

**“EASY PARK”**  
**CAR PARK MANAGEMENT SYSTEM**

By  
M.T.S. Ilangakoon  
IM/2013/022

A report submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Science Honours in Management and Information Technology

Name of the Supervisor: Dr. Ruwan Wickramarachchi

Department of Industrial Management  
University of Kelaniya  
Sri Lanka  
2017

## DECLARATION

I hereby certify that this project and the all the artifacts associated with it is my own work and it has not been submitted before nor is currently being submitted for any other degree programme.

Full name of the student: Mudiyansele Tharushi Sandunika Ilangakoon

Student No: IM/2013/022

Signature of the student: .....

Date: .....

Name of the supervisor(s): Dr. Ruwan Wickramarachchi

Signature of the supervisor: .....

Date: .....

## ACKNOWLEDGEMENT

First and foremost, I would like to extend my heartfelt gratitude to my project supervisor, Dr. Ruwan Wickramarachchi for his immense support and guidance given to me from the beginning of the project.

I would also like to convey my thanks to Dr. Shantha Jayalal, the project Coordinator, for initial understanding and guidance we were given and for his support given throughout the project.

I thank all the staff members, especially the panel of judges who gave valuable inputs and feedback on the poster presentation of the system I was proposing to do. Those comments helped immensely to identify points to be improved and to have a clear idea of the system scope.

My sincere thanks go to all my friends and batch mates who have been sharing knowledge and extending support in case of need and to each and every individual who have been by supportive by all mentioned and non – mentioned means. Last but not least, my gratitude always extended to my parents and family who is behind everything I am today for being there with me since the beginning.

## ABSTRACT

In present day, with developments in information technology, every organization tries to become smarter by integrating smart systems in their business processes. This has been amplified with the increased usage of Internet. The Internet has revolutionized communication and Internet of Things concept came into play by connecting everyday objects in a network allowing them to send and receive data.

The busy schedules and the improved standards of living have caused the increased use of private vehicles by the community. But a lot of valuable time of people is wasted in trying to find a car park to park their vehicle.

“Easy Park” Car Park Management System is a mobile application designed with a number of features to help the drivers find a car park easily and quickly.

My main objective is to facilitate solutions for the unarticulated needs of a driver in Sri Lanka. Once the mobile application is developed, the driver can use it from via any mobile device which provide Internet facility. This enables a driver to find the nearest car park to his destination. It provides the reservation facility of a parking space of a selected car park, payment handling facility, GPS facility to get directions to the car park, SMS facility to get updates on parking reservations and report generation facilities.

The report expects to discuss the issues faced by a particular driver, the findings of the system analysis done by the author and functional and technical features of the proposed “Easy Park” Car Park Management System.

# TABLE OF CONTENTS

DECLARATION .....	i
ACKNOWLEDGEMENT .....	ii
ABSTRACT .....	iii
TABLE OF CONTENTS .....	iv
LIST OF TABLES .....	viii
LIST OF FIGURES .....	ix
1. INTRODUCTION.....	1
1.1 Introduction to “Easy Park” Car Park Management System .....	2
1.2 Business Process .....	3
1.3 Problem Definition.....	3
1.4 Aims and Objectives .....	4
1.5 Scope with Clear Boundaries .....	5
1.6 Organization of the Report.....	6
1.7 Summary .....	6
2. SYSTEM ANALYSIS .....	7
2.1 Use Case Diagram for the Current System .....	8
2.2 Activity Diagrams for the Current System .....	9
2.2.1 Activity diagram to describe the functionality of parking a car .....	9
2.2.2 Activity diagram to describe the functionality of the payment handling .....	10
2.3 System Requirement Analysis .....	11
2.3.1 Functional Requirements .....	11
2.3.2 Non-Functional Requirements.....	12
2.4 Requirement Specification.....	13
2.5 Business System Options (BSOs).....	14
2.5.1 Business System Options Identification .....	15
2.5.2 BSO 1: Mobile application for the driver .....	15
2.5.3 BSO 2: Web based solution for the driver.....	16
2.6 Evaluation of BSOs.....	17
2.6.1 Functional requirements vs BSOs .....	17

2.6.2	Non-functional requirements vs BSOs .....	18
2.6.3	Selected BSO .....	19
2.7	Summary .....	19
3.	SYSTEM DESIGN .....	20
3.1	Use Case Diagrams for the Proposed System .....	21
3.1.1	Overall use case diagram for proposed system .....	21
3.1.2	Search car park use case .....	23
3.1.3	Reserving parking space use case .....	24
3.1.4	Generate reports use case .....	25
3.2	Use Case Descriptions for the Proposed System .....	26
3.2.1	User login use case description .....	26
3.2.2	Creating user account use case description .....	27
3.2.3	Search car park use case description .....	28
3.2.4	Reserve parking space use case description .....	29
3.2.5	Handle payment use case description .....	30
3.2.6	Generate reports use case description .....	31
3.2.7	Update user information use case description .....	32
3.2.8	Update car park status use case description .....	33
3.2.9	Create sensor account use case description .....	34
3.2.10	Updating reservation use case description .....	35
3.2.11	Sending SMS notification use case description .....	36
3.3	Activity Diagram for the Proposed System .....	37
3.3.1	Activity diagram for user login .....	37
3.3.2	Activity diagram for creating user accounts .....	38
3.3.3	Activity diagram for searching a car park .....	39
3.3.4	Activity diagram for reserving a parking space .....	40
3.3.5	Activity diagram for payment handling .....	41
3.3.6	Activity diagram for updating user information .....	42
3.3.7	Activity diagram for generating reports .....	43
3.3.8	Activity diagram for updating car park status .....	44
3.3.9	Activity diagram for updating reservation .....	45

3.3.10	Activity diagram for sending SMS notification .....	46
3.4	Sequence Diagrams for the Proposed System .....	47
3.4.1	Sequence diagram for user login .....	47
3.4.2	Sequence diagram for creating user account .....	49
3.4.3	Sequence diagram for searching a car park .....	50
3.4.4	Sequence diagram for reserving a parking space .....	51
3.4.5	Sequence diagram for payment handling .....	52
3.4.6	Sequence diagram for generating reports .....	53
3.5	Class Diagram for Proposed System.....	54
3.6	Database Design.....	55
3.6.1	Entity - Relationship model .....	55
3.6.2	Normalized database design .....	57
3.7	GUI Design .....	58
3.7.1	Welcome Page GUI .....	58
3.7.2	User login GUI .....	59
3.7.3	Create user GUI .....	59
3.7.4	Search car park GUI .....	60
3.7.5	View car park GUI .....	60
3.7.6	Report .....	61
3.7.7	Screen of the car park .....	62
3.8	Summary .....	62
4.	DEVELOPMENT .....	63
4.1	Programming Languages and Development Tools.....	64
4.1.1	Programming Language .....	64
4.1.2	Development Tools and Technologies .....	66
4.2	Third Party Components and Libraries.....	68
4.3	Algorithms used in the System .....	69
4.4	Hashing of the Confidential Data .....	70
4.5	Summary .....	71
5.	TESTING .....	72
5.1	Test Plan and Test Strategy.....	73

