b>NFR_ID</b>	<b>Requirements</b>
<b>NFR01</b>	The main page load time must not be more than 5 seconds for users that access the application.
<b>NFR02</b>	The system should be available at all times.
<b>NFR03</b>	The report generation should not take longer than 10 seconds.
<b>NFR04</b>	The application must be scalable enough to support 300000 users at a time.

**Table 3.3.3 Non-functional Requirements**

### 3.4 Structured Chart of Proposed System



**Figure 3.4 Structured Chart of proposed system**

### 3.5 Work Breakdown

Work breakdown is a method to divide and conquer large projects to get things done faster and more efficiently. In this stage we will determine parts and responsibilities and how to best use each member's skillset to do his part in the project.

#### 3.5.1 Task Distribution

We establish the organizational chart and table for the tasks and define the levels of responsibilities of our team members.

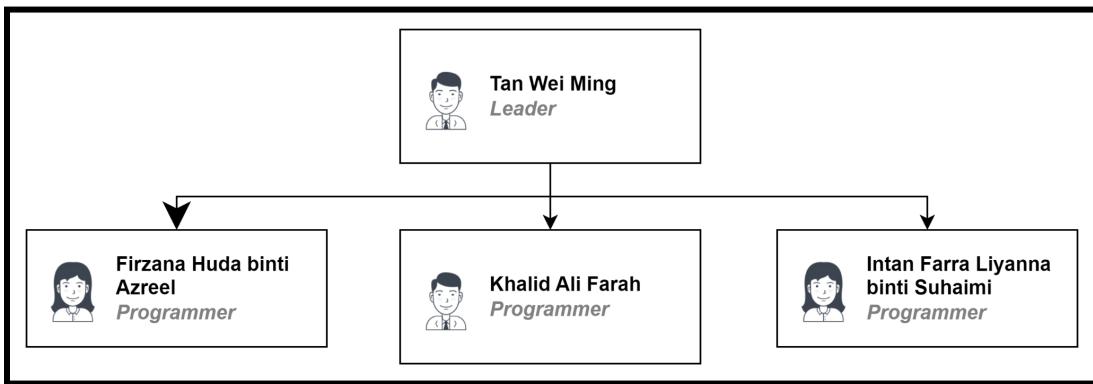


Figure 3.5.1 Organizational chart

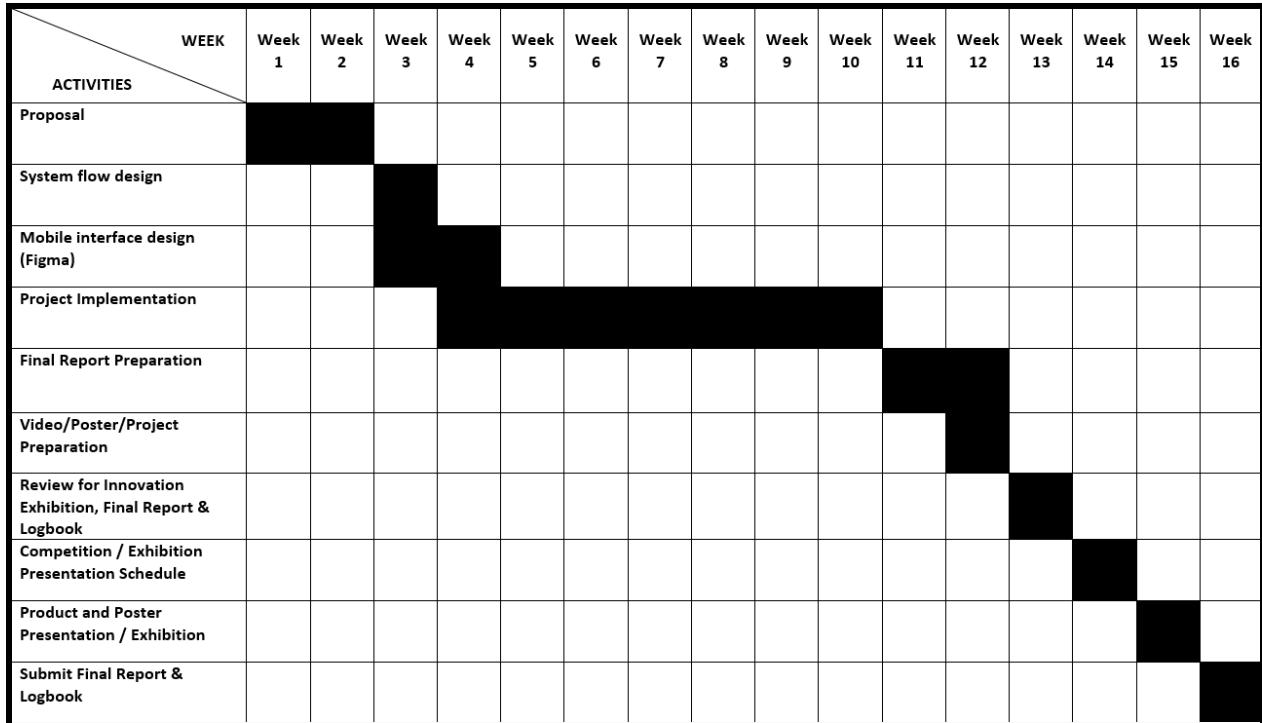
The following table shows the task distribution of each team member.

No.	Name	Task
1	Tan Wei Ming	<ul style="list-style-type: none"><li>• OCR Module</li><li>• Due Time Module</li></ul>
2	Firzana Huda binti Azreel	<ul style="list-style-type: none"><li>• Block User Module</li><li>• Price Update Module</li><li>• Report Module</li></ul>
3	Khalid Ali Farah	<ul style="list-style-type: none"><li>• Login Module</li><li>• Register Module</li></ul>
4	Intan Farra Liyanna binti Suhaimi	<ul style="list-style-type: none"><li>• Payment Module</li><li>• Booking Module</li></ul>

Table 3.5.1 Table of tasks

### 3.6 Gantt Chart

Project Gantt Chart is important in order for the team to check if their works are on track or already pass the planned period. The following diagram shows the project activities and duration to finish the tasks.



**Figure 3.6 Gantt Chart**

### 3.7 Conclusion

In conclusion, each team member has their own commitment and responsibility in the project in order to ensure the system is completed as planned. There is also complete planning which has been done to be the guidance to the developer to develop the system.

## CHAPTER 4: SYSTEM DESIGN

### 4.1 Introduction

In this chapter, we will describe the physical structure of the system using conceptual database design, logical database design and physical database design. Each design will include a few diagrams related.

### 4.2 Database Design

#### 4.2.1 Entity Relationship Diagram (ERD)

Figure below shows the Entity Relationship Diagram of the system.

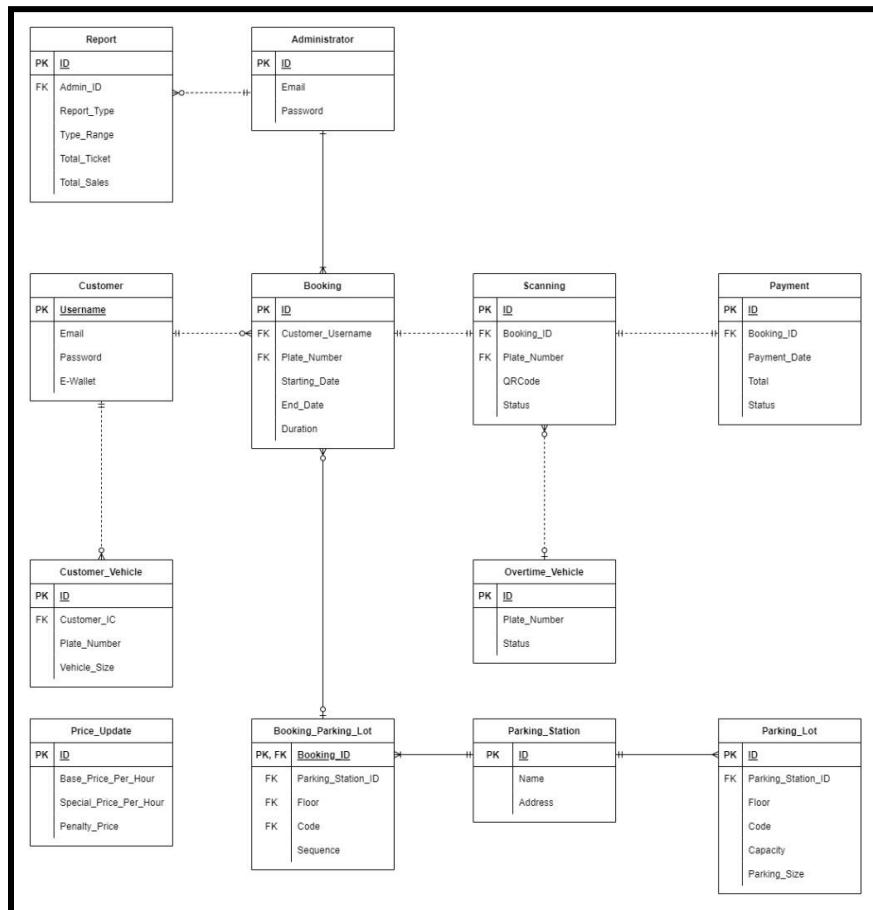


Figure 4.2.1 Entity Relationship Diagram (ERD)

#### 4.2.2 Data Dictionary

##### a. Administrator

Administrator					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	6	CHAR	ADXXXX	PK	Unique ID for Administrator
Email	80	VARCHAR	XX@XX	Not Null	Administrator Email
Password	50	VARCHAR		Not Null	Administrator Password

Table 4.2.2a Data Dictionary for Table Administrator

##### b. Customer

Customer					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>Username</u>	50	VARCHAR		PK	Username of customer
Email	80	VARCHAR	XX@XX	Not Null	Customer email
Password	50	VARCHAR		Not Null	Customer password
E-Wallet	(7,2)	DOUBLE	XXX	Null	Electronic wallet of customer

Table 4.2.2b Data Dictionary for Table Customer

**c. Customer\_Vehicle**

Customer_Vehicle					
Field Name	File Size	Data Type	Data Format	Constraint	Example
<u>ID</u>	7	CHAR	VXXXXXX	PK	Unique ID for customer vehicle
Customer_Username	50	VARCHAR		FK	Foreign key constraint referencing customer username
Plate_Number	15	VARCHAR		Not Null	Customer vehicle plate number
Vehicle_Size	5	CHAR		Null	Vehicle size with 2 options: small, or large

**Table 4.2.2c Data Dictionary for Table Customer\_Vehicle**

**d. Parking\_Lot**

Parking_Lot					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	5	VARCHAR	PLXXX	PK	Unique ID for Parking_Lot
Parking_Station_ID	5	VARCHAR	PSXXX	FK	Foreign key with reference to ID of Parking Station
Floor	2	INT		Not Null	Parking_lot floor
Code	1	CHAR	A	Not Null	code of Parking_lot
Capacity	3	INT		Not Null	Capacity of Parking_Lot
Parking_Size	5	CHAR		Not Null	Parking lot size with 2 options: small, or large

**Table 4.2.2d Data Dictionary for Table Parking\_Lot**

### e. Booking

Booking					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	6	VARCHAR	BXXXXX	PK	Unique ID for Booking
Customer_Username	50	VARCHAR		FK	Foreign key constraint referencing Customer Username
Plate_Number	15	VARCHAR		FK	Foreign key constraint referencing Customer_Vehicle Plate_number
Parking_ID	5	VARCHAR	PLXXX	FK	Foreign key constraint referencing Parking_ID
Starting_Date		DATETIME	dd-mm-yyyy hh:mm:ss	Not Null	The starting date for Booking
End_Date		DATETIME	dd-mm-yyyy hh:mm:ss	Not Null	The end date of Booking
Duration	5	INT		Not Null	The Duration of Booking in hour

**Table 4.2.2e Data Dictionary for Table Booking**

### f. Booking\_Parking\_Lot

Booking_Parking_Lot					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>Booking_ID</u>	6	VARCHAR	BXXXXX	PK, FK	Primary key for Booking_Parking_Lot
Parking_Station_ID	5	VARCHAR	PSXXX	FK	Foreign key with reference to ID of Parking Station
Floor	2	INT		FK	Foreign key with reference to floor of Parking_Lot
Code	1	CHAR	A	FK	Foreign key with reference to code of Parking_Lot
Sequence	3	INT	001	Not Null	Index of parking lot

**Table 4.2.2f Data Dictionary for Table Booking\_Parking\_Lot**

### **g. Parking\_Station**

Parking_Station					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	5	VARCHAR	PSXXX	PK	Primary key for Parking_Station
Name	100	VARCHAR		Not Null	Name of the station
Address	150	VARCHAR		Not Null	Address of the station

**Table 4.2.2g Data Dictionary for Table Parking\_Station**

### **h. Payment**

Payment					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	6	VARCHAR	PXXXXX	PK	Unique ID for Payment
Booking_ID	6	VARCHAR	BXXXXX	FK	Foreign key constraint referencing Booking ID
Payment_Date		DATETIME	dd-mm-yyyy hh:mm:ss	Null	The date of Payment
Total	(6, 2)	DOUBLE		Not Null	The total of Payment
Status	10	VARCHAR		Not Null	Payment status

**Table 4.2.2h Data Dictionary for Table Payment**

### **i. Overtime\_Vehicle**

Overtime_Vehicle					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	5	VARCHAR	OVXXX	PK	Unique ID for Overtime vehicle
Plate_Number	8	VARCHAR		FK	Foreign key constraint referencing Scanning Plate_Number
Status	10	VARCHAR		Not Null	The status of Overtime vehicle (pending / parked / retrieved)

**Table 4.2.2i Data Dictionary for Table Overtime\_Vehicle**

### j. Backup\_Record

Backup_Record					
Field Name	File Size	Data Type	Data Format	Constraint	Description
ID	7	VARCHAR	BRXXXXX	PK	Unique ID for Backup_Record
Customer_Username	50	VARCHAR		Not Null	Customer username of Backup_Record
Plate_Number	15	VARCHAR		Not Null	Car plate number of Backup_Record
Starting_Date		DATETIME	dd-mm-yyyy hh:mm:ss	Not Null	Starting date of Backup_Record
End_Date		DATETIME	dd-mm-yyyy hh:mm:ss	Not Null	End date of Backup_Record
Duration	50	VARCHAR		Not Null	Total duration of booking of Backup_Record
Report_ID	6	VARCHAR	RXXXXX	FK	Foreign key referencing Report_ID

Table 4.2.2j Data Dictionary for Table Backup\_Record

### k. Scanning

Scanning					
Field Name	File Size	Data Type	Data Format	Constraint	Description
ID	6	VARCHAR	SXXXXX	PK	Unique ID for scanning
Booking_ID	6	VARCHAR	BXXXXX	FK	Foreign key referencing Booking_ID
Plate_Number	15	VARCHAR		FK	Foreign key referencing Booking Plate_Number
QRCode	50	VARCHAR			QR code content string for Scanning
Status	10	VARCHAR			The status of scanning

Table 4.2.2k Data Dictionary for Table Scanning

## I. Price\_Update

Price_Update					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	3	CHAR	PRX	PK	Unique ID for Price Update
Base_Price_Per_Hour	2	INT			Base price for hourly parking
Special_Price_Per_Hour	2	INT			Price for holiday or weekend parking
Penalty_Price	3	INT			Price for vehicle that is under penalty

**Table 4.2.2l Data Dictionary for Table Price\_Update**

## m. Report

Report					
Field Name	File Size	Data Type	Data Format	Constraint	Description
<u>ID</u>	6	VARCHAR	RXXXXX	PK	Unique ID for Report
Admin_ID	6	VARCHAR	ADXXXX	FK	Foreign key constraint referencing Administrator ID
Report_Type	10	VARCHAR		Not Null	Report type that indicates whether the report is yearly, monthly
Type_Range	15	VARCHAR		Not Null	Type range shows the year, range month or month of the report
Total_Ticket		INT		Not Null	Total ticket of the chosen range
Total_Sales	(10, 2)	DOUBLE		Not Null	Total sales of the report

**Table 4.2.2m Data Dictionary for Report**

## 4.3 Module Integration and Interface Design

The interface design of the Smart Parking System has been created by using Figma.