5.1.1	White Box Testing.....	73
5.1.2	Black Box Testing .....	73
5.2	Testing Environment.....	74
5.3	Sample Test Cases .....	74
5.4	Test Report.....	79
5.5	Summary .....	80
6.	IMPLEMENTATION .....	81
6.1	System Requirements.....	82
6.1.1	Hardware Requirements .....	82
6.1.2	Software Requirements.....	82
6.2	User Manual.....	82
6.2.1	Home Page of the System.....	83
6.2.2	Plan your Parking Interface .....	83
6.2.3	Signin Interface.....	84
6.2.4	Home Page of the user profile .....	85
6.2.5	User Signup .....	87
6.2.6	Reserving a Parking Space by the Driver .....	88
6.2.7	Reserving a Parking Space by the Car Park .....	92
6.2.8	Parking Process in the Car Park .....	93
6.2.10	Reports.....	97
6.3	Summary .....	99
7.	EVALUATION AND CONCLUSION.....	100
7.1	Degree of objectives met .....	101
7.2	Usability, Accessibility, Reliability and Friendliness.....	102
7.3	Limitations and Drawbacks .....	103
7.4	Future Modifications, Improvements and Extensions .....	103
7.5	Summary .....	103
	REFERENCES .....	104



## LIST OF TABLES

Table 2.1 Functional Requirements.....	11
Table 2.2 Non-Functional Requirements.....	12
Table 2.3 Functional requirements vs BSOs .....	17
Table 2.4 Non-Functional requirements vs BSOs .....	18
Table 3.1 User login use case description .....	26
Table 3.2 Create account use case description .....	27
Table 3.3 Search car park use case description .....	28
Table 3.4 Reserve parking space use case description .....	29
Table 3.5 Handle payments use case description .....	30
Table 3.6 Generate reports use case description.....	31
Table 3.7 Update user information use case description .....	32
Table 3.8 Update car park status use case description .....	33
Table 3.9 Create sensor account use case description .....	34
Table 3.10 Updating reservation use case description .....	35
Table 3.11 Sending SMS notification use case description.....	36
Table 5.1 User Login Test Case .....	74
Table 5.2 Add User Test Case .....	75
Table 5.3 Add Car Park Test Case .....	76
Table 5.4 Parking Space Reservation by Driver Test Case .....	76
Table 5.5 Parking Space Reservation by the Car Park Test Cases .....	77
Table 5.6 Parking Process – Checkin Test Case .....	78
Table 5.7 Parking Process – Checkout Test Case .....	78
Table 5.8 Generate Reports Test Case.....	79

## LIST OF FIGURES

Figure 1.1 Scope of the Project .....	5
Figure 2.1 Overall Use Case diagram for the Existing System .....	8
Figure 2.2 Activity Diagram for the Car Parking Process of the Existing System .....	9
Figure 2.3 Activity Diagram for Payment Handling Process of the Existing System.....	10
Figure 3.1 Use case diagram for system login and system functionalities interaction.....	21
Figure 3.2 Use case diagram for overall system.....	22
Figure 3.3 Use case diagram for searching a car park.....	23
Figure 3.4 Use case diagram for reserving a parking space .....	24
Figure 3.5 Use case diagram for generating reports .....	25
Figure 3.6 Activity diagram for user login .....	37
Figure 3.7 Activity diagram for creating user account.....	38
Figure 3.8 Activity diagram for searching a car park.....	39
Figure 3.9 Activity diagram for reserving a parking space .....	40
Figure 3.10 Activity diagram for handle payment .....	41
Figure 3.11 Activity diagram for updating user information .....	42
Figure 3.12 Activity diagram for generating reports .....	43
Figure 3.13 Activity diagram for updating car park status.....	44
Figure 3.14 Activity diagram for updating reservation .....	45
Figure 3.15 Activity diagram for sending SMS notifications .....	46
Figure 3.16 Sequence diagram for user login.....	47
Figure 3.17 Sequence diagram for creating user account.....	49
Figure 3.18 Sequence diagram for searching a car park.....	50
Figure 3.19 Sequence diagram for reserving a parking space .....	51
Figure 3.20 Sequence diagram for payment handling .....	52
Figure 3.21 Sequence diagram for generating reports.....	53
Figure 3.22 Class diagram for the proposed system.....	54
Figure 3.23 Entity – Relationship model.....	55
Figure 3.24 Normalized database design.....	57
Figure 3.25 Welcome Page GUI.....	58
Figure 3.26 User login GUI.....	59

Figure 3.27 Create user GUI.....	59
Figure 3.28 Search Car Park GUI.....	60
Figure 3.29 View Car Park GUI.....	60
Figure 3.30 Report.....	61
Figure 3.31 Screen of the car park.....	62
Figure 6.1 Home Page .....	83
Figure 6.2 Plan Your Parking Interface.....	83
Figure 6.3 Signin Interface .....	84
Figure 6.4 Admin User Profile Interface .....	85
Figure 6.5 Driver User Profile Interface.....	85
Figure 6.6 Car park Profile Interface.....	86
Figure 6.7 Select User Type .....	87
Figure 6.8 Driver Registration Form .....	87
Figure 6.9 Car Park registration Form.....	88
Figure 6.10 Viewing the google map .....	89
Figure 6.11 Map with Car Parks.....	89
Figure 6.12 Info Window .....	90
Figure 6.13 Parking Reservation Form viewed by the driver.....	90
Figure 6.14 Reservation Details .....	91
Figure 6.15 Adding a reservation by the car park .....	92
Figure 6.16 Parking Reservation form viewed by the car park .....	93
Figure 6.17 Insert Checkin Details .....	94
Figure 6.18 Checkin Details .....	94
Figure 6.19 Insert Checkout details.....	95
Figure 6.20 Checkout Details .....	96
Figure 6.21 Payment Details .....	96
Figure 6.22 Generating Reports.....	97
Figure 6.23 Reports .....	97
Figure 6.24 Frequency Reports .....	98
Figure 6.25 Overall Customer Profitability .....	98
Figure 6.26 Customer Profitability by Car Park Report.....	98

Figure 6.27 Profitability of the Car Parks.....	99
Figure 6.28 Customer Information Report .....	99

# CHAPTER 1

## 1. INTRODUCTION

This chapter gives a brief introduction of the Web based Car Park Management System and related web solution, need of the system and the user base. It also describes the business process, aims and objectives of the proposed system and the scope of the project. At the end, this chapter explains the feasibility and tools.

## **1.1 Introduction to “Easy Park” Car Park Management System**

The lives of people are changing due to developments in Information Technology. With the introduction of the Internet, people tend to embrace Information Technology much further to ease their day to day activities.