### 4.3.1 Smart Parking System Mobile Application

#### 4.3.1.1 Account Registration



Figure 4.3.1.1 Account registration page

#### 4.3.1.2 Login

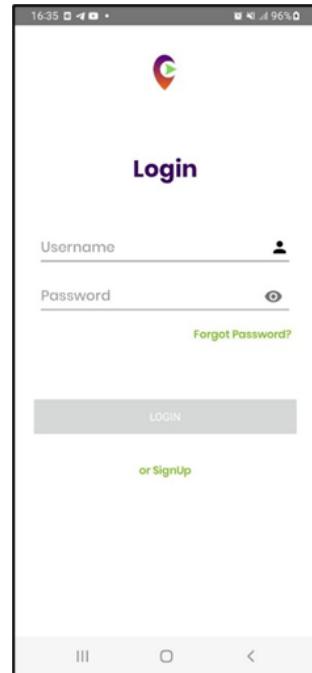


Figure 4.3.1.2 Login page

#### 4.3.1.3 Create Profile

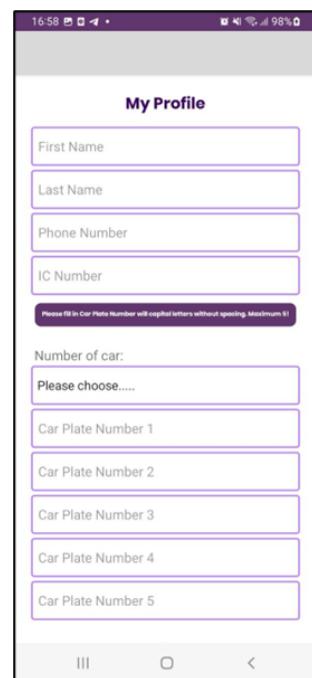


Figure 4.3.1.3 Create profile page

#### 4.3.1.4 Current Page



Figure 4.3.1.4 Current page

#### 4.3.1.5 Current Page with Data

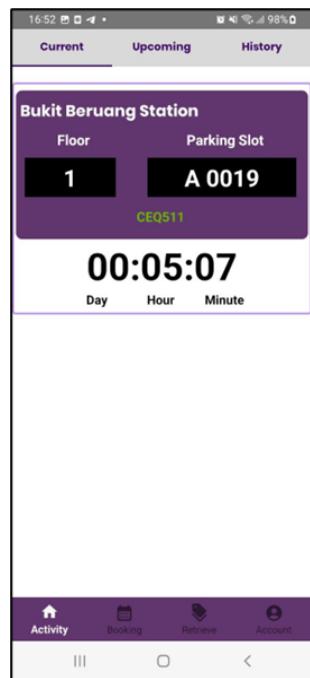


Figure 4.3.1.5 Current Page with Data

#### 4.3.1.6 Upcoming Page



Figure 4.3.1.6 Upcoming page

#### 4.3.1.7 Upcoming Page with Data

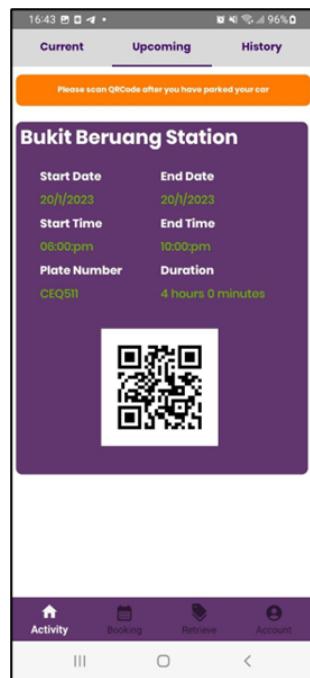


Figure 4.3.1.7 Upcoming page with data

#### 4.3.1.8 History Page

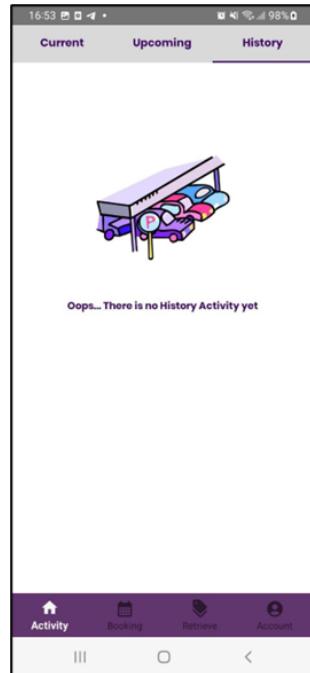


Figure 4.3.1.8 History page

#### 4.3.1.9 History Page with Data

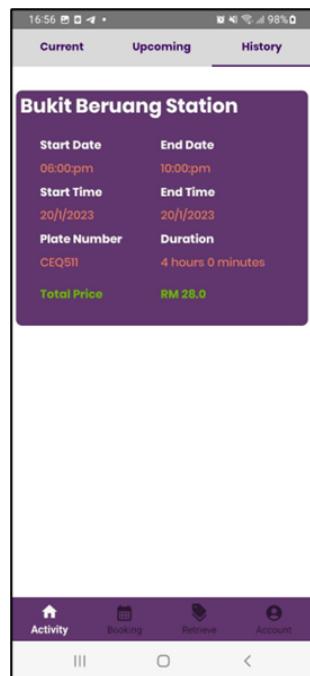


Figure 4.3.1.9 History page with data

#### 4.3.1.10 Google Maps



Figure 4.3.1.10 Google Maps page

#### 4.3.1.11 Booking Page

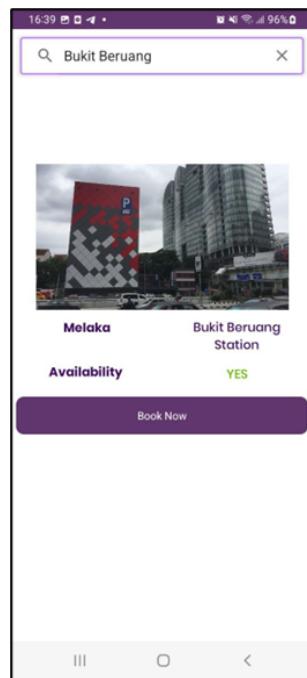


Figure 4.3.1.11 Booking page

#### 4.3.1.12 Booking Requirements

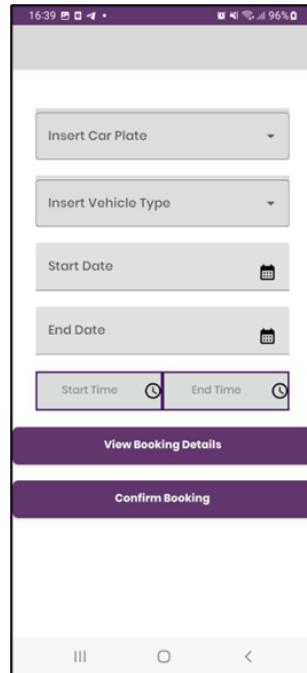


Figure 4.3.1.12 Booking requirements page

#### 4.3.1.13 Booking Details

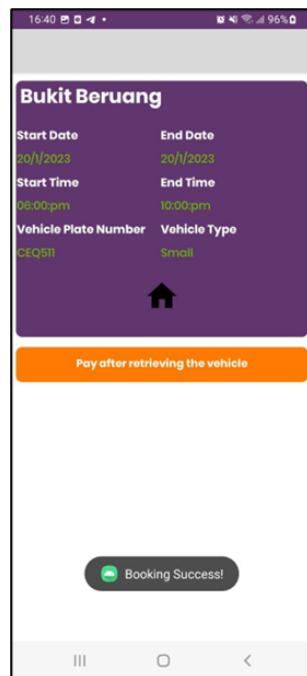


Figure 4.3.1.13 Booking details page

#### 4.3.1.14 Retrieve Page



Figure 4.3.1.14 Retrieve page

#### 4.3.1.15 Retrieve Page with Data

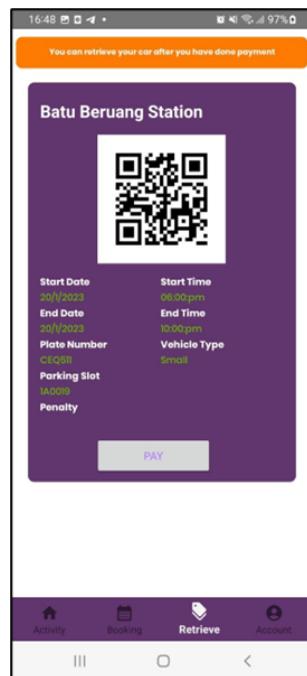


Figure 4.3.1.15 Retrieve page with data

#### 4.3.1.16 Payment Details



Figure 4.3.1.16 Payment details

#### 4.3.1.17 Payment Page

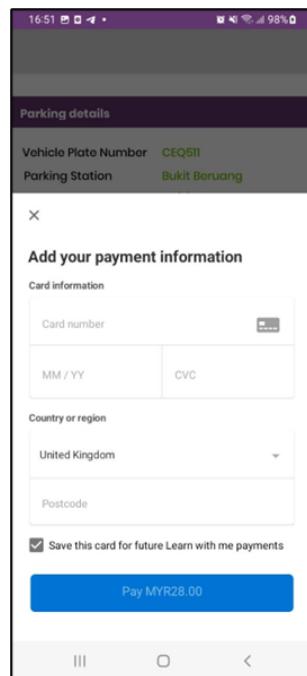


Figure 4.3.1.17 Payment page

#### 4.3.1.18 My Account

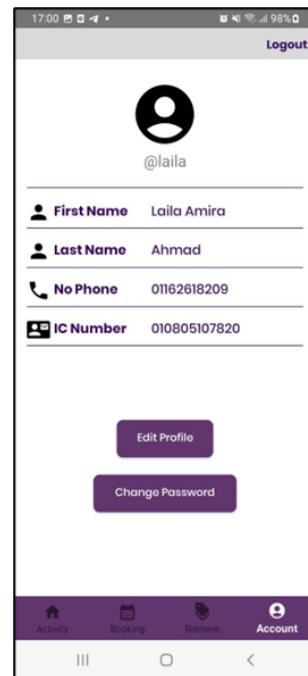


Figure 4.3.1.18 My Account page

#### 4.3.2 Smart Parking System Scanning

##### 4.3.2.1 Scanning Car Plate

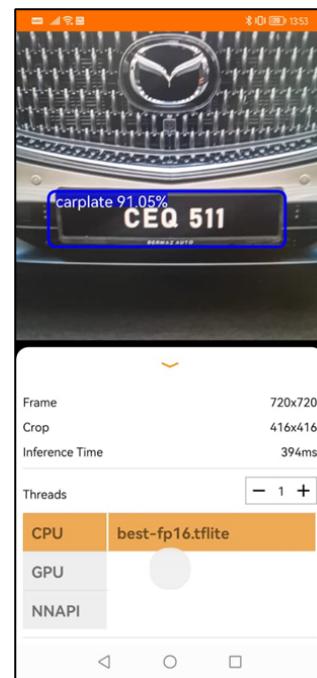
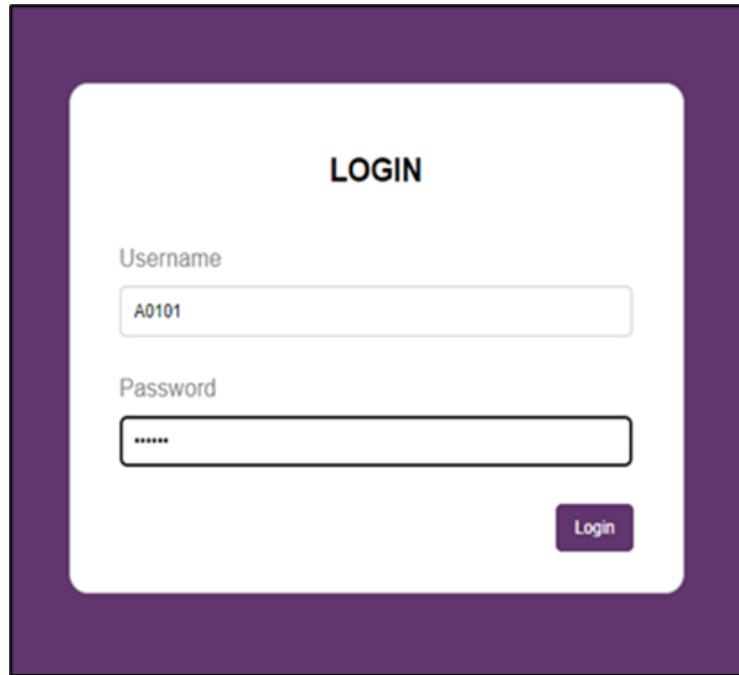


Figure 4.3.2.1 Scanning car plate text

### 4.3.3 Smart Parking System Admin Website

#### 4.3.3.1 Admin Login



The image shows the admin login page of the Smart Parking System. It features a white rectangular form on a dark purple background. At the top center, the word "LOGIN" is written in bold capital letters. Below it, there are two input fields: "Username" with the value "A0101" and "Password" with four asterisks. At the bottom right of the form is a purple "Login" button.