The Internet of Things extends Internet connectivity beyond the traditional devices like desktop and laptop computers, smartphones and tablets to a diverse range of devices and everyday things that utilize embedded technology to communicate and interact with the external environment, all via the Internet.

These developments have changed the daily lifestyles of people rapidly. Today the systems and processes of the organizations have been automated using these improvements in IT and Internet to increase their productivity and efficiency.

According to the current scenario, vehicles have become a major concern when it comes to parking. More fuel is consumed by vehicles in trying to find a car park in a busy city. Even when they find a car park, they have to drive around the car park to find a parking space. In some parks, there's a parking attendant to direct them, but in most of the car parks there's no one to direct. Sometimes drivers tend to park the car on the side of the road causing major traffic jams when they are unable to find a car park or a parking space. It would be better if a platform is developed to let the drivers know about the car parks in a particular city and the free parking spaces available in the car park.

This software is basically a system that facilitates car parks to serve customer at their best. Parking availability changes very quickly and the drivers don't want to show up at a space that is no longer available. So the main priority is to provide accurate and reliable real time data to the drivers all the time. Through this system a person can reserve a parking space in a selected car park that is nearer to his destination. This system enables the car park attendant to calculate parking charges easily with real time data obtained from the sensors. The special feature of this system is the navigation facility which will direct the user to the particular car park and the parking space without an issue.

## **1.2 Business Process**

At present in Sri Lanka, the parking process in a car park is carried out in a completely manual manner.

A vehicle has to drive through the whole car park to find a free parking space when it comes to a car park. If there are no parking spaces, the vehicle have to leave the car park to find another car park. If the driver is a newcomer to the city, he has to drive around the whole city to find a car park.

If there is a parking space, a ticket will be written by the parking attendant and given to the driver where the time the car was parked is recorded. When the car leaves the car park, the due amount is needed to be paid by the driver. The due amount is calculated considering the time the car was parked in the car park.

In most of the car parks, a “parking is full” sign is kept at the entrance by the parking attendant. But since it is visible only when the car goes to the car park, drivers face issues and sometimes traffic jams also occur. This may cause loss of customers to the car parks as well as to the organization or the shop to which the customer has target to go.

## **1.3 Problem Definition**

Following are some major problems identified in the current car parks.

- Lot of fuel is consumed during the search for the parking spaces through the car park
- Driver doesn't know about the parking availability of a particular car park until he gets to the car park
- The parking ticket can be lost
- Inability to issue a ticket with the entrance time if the parking attendant not present
- Time consuming
- Driver doesn't know the directions to find a particular car park in the city and a parking space in a car park

At present, a typical car park in Sri Lanka carries out its operations in a completely manual manner which is very inconvenient for the drivers and the parking attendants.

## 1.4 Aims and Objectives

The main objective of this “Easy Park” car park management system is to go beyond just helping the drivers to locate a parking space. The system enables the drivers to locate the right parking space.

1. Minimize the time consumed with finding a parking space

Parking space reservation functionality is provided by the proposed car park management system. Parking space of a particular car park in the selected city can be reserved by the user while staying at home. This will reduce the fuel consumption and the time wasted in finding a parking space. The estimated parking time can be given by the user, which could be used for management purposes of the car park. The parking spaces will be allocated according to the time the vehicle will be parked in the car park.

The GPS facility will be provided by this future system to direct the driver to the relevant car park where the parking space is reserved.

2. Improve efficiency and accuracy

Electronic payment methods will be enabled through the system. The parking ticket will be sent automatically to the particular user account. The entrance time and exit time will be extracted from the system through RFID technology and used for the calculation of the parking charges. The payment amount will be automatically deducted from the subscription.

3. Provide timely data

Timely and efficient data will be transmitted through the sensors and other physical devices in the car park. Based on the transmitted data SMS notifications will be sent to the user to communicate about the updates in reservation facilities of a selected car park or a selected city.

4. Improved decision making

Assistance for the management decision making process will be provided by the new system with an array of statistical and operational research analysis tools to generate reports.



## 1.5 Scope with Clear Boundaries

This section will discuss the scope of the project as being depicted by the figure 1.1 below.

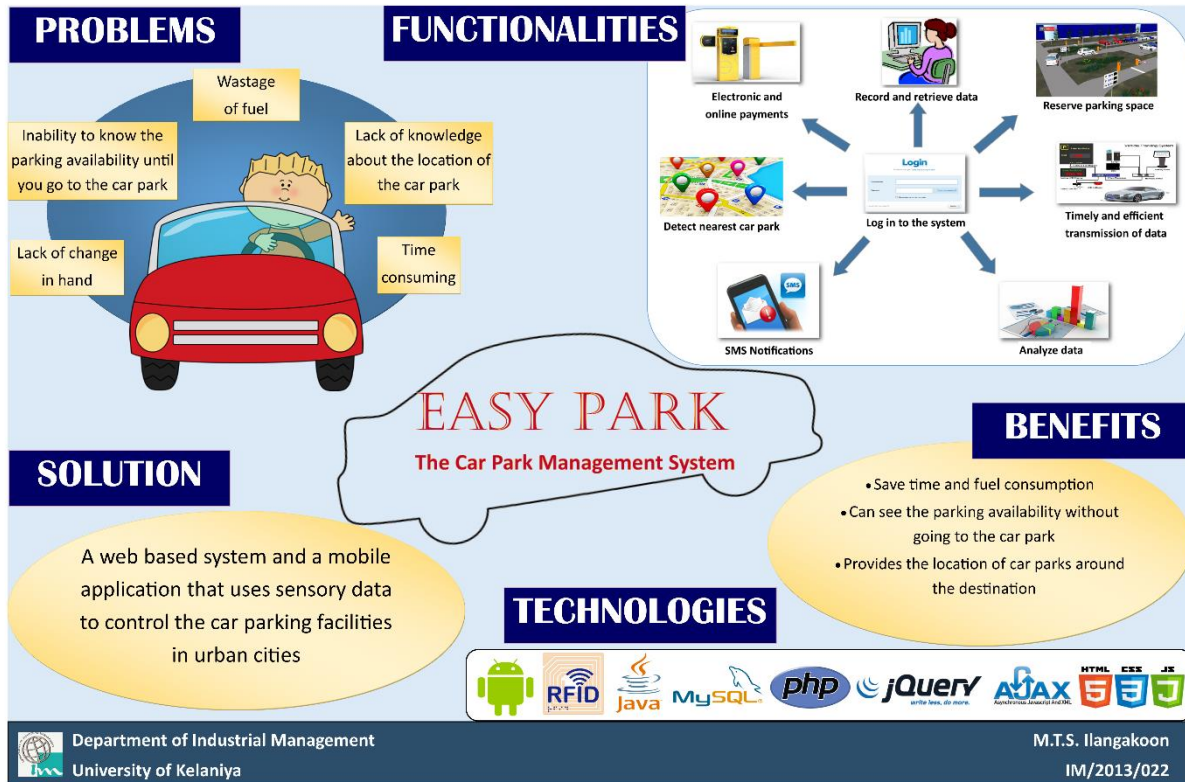


Figure 1.1 Scope of the Project

As illustrated in figure 1.1 “Easy Park” car park management system provides a web based system and a mobile application that uses sensory data to control the car parking facilities in urban cities. The project will cover registration of users to the system by creating user accounts for the drivers, car parks and the system administrator. The location of the car park is stored in the system and it is shown in the map when the driver searched for car parks in a particular destination. Parking slots can be reserved by the driver by giving the date, entrance time and exit time. SMS notifications will be sent to the driver reminding him about the reservation. RFID technology will be used to track the entrance and exit time of a vehicle to calculate the parking charges. The payment amount will be automatically deducted from the driver’s subscription. Timely and accurate data will be transferred by the sensors in the parking slot stating whether a particular slot is available for parking.