Figure 4.3.3.1 Admin login page

#### 4.3.3.2 Admin Dashboard



Figure 4.3.3.2 Admin dashboard page

### 4.3.3.3 Price Update Table

Price Update Table					
Normal Price (Hour)	Holiday Price (Hour)	Normal Price (Day)	Holiday Price (Day)	Penalty Price	Edit Price
RM 0	RM 0	RM 0	RM 0	RM 0	<button>Edit Price</button>
<a href="#">Go back!</a>					

### **Figure 4.3.3.3 Price update table page**

#### **4.3.3.4 History Review**

History Review			
From Date	To Date	Click to Filter	
<input type="text" value="mm/dd/yyyy"/> <span>  <input type="button" value="Calendar"/></span>	<input type="text" value="mm/dd/yyyy"/> <span>  <input type="button" value="Calendar"/></span>	<span>Filter</span>	
Show <input type="text" value="10"/> entries		Search: <input type="text"/>	
ID <span>▲</span>	Booking_ID	Total	Status
<span>◆</span> <b>Payment_Date</b> <span>◆</span>			
No data available in table			
Showing 0 to 0 of 0 entries		<span>Previous</span>	<span>Next</span>
<span>Go back!</span>			

#### **Figure 4.3.3.4 History review page**

#### 4.3.3.5 History Review with Data

History Review					
From Date		To Date		Click to Filter	
01/12/2023		01/24/2023		Filter	
Show 10 entries					
ID	Booking_ID	Payment_Date	Total	Status	
4	B00003	2023-01-12 17:10:50	5.00	Paid	
5	B00004	2023-01-13 17:10:50	13.00	Paid	
11	B00010	2023-01-17 17:10:50	30.00	Paid	
2	B00001	2023-01-12 17:10:50	0.00	Not Paid	
3	B00002	2023-01-12 17:10:50	0.00	Not Paid	
6	B00005	2023-01-13 17:10:50	0.00	Not Paid	
7	B00006	2023-01-14 17:10:50	0.00	Not Paid	
8	B00007	2023-01-14 17:10:50	0.00	Not Paid	
9	B00008	2023-01-15 17:10:50	0.00	Not Paid	
10	B00009	2023-01-17 17:10:50	0.00	Not Paid	
Showing 1 to 10 of 10 entries					
<a href="#">Go back!</a>					
<a href="#">Previous</a> <a href="#">1</a> <a href="#">Next</a>					

Figure 4.3.3.5 History review with data page

#### 4.4 Conclusion

In conclusion, the system design is the most important part. It is because it helps developers to visualize what the system should look like or the environment of the system.

## **CHAPTER 5: SYSTEM IMPLEMENTATION**

### **5.1 Introduction**

During system implementation, all the planning and designing will start to convert to the real product. All the production of the system which covers from installation of compilers, developing system, testing and debugging will be carried out by all the team members. The aim of implementation is to make sure the final product is able to operate smoothly according to what is desired during the designing phase.

### **5.2 System Development Environment**

#### **5.2.1 Android Studio**

During system implementation, all the planning and designing will start to convert to the real product. All the production of the system which covers from installation of compilers, developing system, testing and debugging will be carried out by all the team members. The aim of implementation is to make sure the final product is able to operate smoothly according to what is desired during the designing phase.

#### **5.2.2 Visual Studio Code**

VS Code is a source-code editor that was made by Microsoft that supports Windows 7 and above. It supports a wide collection of programming languages such as Java, C++, PHP, Python and more. VS Code supports the download of extensions such as debugger, code highlighter, and code formatting. These extensions allow the users to code in different languages. In our project, the VS Code will be used mainly to code PHP programs and establish connections between the application and website to the xampp database.

#### **5.2.3 XAMPP**

XAMPP is one of the widely known cross-platform web servers that helps developers to test their systems and programs on a local webserver. XAMPP is actually an acronym for Cross-Platform, Apache, MySQL, PHP and Perl. It provides Apache HTTP Server, MariaDB and

interpreters for 11 languages. There are multiple components that are parts of XAMPP which will be used in our system are phpMyAdmin, Apache and MySQL. Apache is a HTTP cross-platform web server which takes in HTTP requests from the users and presents them in the form of a webpage. MySQL is an open-source relational database management system in which the users can perform CRUD actions (Create, Read, Update, Delete) on the data. Last but not least, phpMyAdmin is a web application that is used to manage the users MySQL database system.

## 5.3 System Configuration Management

### 5.3.1 Installation and Setup of Android Studio

- Install the JDK for the Android Studio from the link  
<https://www.oracle.com/java/technologies/downloads/>

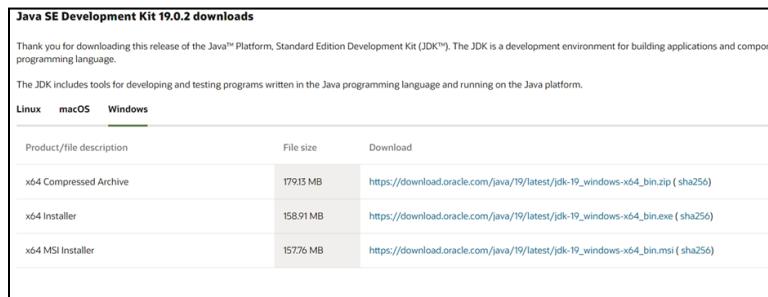


Figure 5.3.1a JDK installation interface

- Install Android Studio from the link  
[https://developer.android.com/studio?gclid=Cj0KCQiAt66eBhCnARIsAKf3ZNEDluOoP42W-\\_xfLpXMaRbxXpsfrrOUCnMZKmEJ-52jIv3bX0XR8nIaAt2FEALw\\_wcB&gclsrc=aw.ds](https://developer.android.com/studio?gclid=Cj0KCQiAt66eBhCnARIsAKf3ZNEDluOoP42W-_xfLpXMaRbxXpsfrrOUCnMZKmEJ-52jIv3bX0XR8nIaAt2FEALw_wcB&gclsrc=aw.ds)

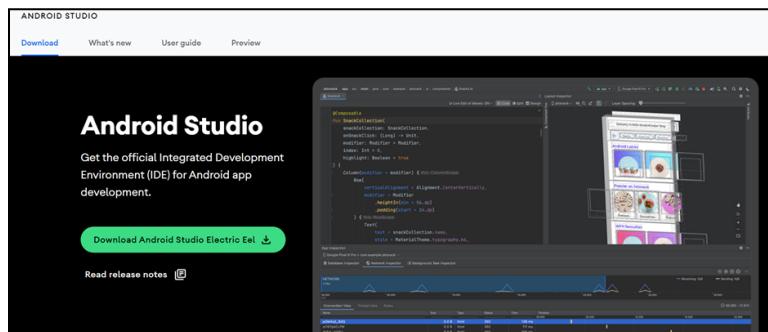


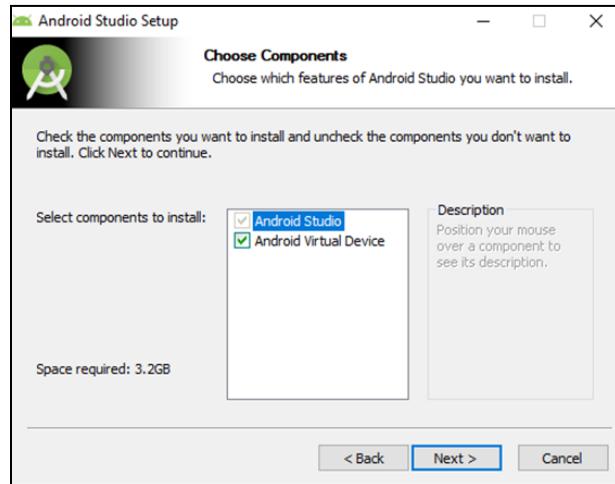
Figure 5.3.1b Android Studio installation interface

- Run the .exe file and click ‘Next’



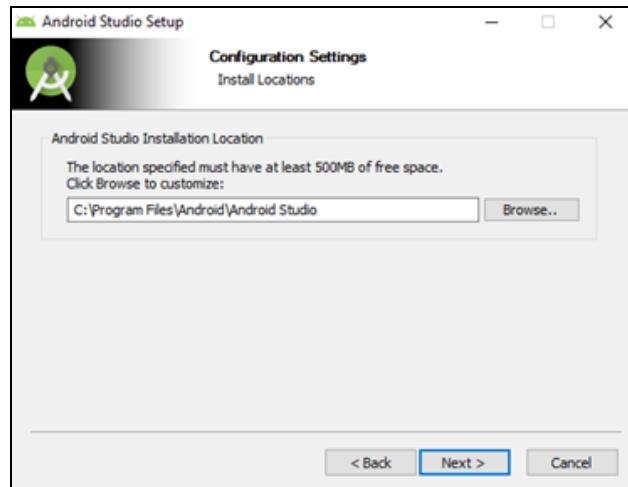
**Figure 5.3.1c** Android Studio setup interface

- In the Choose Components page, follow the default and click the ‘Next’ button.



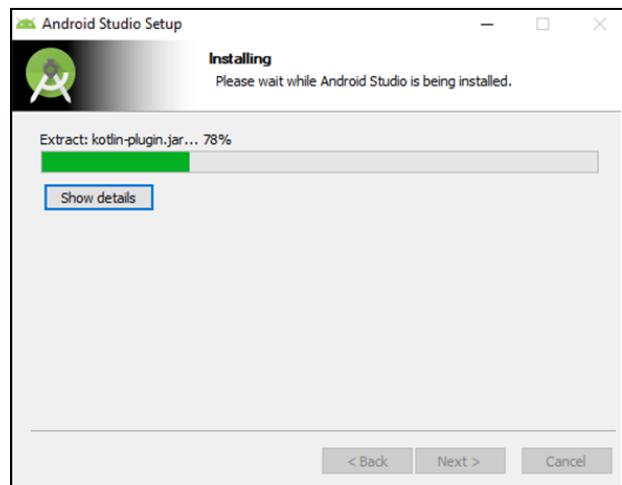
**Figure 5.3.1d** Android Studio setup interface

- In the Configuration Settings page, choose a location and click the “Next” button.



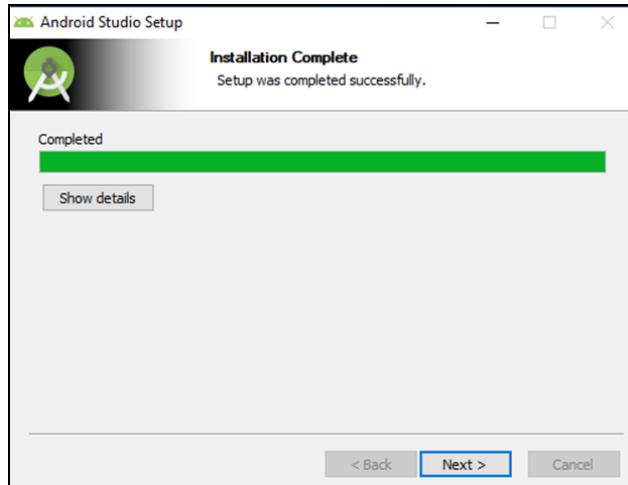
**Figure 5.3.1e** Android Studio setup interface

- Click the “Install” button and wait for the installation to finish.



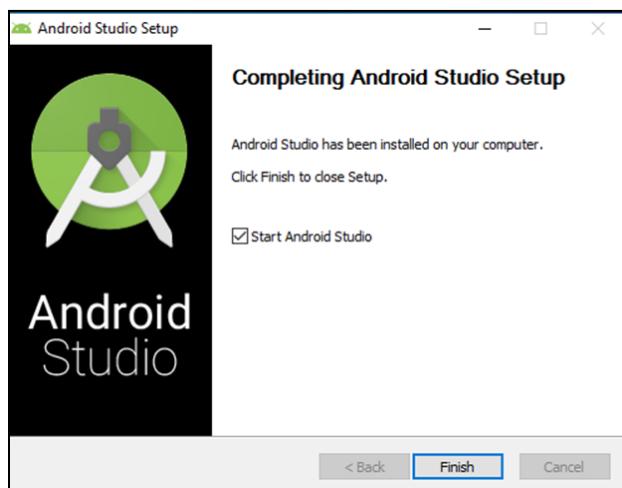
**Figure 5.3.1f** Android Studio setup interface

- Once the installation is done, click the “Next” button.



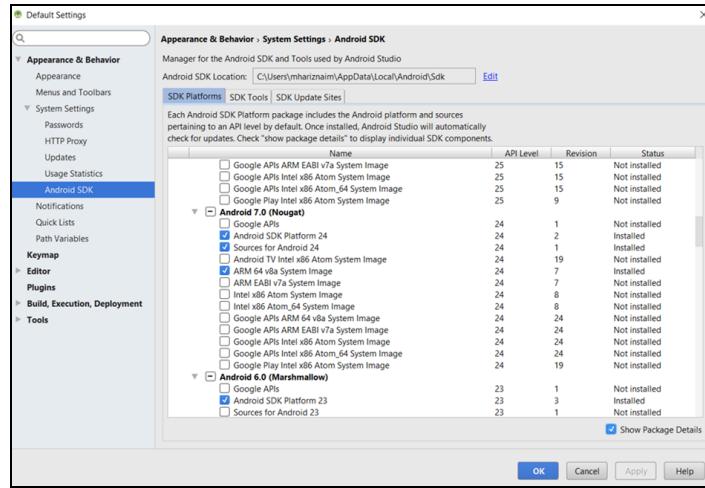
**Figure 5.3.1g** Android Studio setup interface

- Click the “Finish” button to launch the Android Studio.



**Figure 5.3.1h** Android Studio setup interface

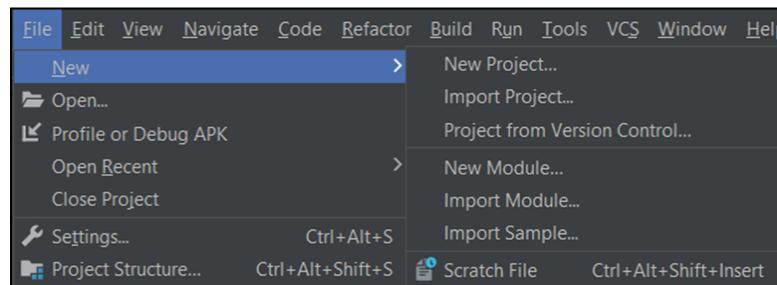
- When Android Studio is opened, click “File” > “Settings” > “Appearance & Behavior” > “System Settings” > “Android SDK” and click the following SDK and click the “OK” button to download.



**Figure 5.3.1i Android Studio interface**

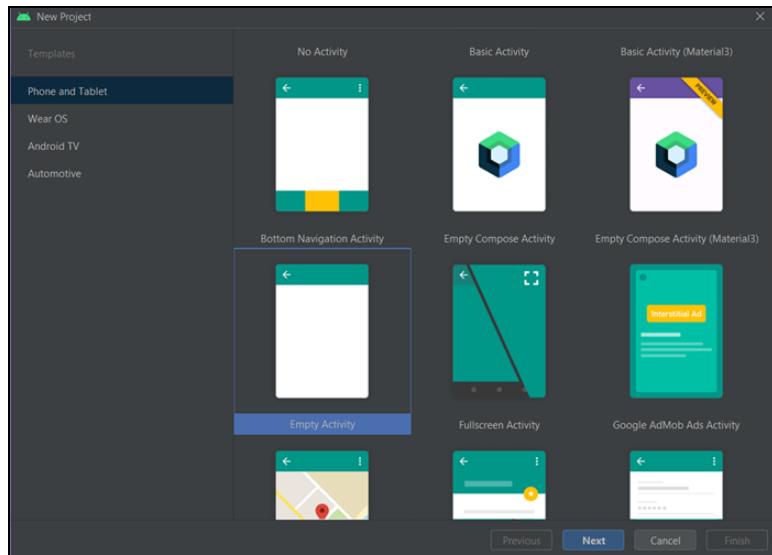
## HOW TO USE ANDROID STUDIO

- Go to File > New > New Project



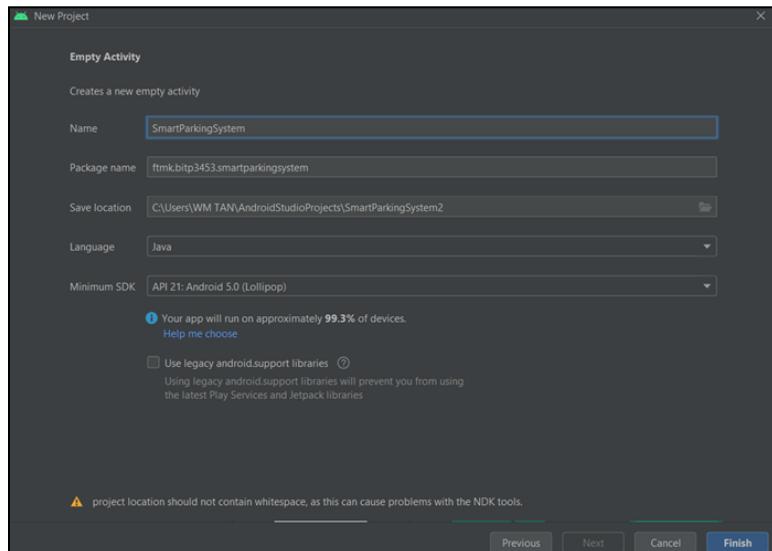
**Figure 5.3.1j Android Studio interface**

- Choose the 'Empty Activity' and click the 'Next' button.

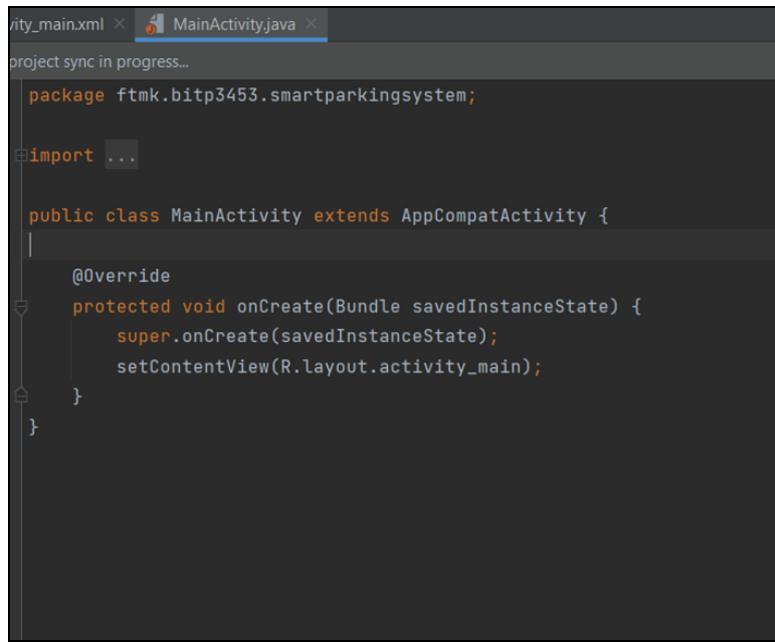


**Figure 5.3.1k** Android Studio interface

- Give the project a name and click ‘Finish’ . A project will be created together with the activity .xml file.



**Figure 5.3.1l** Android Studio interface



A screenshot of the Android Studio interface. The top bar shows tabs for 'activity\_main.xml' and 'MainActivity.java', with 'MainActivity.java' currently selected. A status bar below the tabs indicates 'project sync in progress...'. The main area displays the Java code for the MainActivity class:

```
package ftmk.bitp3453.smartparkingsystem;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Figure 5.3.1m Android Studio interface

### 5.3.2 Installation and Setup of Visual Studio Code

- Download the exe file for Visual Studio Code through the link <https://code.visualstudio.com/download> and click on the Windows.

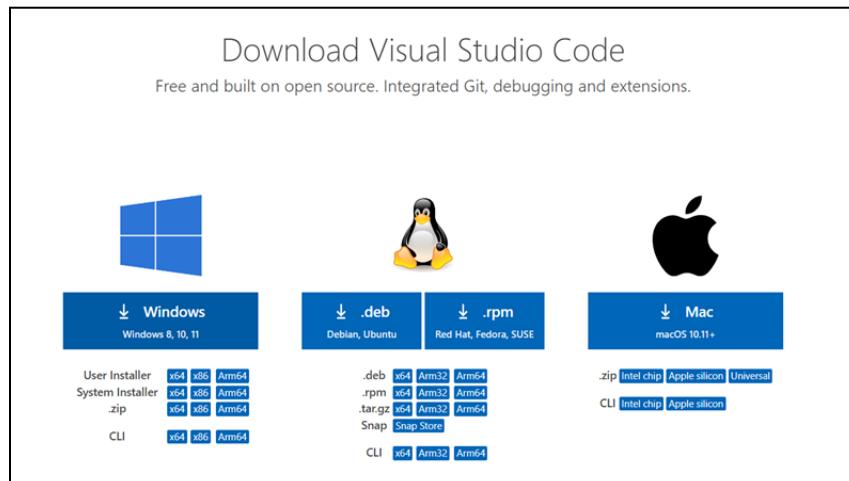


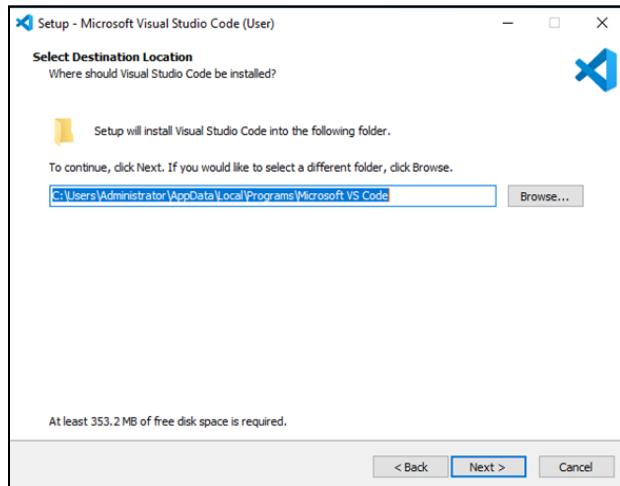
Figure 5.3.2a Download Visual Studio Code interface

- Run the exe file, accept the agreement and click the “Next” button.



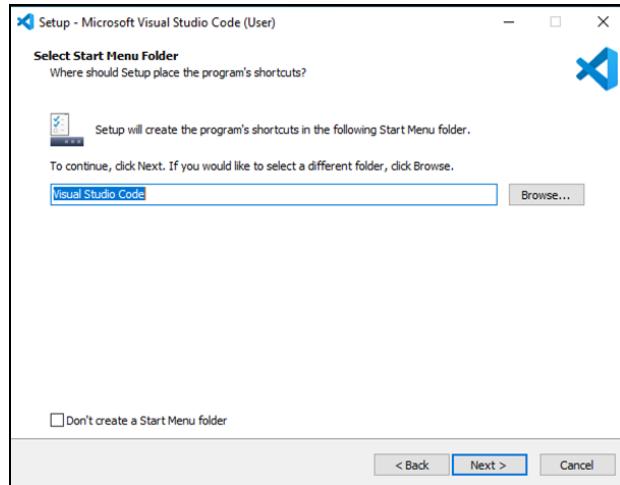
**Figure 5.3.2b Setup Visual Studio Code interface**

- Choose the destination location and click the “Next” button.



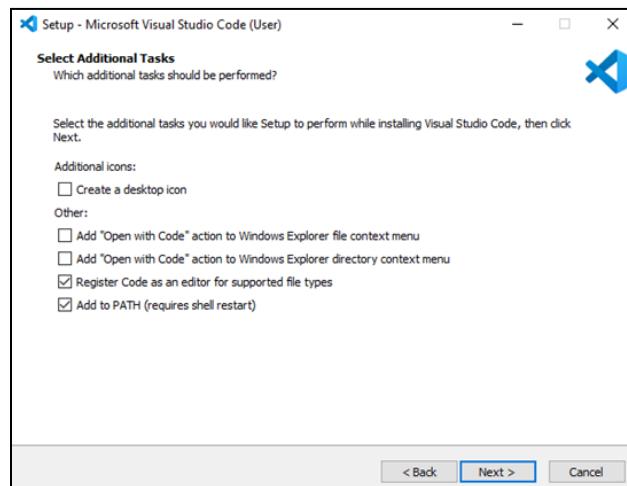
**Figure 5.3.2c Setup Visual Studio Code interface**

- Choose whether to create the Start Menu folder and give a name to it if yes, then click the “Next” button.



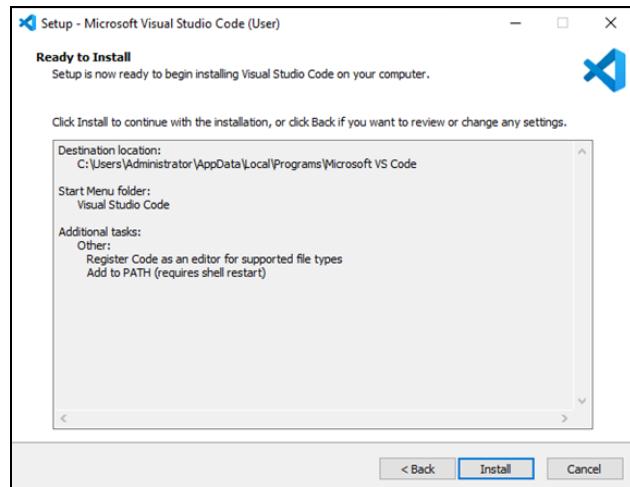
**Figure 5.3.2d** Setup Visual Studio Code interface

- Select the additional tasks if needed and click 'Next'.



**Figure 5.3.2e** Setup Visual Studio Code interface

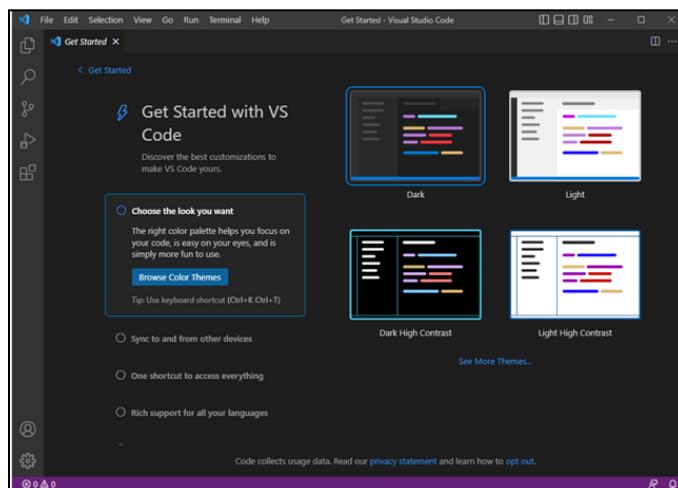
- Click the 'Install' button.



**Figure 5.3.2f Setup Visual Studio Code interface**

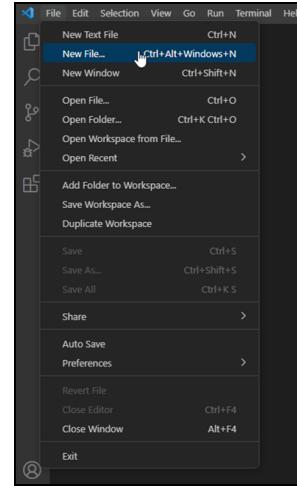
### **HOW TO USE VISUAL STUDIO CODE (VS CODE)**

- Open the VS Code



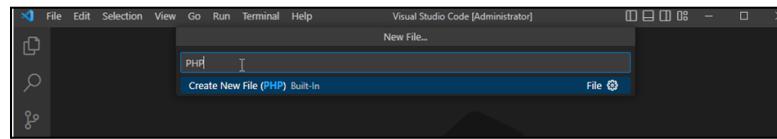
**Figure 5.3.2g VS Code interface**

- Choose File > New File to create a new file.



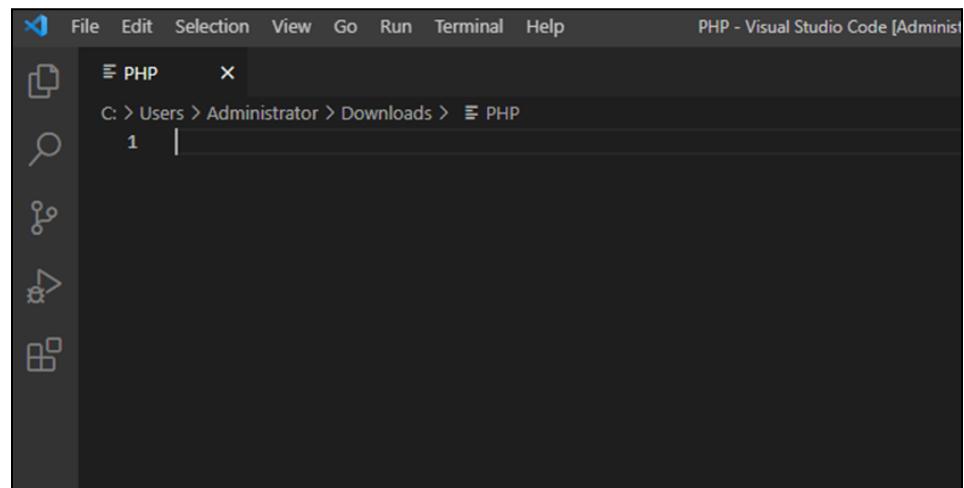
**Figure 5.3.2h VS Code interface**

- A dropdown menu will be shown, insert the type of file to use, then press enter.



**Figure 5.3.2i VS Code interface**

- The file is created and can start coding.



**Figure 5.3.2j VS Code interface**

- In order to connect to MySQL server, the code above is needed before running any SQL queries.

```
<?php
require "DataBase.php";

$servername = 'localhost';
$username = 'root';
$password = '';
$dbname = 'smartparkingsystem';

$connect = mysqli_connect($servername, $username, $password, $dbname);
```

**Figure 5.3.2k VS Code interface**

- This is one example.