## **1.6 Organization of the Report**

This thesis covers the system analysis and design segments of the proposed system for a car parking process.

Chapter 1 introduces the current business process, problems and the objectives of the proposed system. It clearly defines the scope and the boundaries.

Chapter 2 presents the clear analysis of the existing system and the requirement analysis of the system. It further analyses the available business system options (BSOs) and defines the best option to proceed with.

Chapter 3 is the depiction of the furtherance of the project after the requirement analysis and specification. This will provide a better understanding of the system behavior and the interactions with the diagrams used. One diagram will lead to another to explain the functionality, entries and their relationships. By the end of the chapter, database design will show the tables which will be used in the system and Graphical User Interfaces will further elaborate its performance.

## **1.7 Summary**

In Chapter 1, the nature of the business process and the current functionalities were identified. Then the problems and issues were defined, through which the objectives were emphasized. Furthermore, clear boundaries and scope for the system were defined.

# CHAPTER 2

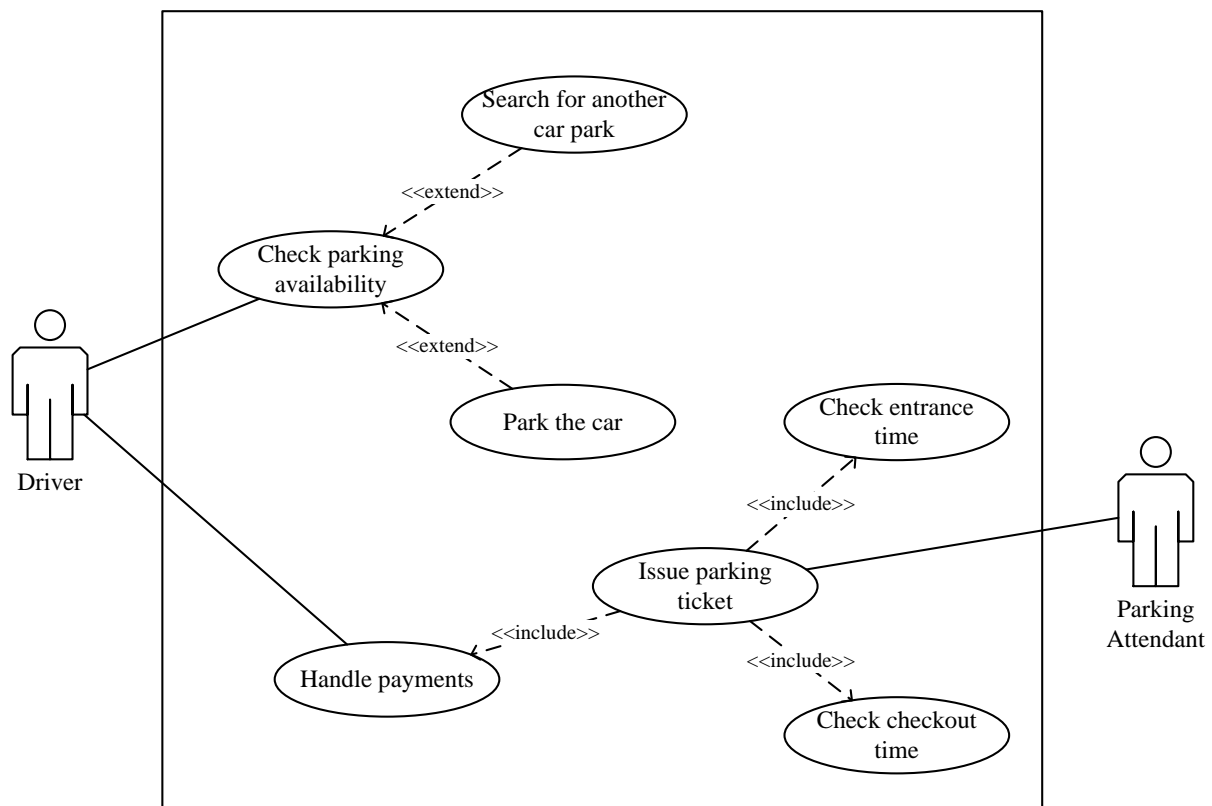
## 2. SYSTEM ANALYSIS

This chapter mainly covers the requirement analysis of the system. It describes the requirement gathering techniques, how the data are analyzed and the requirement definitions. Then going into much deeper, it shows what are the business system options of the project and which BSO is the option to design the system.

## 2.1 Use Case Diagram for the Current System

The functionalities of existing process are completely handled manually, so we cannot clearly identify an exact boundary for the manual process because it's open for all users who involve in this process under different levels. Therefore, the consistency of the process is very poor. Below use case diagram shows the overall scope of the existing process which completely carries out manually.

The figure 2.1 illustrates the overall use case diagram for the existing car parking system



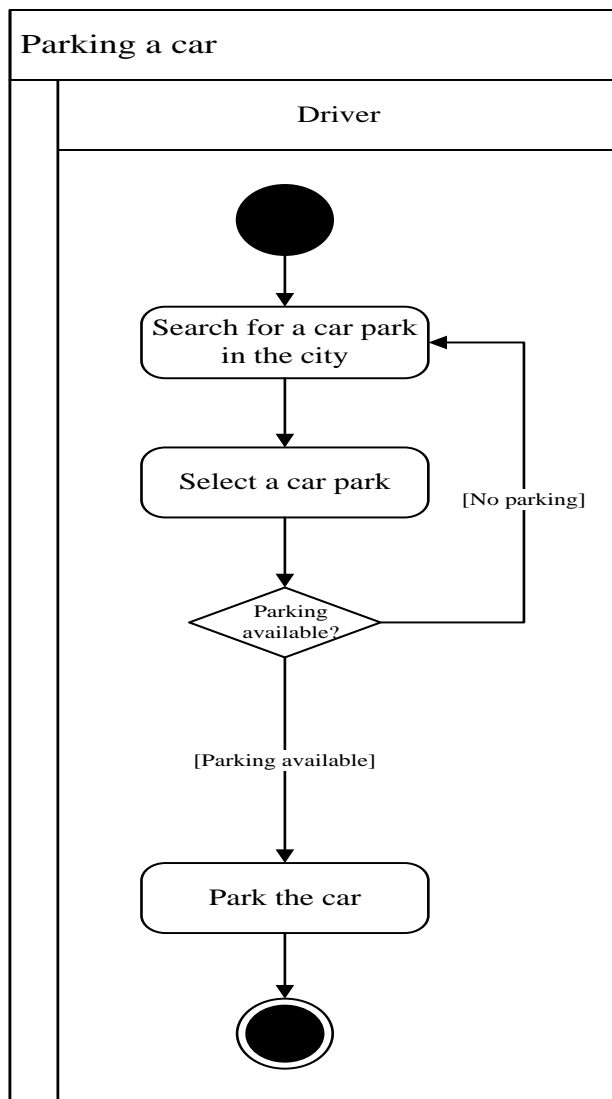
**Figure 2.1 Overall Use Case diagram for the Existing System**