```
<?php
require "DataBase.php";

$servername = 'localhost';
$username = 'root';
$password = '';
$dbname = 'smartparkingsystem';

$connect = mysqli_connect($servername, $username, $password, $dbname);

if ($connect) {
    $sql = "select Plate_Number from overtime_vehicle";
    $result = mysqli_query($connect, $sql);

    // create an array
    $scanArr = array();
    if (mysqli_num_rows($result) != 0)
    {
        while($row = mysqli_fetch_assoc($result))
        {
            $scanArr[] = $row;
        }

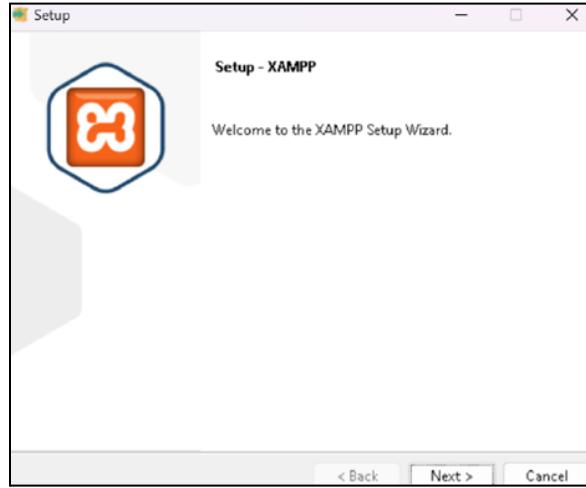
        echo json_encode($scanArr);
    } else echo "No overtime yet";
} else echo "Error: Database connection";

?>
```

**Figure 5.3.2l VS Code interface**

### 5.3.3 Installation and Setup of XAMPP

- Run the .exe file to install XAMPP.



**Figure 5.3.3a Setup XAMPP interface**

- For the installation, just follow the default.
- After the installation, the folder ‘xampp’ will be created at your chosen drive.

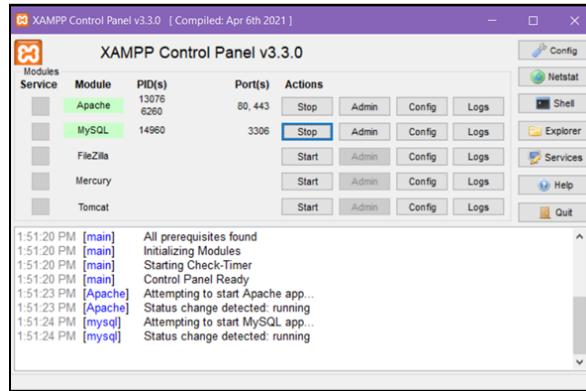
Name	Date modified	Type	Size
FFOutput	07-Feb-20 12:01 PM	File folder	
Log	15-Jan-23 7:53 PM	File folder	
Microsoft	09-May-21 11:15 PM	File folder	
MyDrivers	07-Oct-20 1:21 PM	File folder	
PerfLogs	07-Dec-19 5:14 PM	File folder	
Program Files	07-Jan-23 9:19 PM	File folder	
Program Files (x86)	15-Jan-23 7:53 PM	File folder	
Python310	10-Jul-22 3:04 PM	File folder	
QMDownload	05-Feb-20 5:05 PM	File folder	
Riot Games	07-Jan-23 9:12 PM	File folder	
SystemID	21-Feb-21 6:11 AM	File folder	
Temp	25-Sep-20 5:01 PM	File folder	
Users	18-Mar-21 2:00 AM	File folder	
Windows	11-Jan-23 7:53 AM	File folder	
Windows10Upgrade	23-Feb-21 1:49 PM	File folder	
xampp	13-Oct-22 10:53 AM	File folder	
下载目录	11-Aug-21 6:30 PM	File folder	
迅雷下载	09-May-22 2:59 AM	File folder	
迅雷云盘	20-Jan-23 4:42 AM	File folder	

**Figure 5.3.3b List of folders in chosen drive**

- The database created and PHP written should be inside the “htdocs” folder in “xampp” to establish the connection between the system and MySQL database.

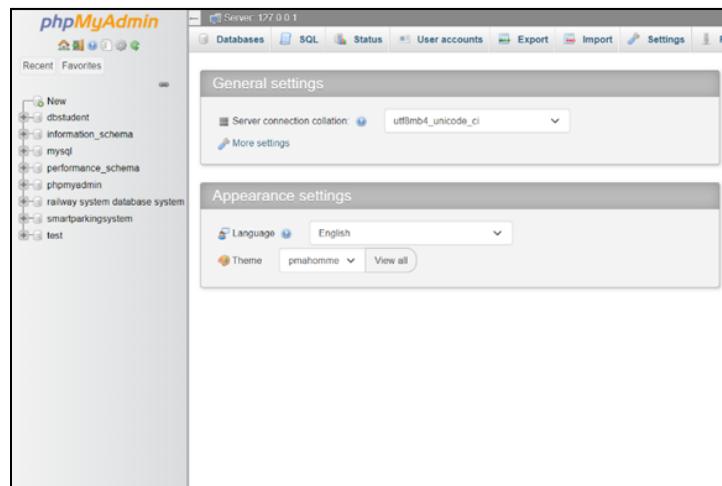
## **HOW TO USE XAMPP**

- When opening the XAMPP interface, click on “Start” for both Apache and MySQL to start a web hosting service locally.



**Figure 5.3.3c XAMPP interface**

- Click on the “Admin” button of the MySQL to open the phpMyAdmin.



**Figure 5.3.3d phpMyAdmin interface**

## 5.4 Security Characteristics

No.	Module	Security Characteristics	Description
1	Registration	Re-type the same chosen password.	User needs to re-type the password to ensure that the same password is entered. A warning will appear if the password is not the same.
2	Registration, Login	Password is hidden behind the asterisk.	The password that the user enters will be hidden behind the asterisk. But, the user has the choice to display the password by clicking the 'eye' icon.
3	Registration, Login	Hash Code for password	The password will become a hash code when entered into the database.
4	QR Code (Third Scan)	Advanced Encryption Standard (AES) encryption.	The QR Code shown to the user will be a license plate that is encrypted using the AES encryption.
5	QR Code (Retrieve Vehicle)	Add string to the original QR Code.	The QR Code for retrieval of the vehicle will be further encrypted with a set string.

**Table 5.4 Security Characteristics**

## 5.5 Output of Implementation

This is the flow of the final output of the system implementation

### 5.5.1 Smart Parking System Mobile Application

#### 5.5.1.1 Account Registration

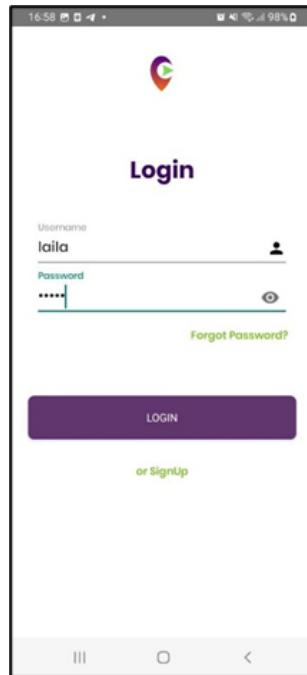
Users need to create an account before login into the application. Figure 5.5.1.1 shows an account created by a user before login into the application.



Figure 5.5.1.1 Account Registration Interface

#### 5.5.1.2 Login

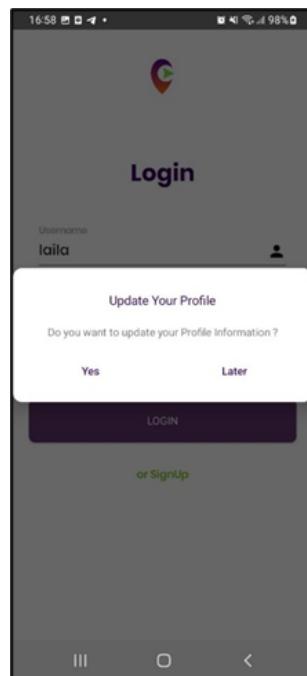
Figure below shows the login interface of the user after registration of an account.



**Figure 5.5.1.2 Login Interface**

### 5.5.1.3 Dialogue Update Profile

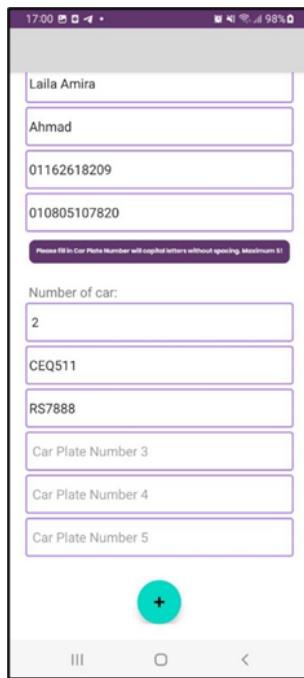
After the user successfully logs into the application, an Update Profile dialogue will be shown in figure below.



**Figure 5.5.1.3 Update Profile Dialogue**

#### **5.5.1.4 Update Profile**

In this page, users need to enter their personal information and car plate number to be registered into the system.



**Figure 5.5.1.4 Update Profile Interface**

#### **5.5.1.5 My Account Profile**

After the registration process, the user's account profile will be updated as shown in figure below.



**Figure 5.5.1.5 My Account Profile**

### 5.5.1.6 Google Maps

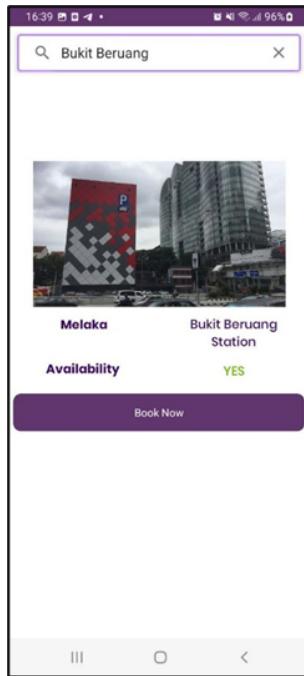
Users can start booking for their vehicles by entering the destination in Google Maps as shown below. Currently, Smart Parking System is only registered in three places which are Bukit Beruang, MITC Melaka and Dataran Pahlawan.



**Figure 5.5.1.6 Google Maps**

### **5.5.1.7 Booking Page**

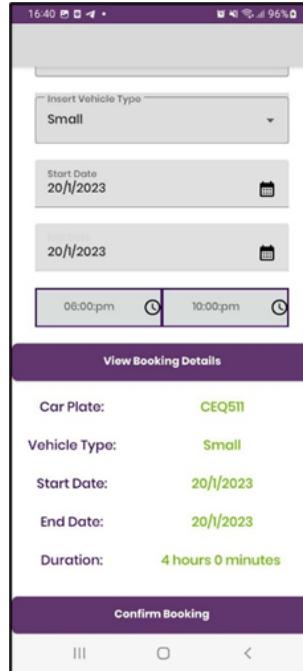
Users can press the ‘Book Now’ button to continue the booking process.



**Figure 5.5.1.7 Booking Interface**

### **5.5.1.8 Booking Requirements**

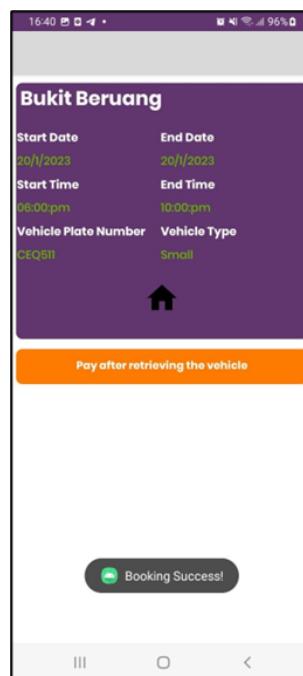
Figure below shows the booking requirements before users press the ‘Confirm Booking’ button.



**Figure 5.5.1.8 Booking Requirements**

### 5.5.1.9 Booking Details

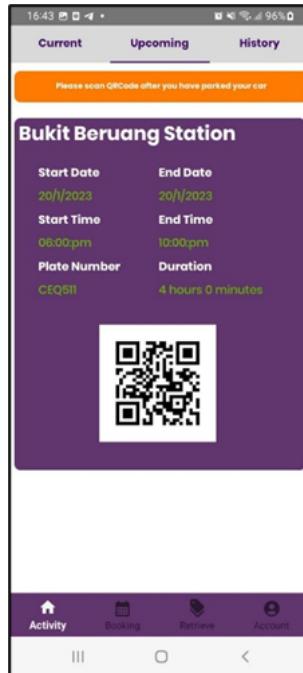
Figure below shows the successful booking details after users have done booking.



**Figure 5.5.1.9 Booking Details**

### **5.5.1.10 Upcoming Page**

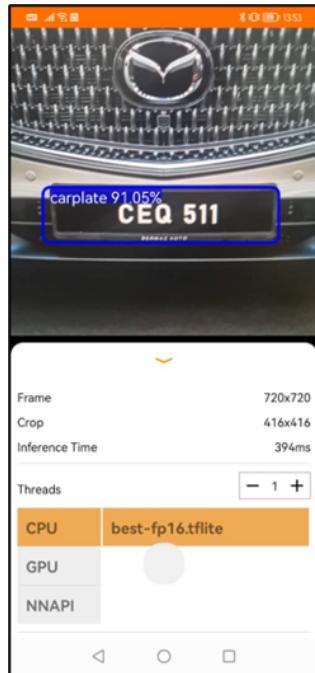
Figure below shows the refresh Upcoming Page after users have done booking.



**Figure 5.5.1.10 Upcoming Page**

### **5.5.1.11 First and Second Scanning**

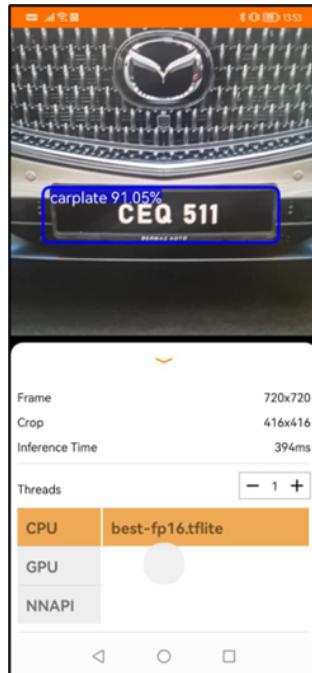
Figure below shows the first scanning before the vehicle passes the security guard and second scanning before entering the lift.



**Figure 5.5.1.11 First and Second Scanning**

### 5.5.1.12 Third Scanning

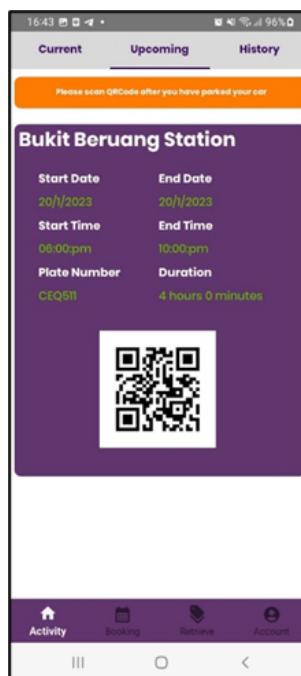
After the User has done the first scanning at the guard house, the car plate needs to be scanned again for the second time in front of the lift of the building. Next, after the lift has already closed, users need to scan QR Code to get the “parked” status.



**Figure 5.5.1.12 Scanning Parked**

### 5.5.1.13 Upcoming Page after Parking

Figure below shows the Upcoming Page after users have parked their car. Users need to scan the QR Code at the scanner in the lift to change the status to “parked”.



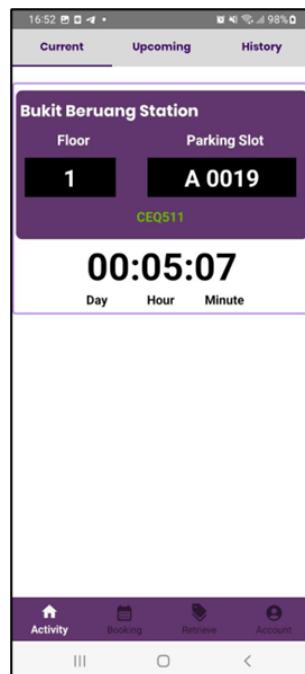
**Figure 5.5.1.13 Upcoming Page after Parking**

#### 5.5.1.14 QR Code Scanning



**Figure 5.5.1.14 QR Code Scanning**

#### 5.5.1.15 Current Page after Parking



**Figure 5.5.1.15 Current Page after Parking**

#### **5.5.1.16 Retrieve Page after Parking**

Figure 5.5.5.15 and figure below shows both the Current and Retrieve page that will appear after the status has changed to “parked”. To retrieve the car, the user needs to pay first before scanning the QR Code.



**Figure 5.5.1.16 Retrieve Page after Parking**

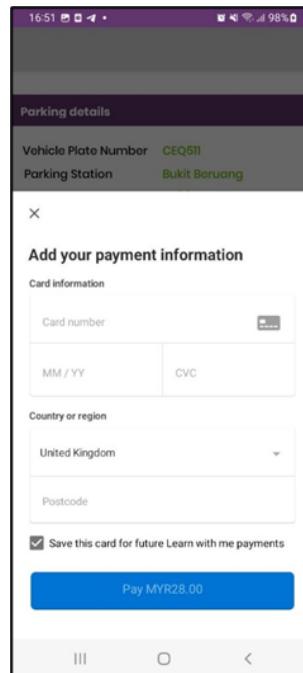
#### **5.5.1.17 Payment Details**

Figure below shows the details of the payment that should be paid by the user.



**Figure 5.5.1.17 Payment Details**

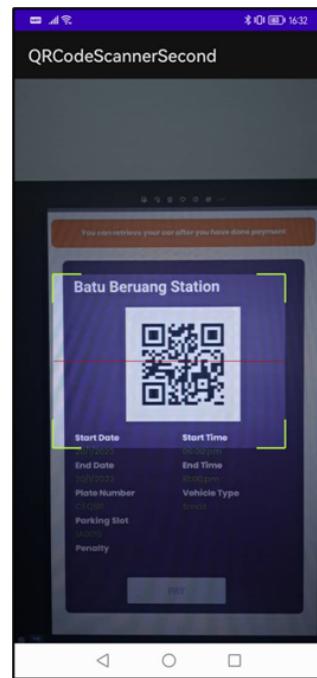
### 5.5.1.18 Payment Activity



**Figure 5.5.1.18 Payment Activity**

### 5.5.1.19 Scanning QR Code at Retrieve

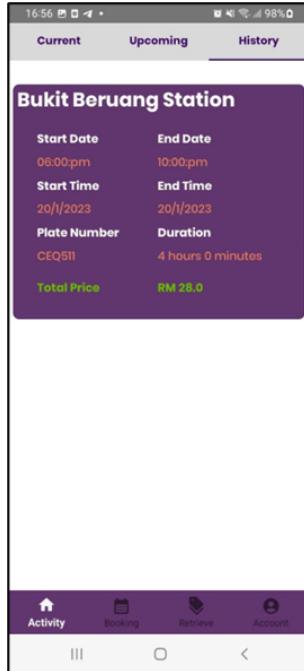
After the user has done the payment in Figure 5.5.5.18, the user needs to scan the QR Code at the scanner to retrieve the car.



**Figure 5.5.1.19 Scanning QR Code at Retrieve**

#### **5.5.1.20 History after Retrieve**

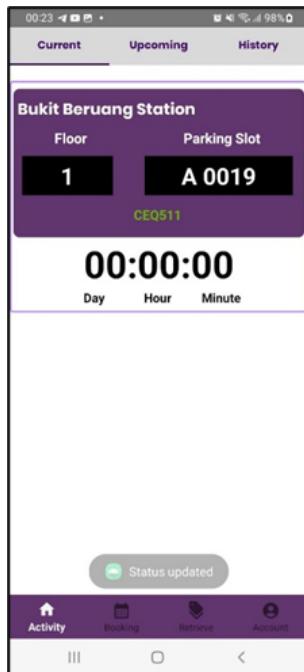
Figure below shows the History Page after the user has completely Retrieved the car and went back home.



**Figure 5.5.1.20 History after Retrieve**

### 5.5.1.21 Penalty Current Page

This Penalty happened when User did not retrieve their car after the countdown for the current page became zero just shown in figure below.



**Figure 5.5.1.21 Penalty Current Page**

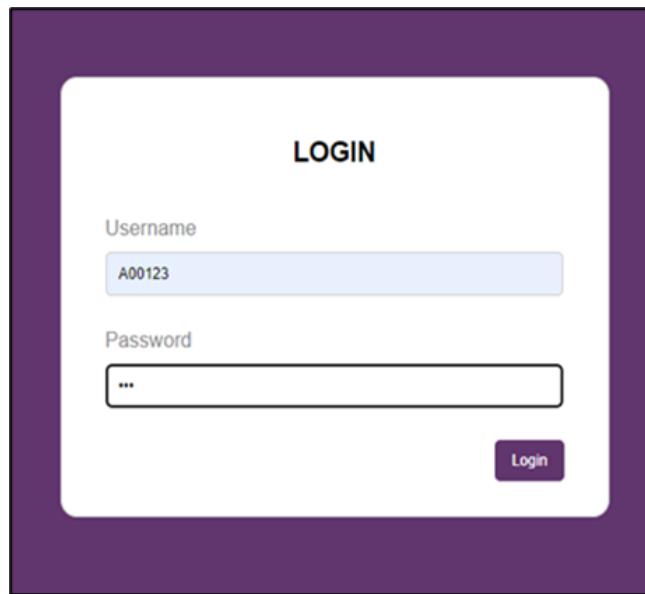
### 5.5.1.22 Penalty Payment



Figure 5.5.1.22 Penalty Payment

## 5.5.2 Admin Website

### 5.5.2.1 Admin Login

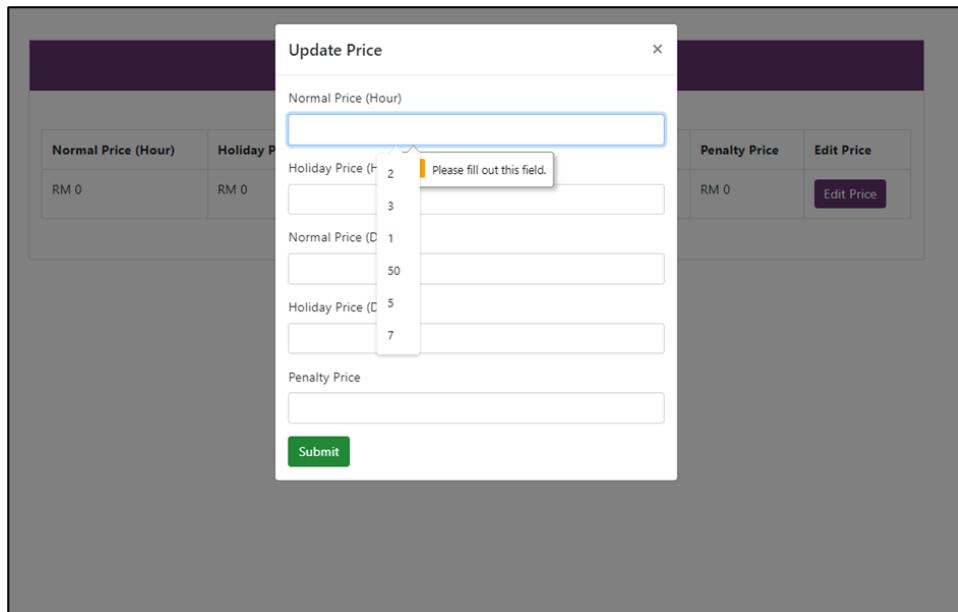


The image shows a login interface titled "LOGIN". It features two input fields: "Username" containing "A00123" and "Password" containing three dots (...). A "Login" button is located at the bottom right of the form area.

Figure 5.5.2.1 Admin Login Page

### 5.5.2.2 Price Update Table

Figure below shows the ways to update the price table by inserting the information details.



The image displays a modal window titled "Update Price" over a table. The table has columns for "Normal Price (Hour)" and "Holiday Price". The first row shows "RM 0" in both columns. The second row shows "RM 0" in the "Normal Price (Hour)" column and a dropdown menu in the "Holiday Price" column, which is currently set to "2". A tooltip "Please fill out this field." is visible near the dropdown. The third row shows "1" in the "Normal Price (Hour)" column and a dropdown menu in the "Holiday Price" column, which is currently set to "50". The fourth row shows "5" in the "Normal Price (Hour)" column and a dropdown menu in the "Holiday Price" column, which is currently set to "7". Below the table is a "Penalty Price" input field and a "Submit" button.

Figure 5.5.2.2 Price Update Table

### 5.5.2.3 History Review

Figure below is the interface of the history review of the admin page.

History Review					
From Date		To Date		Click to Filter	
ID	Booking_ID	Payment_Date	Total	Status	
10	B00009	2023-01-17 17:10:50	0.00	Not Paid	
11	B00010	2023-01-17 17:10:50	30.00	Paid	
9	B00008	2023-01-15 17:10:50	0.00	Not Paid	
7	B00006	2023-01-14 17:10:50	0.00	Not Paid	
8	B00007	2023-01-14 17:10:50	0.00	Not Paid	
5	B00004	2023-01-13 17:10:50	13.00	Paid	
6	B00005	2023-01-13 17:10:50	0.00	Not Paid	
2	B00001	2023-01-12 17:10:50	0.00	Not Paid	
3	B00002	2023-01-12 17:10:50	0.00	Not Paid	
4	B00003	2023-01-12 17:10:50	5.00	Paid	
Showing 1 to 10 of 10 entries			Previous	1	Next
<a href="#">Go back!</a>					

Figure 5.5.2.3 History Review Page

## 5.6 Conclusion

In conclusion, system testing must be done carefully to ensure that the implemented system is constructed correctly and successfully.

## **CHAPTER 6: SYSTEM TESTING**

### **6.1 Introduction**

System Testing is a level of testing that verifies a whole integrated and a complete finished integrated system product. The objective of a system testing is to assess and evaluate the complete end-to-end system requirements and specifications. System testing verifies and checks to ensure an application performs and works as intended. It verifies various types of testing such as the functionality and non-functionality of an application. For example, the testing might validate the user input and intended output throughout the system.

#### **6.1.1 Black Box Testing**

Black Box Testing which also known as Behavioral Testing is a system or an application testing method to test the functionalities of the software without having to go through the structure of the internal coding, the programming language that been used, the implementation details and also the library that have been used to develop the application. This Black Box Testing method is based on the software specifications and requirements entirely and only focus on the input, and the output of the system or the application. Equivalence Class Testing and Boundary Value Testing are the examples of Black Box Testing.

#### **6.1.2 White Box Testing**

As for White Box Testing, opposite to the Black Box Testing, this testing method is focus on the system internal coding structure, the classes and objects used in the coding, and also the design of the coding to test, validate and verify the flow of the input and output to improve the security and usability of the system. Other names for the White Box Testing are Code-based testing, Glass Box Testing because the code is visible to the testers.

## 6.2 Test Result Analysis (Functional Requirement)

### 6.2.1 Test Case 1: Successful User Login

<b>Test Case ID</b>	TCUL01
<b>Functional Requirement</b>	Login into the system
<b>Purpose</b>	To test whether a user can login into the system with the right input or not.
<b>Pre-Condition</b>	Users need to sign up for an account before login into the system.
<b>Test Data</b>	Username: laila Password: ***** {laila123 – correct password}
<b>Steps</b>	1) Enter Username and Password 2) Click Login button when it is enabled
<b>Expected Result</b>	A dialog to update profile will be popped out.
<b>Actual Result</b>	A dialog to update profile will be popped out
<b>Pass/Fail</b>	Pass

**Table 6.2.1 Test Case of Successful User Login**

### 6.2.2 Test Case 2: Unsuccessful User Login

<b>Test Case ID</b>	TCUL02
<b>Functional Requirement</b>	Login into the system
<b>Purpose</b>	To test whether a user can login into the system with wrong input or not.

<b>Pre-Condition</b>	Users need to sign up for an account before login into the system.
<b>Test Data</b>	Username: laila Password: ***** {loila123 – incorrect password}
<b>Steps</b>	1) Enter Username and wrong Password 2) Click Login button when it is enabled
<b>Expected Result</b>	A toast of ‘Username or Password Wrong’ will be popped out.
<b>Actual Result</b>	A toast of ‘Username or Password Wrong’ will be popped out.
<b>Pass/Fail</b>	Pass

**Table 6.2.2 Test Case of Unsuccessful User Login**

### 6.2.3 Test Case 3: Successful Account Registration

<b>Test Case ID</b>	TCAR01
<b>Functional Requirement</b>	Account registration
<b>Purpose</b>	To test whether a user can register an account before login into the system.
<b>Pre-Condition</b>	Users need to have an existing email.
<b>Test Data</b>	Email: laila@gmail.com Password: laila123 Confirmed Password: laila123 Username : laila

<b>Steps</b>	<ol style="list-style-type: none"> <li>1) Enter an existing email in Email column</li> <li>2) Choose any strong password for Password column</li> <li>3) Retype the password in Confirmed Password column</li> <li>4) Enter a unique Username</li> <li>5) Click terms and condition box otherwise users cannot create an account</li> <li>6) Click Create Account button when it is enabled</li> </ol>
<b>Expected Result</b>	A login page will appear with toast of “ Sign Up Success”
<b>Actual Result</b>	A login page will appear with toast of “ Sign Up Success”
<b>Pass/Fail</b>	Pass

**Table 6.2.3 Test Case of Successful Account Registration**

#### **6.2.4 Test Case 4: Unsuccessful Account Registration**

<b>Test Case ID</b>	TCAR02
<b>Functional Requirement</b>	Account registration
<b>Purpose</b>	To test whether a user can register multiple username for an account before login into the system.
<b>Pre-Condition</b>	Users need to have an existing email.
<b>Test Data</b>	Email: amiralaila@gmail.com Password: amira123 Confirmed Password: amira123 Username : laila {Username already exist}

<b>Steps</b>	<ol style="list-style-type: none"> <li>1) Enter an existing email in Email column</li> <li>2) Choose any strong password for Password column</li> <li>3) Retype the password in Confirmed Password column</li> <li>4) Enter an existing Username</li> <li>5) Click terms and condition box otherwise users cannot create an account</li> <li>6) Click Create Account button when it is enabled</li> </ol>
<b>Expected Result</b>	A toast of “This Username Exists!” will be popped out.
<b>Actual Result</b>	A toast of “This Username Exists!” will be popped out.
<b>Pass/Fail</b>	Pass