The figure 2.1 illustrates how the users interact with the system functionalities of the current system. Check parking availability, handle payments, issuing parking tickets, searching for another car park if parking not available, checking entrance and exit times are the main use cases identified in the current system.

## 2.2 Activity Diagrams for the Current System

### 2.2.1 Activity diagram to describe the functionality of parking a car

The figure 2.2 illustrates the activity diagram to describe the functionality of the car parking process identified in the existing use case in the manual process.

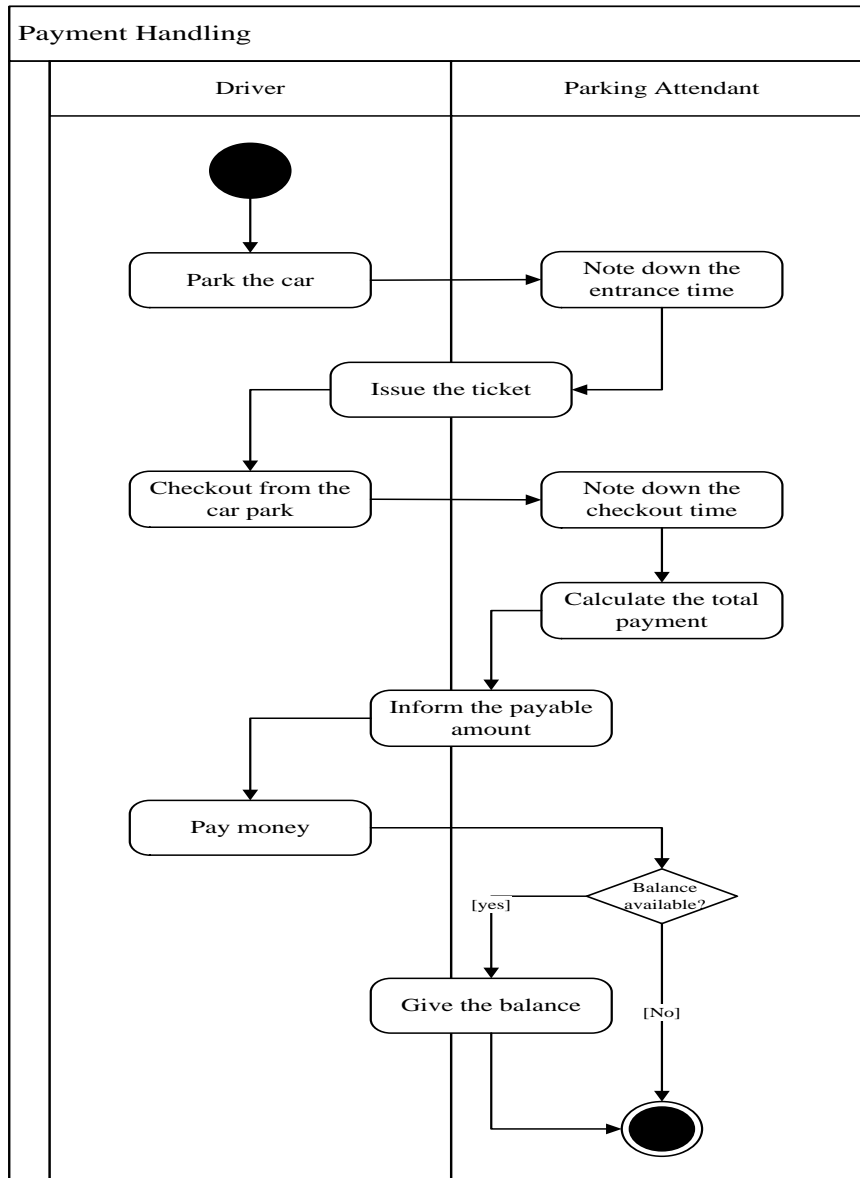


**Figure 2.2 Activity Diagram for the Car Parking Process of the Existing System**

The activity diagram for car parking process of the existing system illustrated in figure 2.2 describes the flow of activities by the relevant users of the system in an elaborated manner. The activities done by a particular user and how a decision made by a user proceeds to the next action is clearly depicted in the diagram.

### 2.2.2 Activity diagram to describe the functionality of the payment handling

The figure 2.3 illustrates the activity diagram to describe the functionality of the payment handling identified in the current manual process.



**Figure 2.3 Activity Diagram for Payment Handling Process of the Existing System**

The activity diagram for user payment handling process of the existing system illustrated in figure 2.3 describes the flow of activities by the relevant users of the system in an elaborated manner. The activities done by a particular user and how a decision made by a user proceeds to the next action is clearly depicted in the diagram.

## 2.3 System Requirement Analysis

System requirements analysis phase aims at providing a full description of the problem based on the concepts defined in the problem domain. This section would brief the reader of the functional and non-functional requirements identified by the author in the system analysis phase and those will be identified as mandatory requirements with the ‘shall and non-mandatory requirements with the wording ‘should’.

### 2.3.1 Functional Requirements

The table 2.1 will show the functional requirements of the system.

**Table 2.1 Functional Requirements**

ID	Requirement
1.	Shall be able to create a user account
2.	Shall be able to reserve a parking space by stating the necessary time and date
3.	Shall be able to delete, update and search reservation details
4.	Shall be able to insert, delete, update and search vehicle owner details in case of an emergency
5.	Shall be able to insert, delete, update and search car park details.
6.	Shall be able to view all the car parks in a particular city
7.	Shall be able to identify free parking spaces in a particular car park
8.	Shall be able to navigate the driver to the car park
9.	Shall be able to send SMS notifications for the user about the reserved parking space
10.	Shall be able to calculate the parking fee by considering the parking time and check out time of a vehicle
11.	Shall be able to handle payments
12.	Shall be able to generate summary reports
13.	Should be able to send the invoice to the user
14.	Should be able to calculate the probability of freeing a car space through statistical analysis

The functional requirements described in table 2.1 are what the system should be able to do in order to satisfy the main objective of it. Mainly these describe what are the tasks or functions which the system can perform after implementation.

### 2.3.2 Non-Functional Requirements

Non-functional requirements describe the behavior of the system other than the main functionalities of it. Hence, they will cover the requirements which are not included in section 2.3.1. The table 2.2 will show the non-functional requirements identified.

**Table 2.2 Non-Functional Requirements**

ID	Requirements
1.	Shall be able to present a more attractive user-friendly interface
2.	Shall be able to be up and running 24 hours
3.	Shall be able to provide fast service
4.	Shall be able to provide accurate and reliable real time data to drivers
5.	Should be platform independent
6.	Should be able to transmit data through sensors accurately and efficiently
7.	Should be able to use the system services through a smart phone
8.	Should be facilitate to access without time and geography restrictions
9.	Should be able to provide safe and secured system

The non-functional requirements in table 2.2 describe the usability, reliability, performance, maintainability and other similar aspects of the system. These set of requirements may not be directly related to the main functionality, but they are of extreme importance to the proper functioning of the system.



## 2.4 Requirement Specification

This section explains the basic functionalities required in the customer side mobile solution of the proposed system. All requirements will be organized in order to give a full view of the system and describe them in an orderly fashion.

### Create customer profile

1. The system shall allow user to create profile with his/her details and set credentials
2. Only one registration is allowed with one valid NIC number of a customer
3. At the registration, some information should be compulsory to retrieve and that information will be accessible to some authorized users
4. The system shall authenticate user credentials to view the profile
5. The system shall allow the user to update the profile information

### Create car park profile

1. The system shall allow the car park attendant to create a profile with the relevant details
2. Any number of car parks can be registered with the same name but with different locations
3. At the registration, some information should be compulsory to retrieve and that information will be accessible to some authorized users
4. The system shall authenticate user credentials to view the profile
5. The system shall allow the user to update the profile information

### Users shall be able to log into the system

1. Users shall provide the username and password when log into the system
2. Users should be able to update information in their accounts whenever they intended
3. Users should be able to logout of the system whenever they intended
4. Users should be able to deactivate their account whenever they intended

Provide search and browse cities, according to the user requirements

1. The system shall provide the search facility
2. The system shall enable the user to enter search text on the screen
3. The system shall display the location and a detailed list of car parks in a particular city
4. The system shall enable users to navigate between search results
5. The system shall notify the user when no matching result found on the search

Provide reservation facility

1. The user shall be able to input reservation time and date
2. The system shall be able to transmit necessary reservation details between the customer and the car park
3. The system shall be able to calculate parking fee according to the defined conditions
4. The system shall be able to send notifications to the user about the reservation facility

Provide comprehensive parking details

1. The system shall provide detailed information about a particular car park
2. The system shall provide the availability of the parking spaces in the car park

## **2.5 Business System Options (BSOs)**

This project has to be focused on satisfying the identified requirements and also focused on the future requirements. According to this project, the features which the BSOs are offering should satisfy mainly the driver's online parking requests, parking reservation requirements and car park attendant's data entry & transmission requirements. In line with these requirements, all the BSOs must have the car park infrastructure and internet facility because any interested user should be able to access to browse and get parking details.

In this section BSOs will be presented and at the end they will be evaluated to come up with the best option. Each BSO will be consisted of overlapping features as well as exclusive features. Since each BSO will be focused on different aspects, each BSO will require different sets of software combinations. Evaluation of the BSOs will be done by comparing them against the functional and non-functional requirements of the system.

### **2.5.1 Business System Options Identification**

BSO 1 – Mobile application for the driver

BSO 2 – Web based solution for the driver

### **2.5.2 BSO 1: Mobile application for the driver**

#### **Description**

This BSO is concerned about introducing a mobile application for the driver. This will be an integration of android application and website with added functionalities. Accessibility is available from anywhere, anytime along with an internet accessibility.

#### **Functionalities**

Driver can create their profiles through the mobile application, secured login, search car parks that are closer to their destination, check parking availability, reserve parking slots in a car park by giving the time and date, view payment details and hold review will integrate with this BSO. The driver can access to the web interface as well as the mobile application using their mobile phones when the internet access is enabled.

The navigation function is provided by the google map API integrated with the mobile application.

#### **Benefits**

- Can be accessed from anywhere through the internet
- Accurate and easy navigation
- API can be integrated easily with least modifications

#### **Issues**

- Incompatibility
- Less shear ability

## **Justification**

This BSO satisfies almost all of the functional requirements and non-functional requirements. Since this is an internet solution, it will give flexibility to both the driver and the car park attendant to use it, access it from anywhere and update information. The google map API used in the mobile application enable the driver to detect the car park easily and provide navigational instructions to find the car park. But the problem is the incompatibility of the application due to different operating systems used in different smart phones.

### **2.5.3 BSO 2: Web based solution for the driver**

#### **Description**

This BSO is concerned about introducing a web-based online application for the drivers.

#### **Functionalities**

User can create their profiles, secured login, search car parks that are closer to their destination, check parking availability, reserve parking slots in a car park by giving the time and date, view payment details and hold review will integrate with this BSO. User also can access to the web interface using their mobile phones when the internet access is enabled.

#### **Benefits**

- Can be accessed from anywhere through the internet
- Give more attractive user interfaces compared to the BSO 1
- Platform independent

#### **Issues**

- Security issues
- Difficult to integrate API to the system
- Rely highly on good internet connection
- Need to use the navigation function separately to get the directions

## Justification

Though the interfaces are more attractive than the mobile application, the problem with this BSO is the inability to get the navigation instructions via the website directly. If the API is added, a considerable amount of modifications needs to be done. Only the location of the car park can be obtained.

## 2.6 Evaluation of BSOs

Evaluation of the BSOs will be done by comparing them first with the functional and non-functional requirements. Then the pros and cons of the BSOs will also be considered. Ultimately the best option could be one of the BSOs. Looking at the two BSOs, it seems that both of them cover almost all the requirements mentioned in the requirements catalogue. But there are some differences when taking the other constraints into consideration. The following tables shows the comparison of the each BSO with the requirements.

### 2.6.1 Functional requirements vs BSOs

**Table 2.3 Functional requirements vs BSOs**

ID	Requirement	BSO 1	BSO 2
1.	Shall be able to create a user account	X	X
2.	Shall be able to reserve a parking space by stating the necessary time and date	X	X
3.	Shall be able to delete, update and search reservation details	X	X
4.	Shall be able to insert, delete, update and search vehicle owner details in case of an emergency	X	X
5.	Shall be able to insert, delete, update and search car park details.	X	X
6.	Shall be able to view all the car parks in a particular city	X	X
7.	Shall be able to identify the free parking spaces in a particular city	X	X
8.	Shall be able to navigate the driver to the car park	X	X
9.	Shall be able to send SMS notifications for the user about the reserved parking space	X	X

10.	Shall be able to calculate the parking fee by considering the parking time and check out time of a vehicle	X	X
11.	Shall be able to handle payments	X	X
12.	Shall be able to generate summary reports	X	X
13.	Should be able to send the invoice to the user	X	X
14.	Should be able to calculate the probability of freeing a car space through statistical analysis	X	X

Table 2.3 illustrates the comparison of the two BSO with the functional requirements. According the above comparison, all the functional requirements are satisfied by the both BSOs.