**Table 6.2.4 Test Case of Unsuccessful User Login**

#### **6.2.5 Test Case 5: Create Profile**

<b>Test Case ID</b>	TCCUP01
<b>Functional Requirement</b>	Create a profile
<b>Purpose</b>	To test whether a user can create the profile and it will show in the Account page or not.
<b>Pre-Condition</b>	Users need to login into the system
<b>Test Data</b>	First Name: Laila Amira Last Name: Ahmad Phone Number: 01162618209 IC Number: 010805107820 Number of Car:2 Car Plate Number 1: CEQ511 Car Plate Number 2: RS7888

	Car Plate Number 3: { empty } Car Plate Number 4: { empty } Car Plate Number 5: { empty }
<b>Steps</b>	1) Enter First Name 2) Enter Last Name 3) Enter Phone Number 4) Enter IC Number 5) Enter Number of car to be registered 6) Enter Car Plate following the number of car registered 7) Click button + when it is enabled
<b>Expected Result</b>	The profile account will be filled with the information registered before.
<b>Actual Result</b>	The profile account will be filled with the information registered before.
<b>Pass/Fail</b>	Pass

**Table 6.2.5 Test Case of Create Profile**

#### **6.2.6 Test Case 6: Booking Location Validation**

<b>Test Case ID</b>	TCB01
<b>Function Requirement</b>	Place a Booking
<b>Purpose</b>	To validate whether the user will be notified that the booking location inserted exists or not.
<b>Pre-Condition</b>	Users need to press the “Booking” button from the bottom navigation bar.
<b>Test Data</b>	Location name: Jasin
<b>Steps</b>	1) Users insert the location name at the searching bar. 2) Enter to search.
<b>Expected Result</b>	A toast message “There is no Smart Parking Available in this location” will be shown to the user.

<b>Actual Result</b>	A toast message “There is no Smart Parking Available in this location” is shown to the user.
<b>Pass/Fail</b>	Pass

**Table 6.2.6 Test Case of Booking Location Validation**

**6.2.7 Test Case 7: Booking Success**

<b>Test Case ID</b>	TCB02
<b>Function Requirement</b>	Place a Booking
<b>Purpose</b>	To validate whether the user will be notified that the booking is successful.
<b>Pre-Condition</b>	Users need to press the “Booking” button from the bottom navigation bar.
<b>Test Data</b>	Location name: Bukit Beruang Insert Car Plate: CEQ511 Insert Vehicle Type: Small Start Date: 20/1/2023 End Date: 20/1/2023 Start Time: 6:00pm End Time: 10:00pm

<b>Steps</b>	<ol style="list-style-type: none"> <li>1) Users insert the location name at the searching bar.</li> <li>2) Enter to search.</li> <li>3) When the location page is shown, click the “Book Now” button.</li> <li>4) User presses the “Insert Car Plate” drop down box.</li> <li>5) User selects the car plate.</li> <li>6) User presses the “Insert Vehicle Type” drop down box.</li> <li>7) User selects the vehicle type.</li> <li>8) User selects the start date and the end date.</li> <li>9) User selects the start time and end time.</li> <li>10) User clicks the “View Booking Details” button.</li> <li>11) User clicks the “Confirm Booking” button.</li> </ol>
<b>Expected Result</b>	A booking details interface will be shown together with the toast message “Booking Success”.
<b>Actual Result</b>	A booking details interface is shown together with the toast message “Booking Success”.
<b>Pass/Fail</b>	Pass

**Table 6.2.7 Test Case of Booking Success**

#### **6.2.8 Test Case 8: Successful Vehicle Plate Number First Scanning**

<b>Test Case ID</b>	TCFS01
<b>Function Requirement</b>	Recognise the vehicle number plate for booking validation.
<b>Purpose</b>	To validate whether the scanned vehicle number plate is the same as the user's vehicle number plate in booking.
<b>Pre-Condition</b>	Users need to make a booking for the vehicle number plate.

<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The first scanner of the vehicle number plate recognition system will produce the “correct” sound.
<b>Actual Result</b>	The first scanner of the vehicle number plate recognition system produces the “correct” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.8 Test Case of Successful Vehicle Plate Number First Scanning**

#### **6.2.9 Test Case 9: Unsuccessful Vehicle Number Plate First Scanning**

<b>Test Case ID</b>	TCFS02
<b>Function Requirement</b>	Recognise the vehicle number plate for booking validation.
<b>Purpose</b>	To validate whether the scanned vehicle number plate is the same as the user's vehicle number plate in booking.
<b>Pre-Condition</b>	Users do not make a booking for the vehicle number plate.
<b>Test Data</b>	Vehicle number plate: RS 7888
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The first scanner of the vehicle number plate recognition system will produce the “error” sound.

<b>Actual Result</b>	The first scanner of the vehicle number plate recognition system produces the “error” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.9 Test Case of Unsuccessful Vehicle Plate Number First Scanning**

**6.2.10 Test Case 10: Vehicle Plate Number First Scanning Validation**

<b>Test Case ID</b>	TCFS03
<b>Function Requirement</b>	Recognise the vehicle number plate for booking validation.
<b>Purpose</b>	To validate whether the vehicle number plate is already scanned to notify users regarding multiple scanning.
<b>Pre-Condition</b>	User's vehicle has already been scanned and successful.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The first scanner of the vehicle number plate recognition system will produce the “double error” sound.
<b>Actual Result</b>	The first scanner of the vehicle number plate recognition system produces the “double error” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.10 Test Case of Vehicle Plate Number First Scanning Validation**

**6.2.11 Test Case 11: Successful Vehicle Plate Number Second Scanning**

<b>Test Case ID</b>	TCSS01
<b>Function Requirement</b>	Recognise the vehicle number plate to retrieve the assigned parking lot details.
<b>Purpose</b>	To validate whether the scanned vehicle number plate has already passed the First Scanning.
<b>Pre-Condition</b>	Users need to pass the First Scanning.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The first scanner of the vehicle number plate recognition system will produce the “correct” sound.
<b>Actual Result</b>	The first scanner of the vehicle number plate recognition system produces the “correct” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.11 Test Case of Successful Vehicle Plate Number Second Scanning**

**6.2.12 Test Case 12: Unsuccessful Vehicle Plate Number Second Scanning**

<b>Test Case ID</b>	TCSS02
---------------------	--------

<b>Function Requirement</b>	Recognise the vehicle number plate to retrieve the assigned parking lot details.
<b>Purpose</b>	To validate whether the scanned vehicle number plate has already passed the First Scanning.
<b>Pre-Condition</b>	Users need to pass the First Scanning.
<b>Test Data</b>	Vehicle number plate: RS 7888
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The second scanner of the vehicle number plate recognition system will produce the “error” sound.
<b>Actual Result</b>	The second scanner of the vehicle number plate recognition system produces the “error” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.12 Test Case of Unsuccessful Vehicle Plate Number Second Scanning**

#### **6.2.13 Test Case 13: Vehicle Plate Number Second Scanning Validation**

<b>Test Case ID</b>	TCSS03
<b>Function Requirement</b>	Recognise the vehicle number plate to retrieve the assigned parking lot details.
<b>Purpose</b>	To validate whether the vehicle number plate is already scanned to notify users regarding multiple scanning.

<b>Pre-Condition</b>	User's vehicle has already been scanned and successful.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The second scanner of the vehicle number plate recognition system will produce the “double error” sound.
<b>Actual Result</b>	The second scanner of the vehicle number plate recognition system produces the “double error” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.13 Test Case of Vehicle Plate Number Second Scanning Validation**

#### **6.2.14 Test Case 14: Successful Upcoming QR Code Scanning**

<b>Test Case ID</b>	TCQRF01
<b>Function Requirement</b>	Recognise the QR Code to park the vehicle.
<b>Purpose</b>	To validate whether the vehicle number plate has already passed the second scan.
<b>Pre-Condition</b>	User's vehicle needs to pass the second scan.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) User clicks the “Activity” of the bottom navigation bar. 2) User clicks the “Upcoming” of the top navigation bar.

	3) User scans the QR Code at the QR Code scanner.
<b>Expected Result</b>	The QR Code scanner will produce the “correct” sound.
<b>Actual Result</b>	The QR Code scanner produces the “correct” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.14 Test Case of Successful Upcoming QR Code Scanning**

**6.2.15 Test Case 15: Unsuccessful Upcoming QR Code Scanning**

<b>Test Case ID</b>	TCQRF02
<b>Function Requirement</b>	Recognise the QR Code to park the vehicle.
<b>Purpose</b>	To validate whether the QR Code is already scanned to notify users regarding multiple scanning.
<b>Pre-Condition</b>	User's Upcoming QR Code has already been scanned and successful.
<b>Test Data</b>	Vehicle number plate: RS 7888
<b>Steps</b>	1) User clicks the “Activity” of the bottom navigation bar. 2) User clicks the “Upcoming” of the top navigation bar. 3) User scans the QR Code at the QR Code scanner.
<b>Expected Result</b>	The QR Code scanner will produce the “error” sound.
<b>Actual Result</b>	The QR Code scanner produces the “error” sound.

Pass/Fail	Pass

**Table 6.2.15 Test Case of Unsuccessful Upcoming QR Code Scanning**

**6.2.16 Test Case 16: Upcoming QR Code Scanning Validation**

<b>Test Case ID</b>	TCQRF03
<b>Function Requirement</b>	Recognise the QR Code to park the vehicle.
<b>Purpose</b>	To validate whether the vehicle number plate has already passed the second scan.
<b>Pre-Condition</b>	User's vehicle needs to pass the second scan.
<b>Test Data</b>	Vehicle number plate: RS 7888
<b>Steps</b>	<ol style="list-style-type: none"> <li>1) User clicks the “Activity” of the bottom navigation bar.</li> <li>2) User clicks the “Upcoming” of the top navigation bar.</li> <li>3) User scans the QR Code at the QR Code scanner.</li> </ol>
<b>Expected Result</b>	The QR Code scanner will produce the “double error” sound.
<b>Actual Result</b>	The QR Code scanner produces the “double error” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.16 Test Case of Upcoming QR Code Scanning Validation**

#### **6.2.17 Test Case 17: Current Page (No vehicle is parked)**

<b>Test Case ID</b>	TCCP01
<b>Function Requirement</b>	Show current parked vehicle details.
<b>Purpose</b>	To validate whether the vehicle is already parked.
<b>Pre-Condition</b>	User didn't scan the Upcoming QR Code at the Upcoming QR Code scanner.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) User clicks the “Activity” of the bottom navigation bar. 2) User clicks the “Current” of the top navigation bar.
<b>Expected Result</b>	Nothing to show the parking details and countdown timer of the vehicle.
<b>Actual Result</b>	Nothing to show the parking details and countdown timer of the vehicle.
<b>Pass/Fail</b>	Pass

**Table 6.2.17 Test Case of Current Page (No Vehicle Parked)**

#### **6.2.18 Test Case 18: Current Page(Vehicle is parked)**

<b>Test Case ID</b>	TCCP02
<b>Function Requirement</b>	Show current parked vehicle details.

<b>Purpose</b>	To validate whether the vehicle is already parked.
<b>Pre-Condition</b>	User already scanned the Upcoming QR Code at the Upcoming QR Code scanner.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) User clicks the “Activity” of the bottom navigation bar. 2) User clicks the “Current” of the top navigation bar.
<b>Expected Result</b>	Parking details and countdown timer for the parked vehicle is shown.
<b>Actual Result</b>	Parking details and countdown timer for the parked vehicle is shown.
<b>Pass/Fail</b>	Pass

**Table 6.2.18 Test Case of Current Page (Vehicle Parked)**

#### **6.2.19 Test Case 19: Penalty Function**

<b>Test Case ID</b>	TCP01
<b>Function Requirement</b>	Penalty to overtime vehicle.
<b>Purpose</b>	To validate whether the vehicle passed the booking end time.

<b>Pre-Condition</b>	The countdown timer goes to 00:00:00. User pressed the “Pay” button for that vehicle in the Retrieve page.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	<ol style="list-style-type: none"> <li>1) User clicks the “Activity” of the bottom navigation bar.</li> <li>2) User clicks the “Retrieve” of the bottom navigation bar.</li> <li>3) User clicks the “Pay” button of the vehicle.</li> </ol>
<b>Expected Result</b>	The “Penalty” in the parking details page will be set to “Yes”.
<b>Actual Result</b>	The “Penalty” in the parking details page is set to “Yes”.
<b>Pass/Fail</b>	Pass

**Table 6.2.19 Test Case of Penalty Function**

#### **6.2.20 Test Case 20: Successful Payment**

<b>Test Case ID</b>	TCP01
<b>Function Requirement</b>	Penalty to overtime vehicle.
<b>Purpose</b>	To validate whether the vehicle passed the booking end time.
<b>Pre-Condition</b>	The countdown timer goes to 00:00:00. User pressed the “Pay” button for that vehicle in the Retrieve page.
<b>Test Data</b>	Vehicle number plate: CEQ 511

<b>Steps</b>	1) User clicks the “Activity” of the bottom navigation bar. 2) User clicks the “Retrieve” of the bottom navigation bar. 3) User clicks the “Pay” button of the vehicle.
<b>Expected Result</b>	The “Penalty” in the parking details page will be set to “Yes”.
<b>Actual Result</b>	The “Penalty” in the parking details page is set to “Yes”.
<b>Pass/Fail</b>	Pass

**Table 6.2.20 Test Case of Successful Payment**

#### **6.2.21 Test Case 21: Successful Retrieve QR Code Scanning**

<b>Test Case ID</b>	TCQRS01
<b>Function Requirement</b>	Recognise the QR Code to retrieve the vehicle.
<b>Purpose</b>	To validate whether the vehicle number plate booking is already paid.
<b>Pre-Condition</b>	Users already pay for the vehicle booking.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) User clicks the “Retrieve” of the bottom navigation bar. 2) User scans the QR Code at the Retrieve QR Code scanner.
<b>Expected Result</b>	The Retrieve QR Code scanner will produce the “correct” sound.
<b>Actual Result</b>	The Retrieve QR Code scanner produces the “correct” sound.

<b>Pass/Fail</b>	Pass
------------------	------

**Table 6.2.21 Test Case of Successful Retrieve QR Code Scanning**

**6.2.22 Test Case 22: Unsuccessful Retrieve QR Code Scanning**

<b>Test Case ID</b>	TCQRS02
<b>Function Requirement</b>	Recognise the QR Code to retrieve the vehicle.
<b>Purpose</b>	To validate whether the vehicle number plate booking is already paid.
<b>Pre-Condition</b>	User didn't pay for the vehicle booking.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) User clicks the “Retrieve” of the bottom navigation bar. 2) User scans the QR Code at the Retrieve QR Code scanner.