### 2.6.2 Non-functional requirements vs BSOs

**Table 2.4 Non-Functional requirements vs BSOs**

ID	Requirements	BSO 1	BSO 2
1.	Shall be able to present a more attractive user-friendly interface	X	
2.	Shall be able to be up and running 24 hours	X	X
3.	Shall be able to provide fast service	X	
4.	Shall be able to provide accurate and reliable real time data to drivers	X	X
5.	Should be platform independent		X
6.	Should be able to transmit data through sensors accurately and efficiently	X	X
7.	Should be able to use the system services through a smart phone	X	X
8.	Should be facilitate to access without time and geography restrictions	X	X
9.	Should be able to provide safe and secured system	X	X

Table 2.4 illustrates the comparison of the two BSO with the non-functional requirements. According the above comparison, most of the non-functional requirements are satisfied by the both BSOs. But attractive user friendly interface and providing fast services are not satisfied by BSO 2 while the platform independency is not satisfied by BSO 1.

### 2.6.3 Selected BSO

The selected BSO is the first one, i.e. **Mobile application for the driver**. The main objective is to assist the driver to choose the most appropriate car park according to his destination and navigate him to that particular car park. All BSOs requires the automation of the current car parking process. But when compared with BSO 2, BSO 1 is the most effective because it has the google map API which provide the navigation instructions to the driver enabling the driver to find the car park easily. In BSO 2 it only provides the location of the car park. To get the navigation instructions the user need to use a separate application. Thus BSO 1 is selected.

## 2.7 Summary

This chapter analyzed the existing features of the process and thus specified the functional and non-functional requirements. Based on the BSOs were determined and the best option was selected which was to develop a mobile application for the driver.

## CHAPTER 3

### 3. SYSTEM DESIGN

This chapter mainly covers the System Design of the system. It is about the continuation of the project after the requirement analysis and specification. It will offer a better understanding of the system behaviors and interactions with the diagrams used. One diagram will lead to another to explain the functionality, entities and their relationships. By the end of the chapter, database design will show the tables which will be used in the system.



### 3.1 Use Case Diagrams for the Proposed System

Use case provides a structured view of the system functionality. They are used to gather the requirements of a system, including internal and external influence. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. This is very important since the next phase of the SDLC is the system designing. Use case diagrams are consisted of several use cases and the actors of the system. Here the use cases are the different tasks the users will do in order to interact with the system. Actors are the users who interact with the system.

#### 3.1.1 Overall use case diagram for proposed system

The figure 3.1 shows how the users will log in to the system and how the system functionalities interact with the system login.

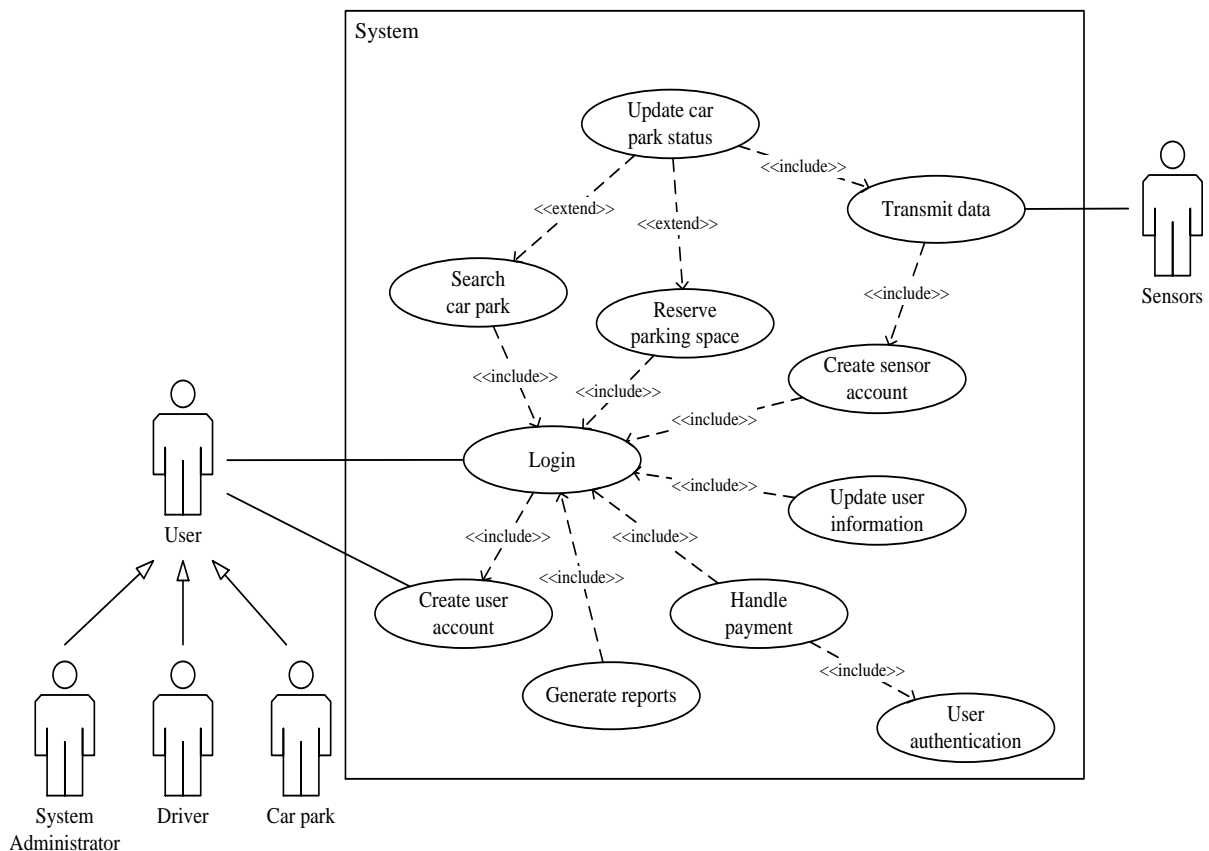
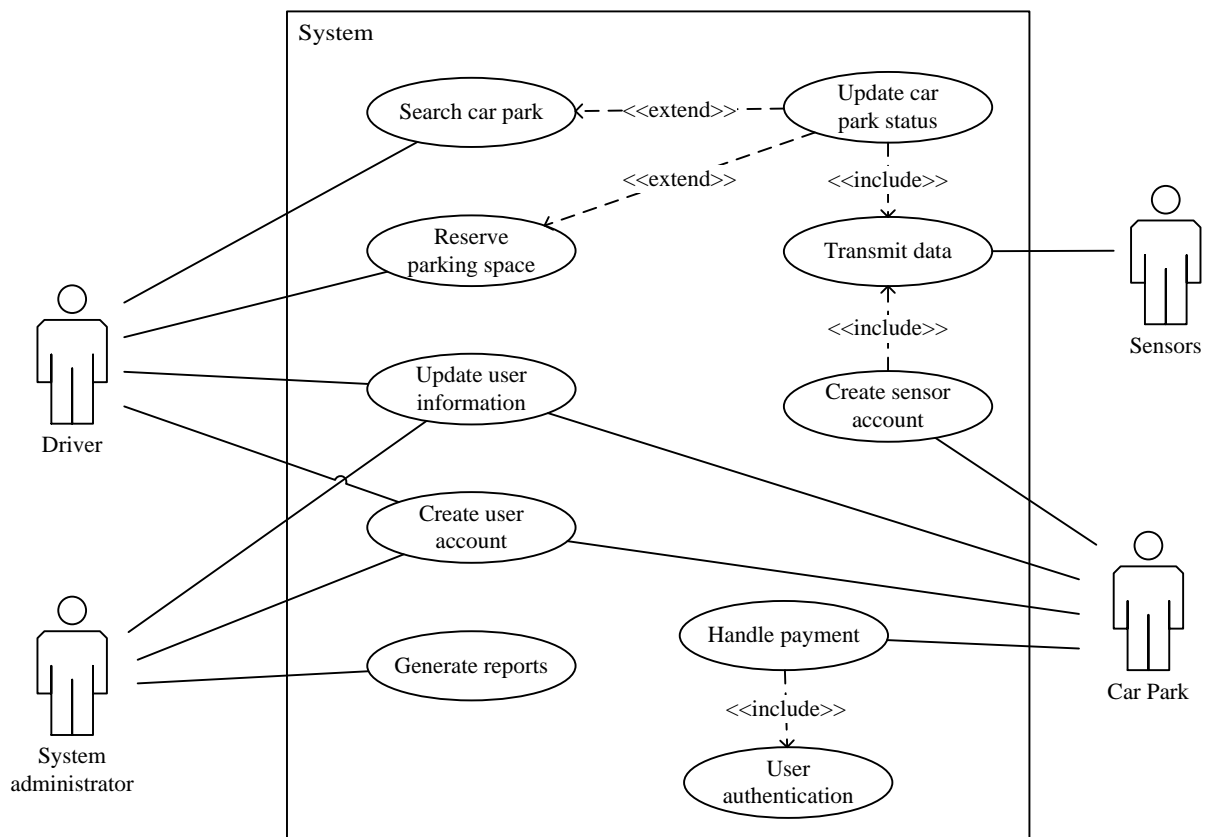