<b>Expected Result</b>	The Retrieve QR Code scanner will produce the “error” sound.
<b>Actual Result</b>	The Retrieve QR Code scanner produces the “error” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.22 Test Case of Unsuccessful Retrieve QR Code Scanning**

**6.2.23 Test Case 23: Successful Vehicle Plate Number Third Scanning**

<b>Test Case ID</b>	TCTS01
---------------------	--------

<b>Function Requirement</b>	Recognise the vehicle number plate to make sure the vehicle is retrieved.
<b>Purpose</b>	To validate whether the scanned vehicle number plate is already retrieved.
<b>Pre-Condition</b>	Users need to be paid and scanned the Retrieve QR Code.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The third scanner of the vehicle number plate recognition system will produce the “correct” sound.
<b>Actual Result</b>	The third scanner of the vehicle number plate recognition system produces the “correct” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.23 Test Case of Successful Vehicle Plate Number Third Scanning**

#### **6.2.24 Test Case 24: Vehicle Plate Number Third Scanning Validation**

<b>Test Case ID</b>	TCTS02
<b>Function Requirement</b>	Recognise the vehicle number plate to make sure the vehicle is retrieved.
<b>Purpose</b>	To validate whether the vehicle number plate is already scanned to notify users regarding multiple scanning.

<b>Pre-Condition</b>	User's vehicle has already been scanned and successful.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) Scan the vehicle number plate.
<b>Expected Result</b>	The third scanner of the vehicle number plate recognition system will produce the “double error” sound.
<b>Actual Result</b>	The third scanner of the vehicle number plate recognition system produces the “double error” sound.
<b>Pass/Fail</b>	Pass

**Table 6.2.24 Test Case of Vehicle Plate Number Third Scanning Validation**

#### **6.2.25 Test Case 25: History Page (No Vehicle is Retrieved)**

<b>Test Case ID</b>	TCHP01
<b>Function Requirement</b>	Show past booking history.
<b>Purpose</b>	To validate whether the vehicle is retrieved and scanned by the Vehicle number plate Third Scanning.
<b>Pre-Condition</b>	User didn't scan the vehicle plate number at the vehicle plate number at third scanning
<b>Test Data</b>	Vehicle number plate: CEQ 511

<b>Steps</b>	1) User clicks the “Activity” of the bottom navigation bar. 2) User clicks the “History” of the top navigation bar.
<b>Expected Result</b>	Past booking details for that vehicle will not be shown.
<b>Actual Result</b>	Past booking details for that vehicle will not be shown.
<b>Pass/Fail</b>	Pass

**Table 6.2.25 Test Case of History Page (No Vehicle is Retrieved)**

#### **6.2.26 Test Case 26 History Page(Vehicle is retrieved)**

<b>Test Case ID</b>	TCHP02
<b>Function Requirement</b>	Show past booking history.
<b>Purpose</b>	To validate whether the vehicle is retrieved and scanned by the Vehicle number plate Third Scanning.
<b>Pre-Condition</b>	User already scanned the vehicle number plate at the Vehicle number plate Third Scanning.
<b>Test Data</b>	Vehicle number plate: CEQ 511
<b>Steps</b>	1) User clicks the “Activity” of the bottom navigation bar. 2) User clicks the “History” of the top navigation bar.
<b>Expected Result</b>	Past booking details for that vehicle will be shown.
<b>Actual Result</b>	Past booking details for that vehicle are shown.

<b>Pass/Fail</b>	Pass
------------------	------

**Table 6.2.26 Test Case of History Page (Vehicle is Retrieved)**

**6.2.27 Test Case 27: Unsuccessful Website Admin Login**

<b>Test Case ID</b>	TCWL01
<b>Function Requirement</b>	Login into the system
<b>Purpose</b>	To validate whether an admin can login into the system with the right input or not.
<b>Pre-Condition</b>	Admin account already created in database.
<b>Test Data</b>	Username: A0101 Password: 012345
<b>Steps</b>	1) Admin inserts the username at the “Username”. 2) Admin inserts the password at the “Password”. 3) Admin presses the “Login” button.
<b>Expected Result</b>	An error message of “Incorrect username or password” will be shown.
<b>Actual Result</b>	An error message of “Incorrect username or password” is shown.
<b>Pass/Fail</b>	Pass

**Table 6.2.27 Test Case of Unsuccessful Website Admin Login**

#### **6.2.28 Test Case 28: Successful Website Admin Login**

<b>Test Case ID</b>	TCWL02
<b>Function Requirement</b>	Login into the system
<b>Purpose</b>	To validate whether an admin can login into the system with the right input or not.
<b>Pre-Condition</b>	Admin account already created in database.
<b>Test Data</b>	Username: A00123 Password: 123
<b>Steps</b>	1) Admin inserts the username at the “Username”. 2) Admin inserts the password at the “Password”. 3) Admin presses the “Login” button.
<b>Expected Result</b>	The Smart Parking System website will be shown.
<b>Actual Result</b>	The Smart Parking System website is shown.
<b>Pass/Fail</b>	Pass

**Table 6.2.28 Test Case of Successful Website Admin Login**

#### **6.2.29 Test Case 29: Update Price Table**

<b>Test Case ID</b>	TCWUP01
<b>Function Requirement</b>	Update the price of hours and days.

<b>Purpose</b>	To validate whether the admin can update the prices for hours and days.
<b>Pre-Condition</b>	Admin already logged in to the website.
<b>Test Data</b>	Normal Price (Hour): 3 Holiday Price (Hour): 5 Normal Price (Day): 6 Holiday Price (Day): 30 Penalty Price: 30
<b>Steps</b>	<ol style="list-style-type: none"> <li>1) Admin selects the “Price Update” button at the left navigation bar.</li> <li>2) Admin selects the “Edit Price” button.</li> <li>3) Admin inserts the price for “Normal Price (Hour)”.</li> <li>4) Admin inserts the price for “Holiday Price (Hour)”.</li> <li>5) Admin inserts the price for “Normal Price (Day)”.</li> <li>6) Admin inserts the price for “Holiday Price (Day)”.</li> <li>7) Admin inserts the price for “Penalty Price”.</li> <li>8) Admin selects the “Submit” button.</li> </ol>
<b>Expected Result</b>	The prices for the Price Update Table will be updated.
<b>Actual Result</b>	The prices for the Price Update Table are updated.
<b>Pass/Fail</b>	Pass

**Table 6.2.29 Test Case of Update Price Table**

#### **6.2.30 Test Case 30: History Review (Filter Dates)**

<b>Test Case ID</b>	TCWHR01
---------------------	---------

<b>Function Requirement</b>	Review past tickets.
<b>Purpose</b>	To validate whether the admin can review the tickets between selected start date and end date.
<b>Pre-Condition</b>	Admin already logged in to the website.
<b>Test Data</b>	From Date: 01/12/2023 To Date: 01/23/2023
<b>Steps</b>	<ol style="list-style-type: none"> <li>1) Admin selects the “History review” button at the left navigation bar.</li> <li>2) Admin clicks the “From Date” icon.</li> <li>3) Admin selects the starting date.</li> <li>4) Admin clicks the “To Date” icon.</li> <li>5) Admin selects the end date.</li> <li>6) Admin selects the “Filter” button.</li> </ol>
<b>Expected Result</b>	The past ticket between the start date and end date will be shown.
<b>Actual Result</b>	The past ticket between the start date and end date are shown.
<b>Pass/Fail</b>	Pass

**Table 6.2.30 Test Case of History Review ( Date Filter)**

#### **6.2.31 Test Case 31: History Review Table (Payment Date Filter)**

<b>Test Case ID</b>	TCWHR02
---------------------	---------

<b>Function Requirement</b>	Sort the ticket based on payment date.
<b>Purpose</b>	To validate whether the admin can sort the tickets ascending or descending based on Payment Date.
<b>Pre-Condition</b>	Admin already filters the starting date and end date.
<b>Test Data</b>	Up and Down button of the Payment Date.
<b>Steps</b>	1) Admin clicks the down arrow button of the Payment Date. 2) Admin clicks the up arrow button of the Payment Date.
<b>Expected Result</b>	Down arrow button: The tickets will be sorted with a descending payment date. Up arrow button: The tickets will be sorted with an ascending payment date.
<b>Actual Result</b>	Down arrow button: The tickets are sorted with a descending payment date. Up arrow button: The tickets are sorted with an ascending payment date.
<b>Pass/Fail</b>	Pass

**Table 6.2.31 Test Case of History Review (Payment Date Filter)**

#### **6.2.32 Test Case 32: History Review Table (Total filter)**

<b>Test Case ID</b>	TCWHR02
<b>Function Requirement</b>	Sort the ticket based on payment date.

<b>Purpose</b>	To validate whether the admin can sort the tickets ascending or descending based on Payment Date.
<b>Pre-Condition</b>	Admin already filters the starting date and end date.
<b>Test Data</b>	Up and Down button of the Payment Date.
<b>Steps</b>	1) Admin clicks the down arrow button of the Payment Date. 2) Admin clicks the up arrow button of the Payment Date.
<b>Expected Result</b>	Down arrow button: The tickets will be sorted with a descending payment date. Up arrow button: The tickets will be sorted with an ascending payment date.
<b>Actual Result</b>	Down arrow button: The tickets are sorted with a descending payment date. Up arrow button: The tickets are sorted with an ascending payment date.
<b>Pass/Fail</b>	Pass

**Table 6.2.32 Test Case of History Review ( Total Filter)**

#### **6.2.33 Test Case 33: History Review Table (Status filter)**

<b>Test Case ID</b>	TCWHR04
<b>Function Requirement</b>	Sort the ticket based on Status.

<b>Purpose</b>	To validate whether admin can sort the tickets ascending or descending based on Status.
<b>Pre-Condition</b>	Admin already filters the starting date and end date.
<b>Test Data</b>	Up and Down button of the Status.
<b>Steps</b>	1) Admin clicks the down arrow button of the Status. 2) Admin clicks the up arrow button of the Status.
<b>Expected Result</b>	Down arrow button: The tickets will be sorted starting with “Paid” status to “Not Paid” status. Up arrow button: The tickets will be sorted starting with “Not Paid” status to “Paid” status.
<b>Actual Result</b>	Down arrow button: The tickets are sorted starting with “Paid” status to “Not Paid” status. Up arrow button: The tickets are sorted starting with “Not Paid” status to “Paid” status.
<b>Pass/Fail</b>	Pass

**Table 6.2.33 Test Case of History Review ( Status Filter)**

### **6.3 Test Result Analysis (Non-functional Requirement)**

#### **6.3.1 Performance**

- The loading screen needs to be loaded within 2 seconds.
- The REST service must respond within 1000 milliseconds.

#### **6.3.2 Availability**

The application will be available 99.9% of the time.

### **6.3.3 Scalability**

The system must be scalable enough to support 100000 visits at the same time with optimal performance.

### **6.3.4 Portability**

The application must support Android devices since Android version 5.1(Lollipop) and above.

### **6.3.5 Maintainability**

The time to restore the system for maintenance must not be more than 3 hours.

### **6.3.6 Security**

All user data in the system must be protected against malware attack.

## **6.4 Conclusion**

In conclusion, system testing is important to ensure the system that the team has developed is in excellent condition and follows the system requirements before being used by the user out there. The weaknesses of the system which were found out during the testing will be improved.

## **CHAPTER 7: CONCLUSION**

### **7.1 Introduction**

In conclusion, Smart Parking System is already developed completely and achieves the system development standards. Each of the developed modules have been explained and described thoroughly throughout this report. The team is also already completing the system testing to ensure the application and website only provide the best results and experience for the user.

### **7.2 Achievement**

Before the team starts to do the implementation progress, the team has listed a few modules which are needed to complete the system. The modules are login module, register module, payment module, booking module, OCR module, due time module, blacklist module, price update module, report module and admin module. The team has successfully developed all the long-listed modules and become an excellent system. All the scopes and objectives also have been accomplished. The system also has been tested by the team to ensure the system is working as proposed.

### **7.3 Project Limitations**

Smart Parking System is a well-developed system. However, there are also weaknesses and limitations of the system which will decrease the quality of the system in the eye of the users. The Smart Parking System is only available in three districts in Melaka. This will be disadvantageous to the people who want to use it outside of Melaka. Furthermore, the system also does not provide an extended time function for the user who wants to extend their parking time. In addition, the system only uses one language which is English, so it is inconvenient for people who do not use the language. Last but not least, the application can only be used in Android. This is also inconvenient for non-Android users.

#### **7.4 Suggestions For Improvement**

The Smart Parking System still has room for improvement for better quality and use. The improvement that can be made is adding more places. So, there are more people who can use the application everywhere. Furthermore, the system also needs to add the extended time function. It will be convenient for people who want to park their car longer. In addition, the system needs to use multi-language in order to give comfort to people who are using it.

#### **7.5 Potential Commercialisation**

The Smart Parking System that has been developed by our team surely has potential to be commercialized out there. It has all the features to turn the normal parking system into an efficient one. But, in Malaysia, the Smart Parking System environment is still not widely set up. The system is usable, however it still needs a few improvements in order to accommodate the larger scale business or companies to use our system. In order to reach the market, the team has to ensure the system is flexible and needs to thrive and prosper in a continuous and unpredictable business environment.

#### **7.6 Conclusion**

To conclude, Smart Parking System is a potential and quality system which will give comfort to the people who are using it. The system is believed to achieve the objectives and settle all the parking problems out there.

## REFERENCES

1. Hamilton, T. (2023, January 10). *Agile Methodology: What is Agile Model in Software Testing?*. Guru99. <https://www.guru99.com/agile-scrum-extreme-testing.html>
2. Gautam, S. (2019, September 20). *The Real Facts of Traffic Jam and Parking Issues in Kuala Lumpur*. GetMyParking. <https://blog.getmyparking.com/2019/09/20/the-real-facts-of-traffic-jam-and-parking-issues-in-kuala-lumpur/>
3. Hamilton, T. (2022, December 31). *How to Write Test Cases in Software Testing with Examples*. Guru99. <https://www.guru99.com/test-case.html>