Figure 3.1 Use case diagram for system login and system functionalities interaction

The figure 3.2 shows how the users are interacting with the system functionalities.

To perform any of the tasks the user should first login (UC 01) to the system.

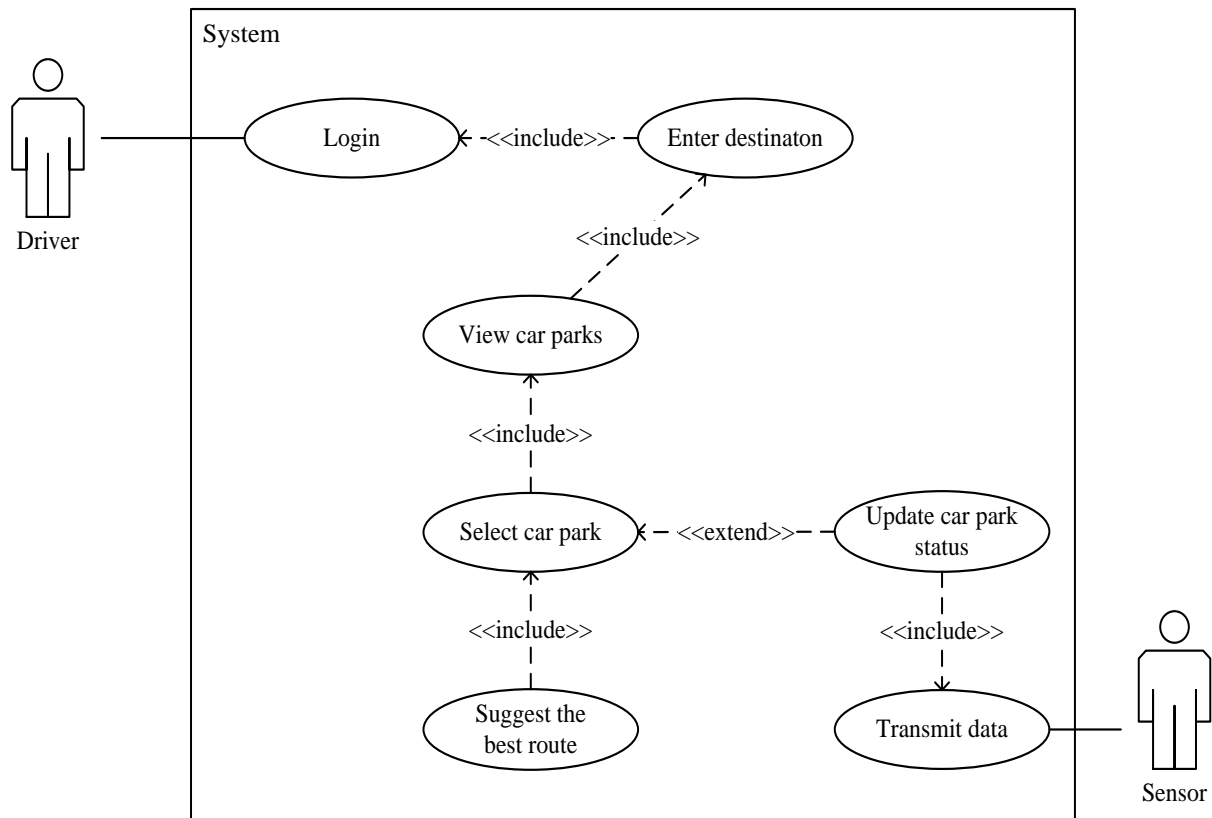


**Figure 3.2 Use case diagram for overall system**

The figure 3.1 and figure 3.2 illustrate how the users interact with the system functionalities after they logged into the system. Searching a car park and reserving a parking space are the main use cases handled by the driver in the system. Handling payments and creating sensor accounts are handled by the car park. Report generation is a functionality handled by the system administrator. Creating account and updating user information are common use cases for system administrator, driver and the car park.

### 3.1.2 Search car park use case

The figure 3.3 illustrates the granular use case diagram for “Search car park” use case in the overall use case that involved by the driver.

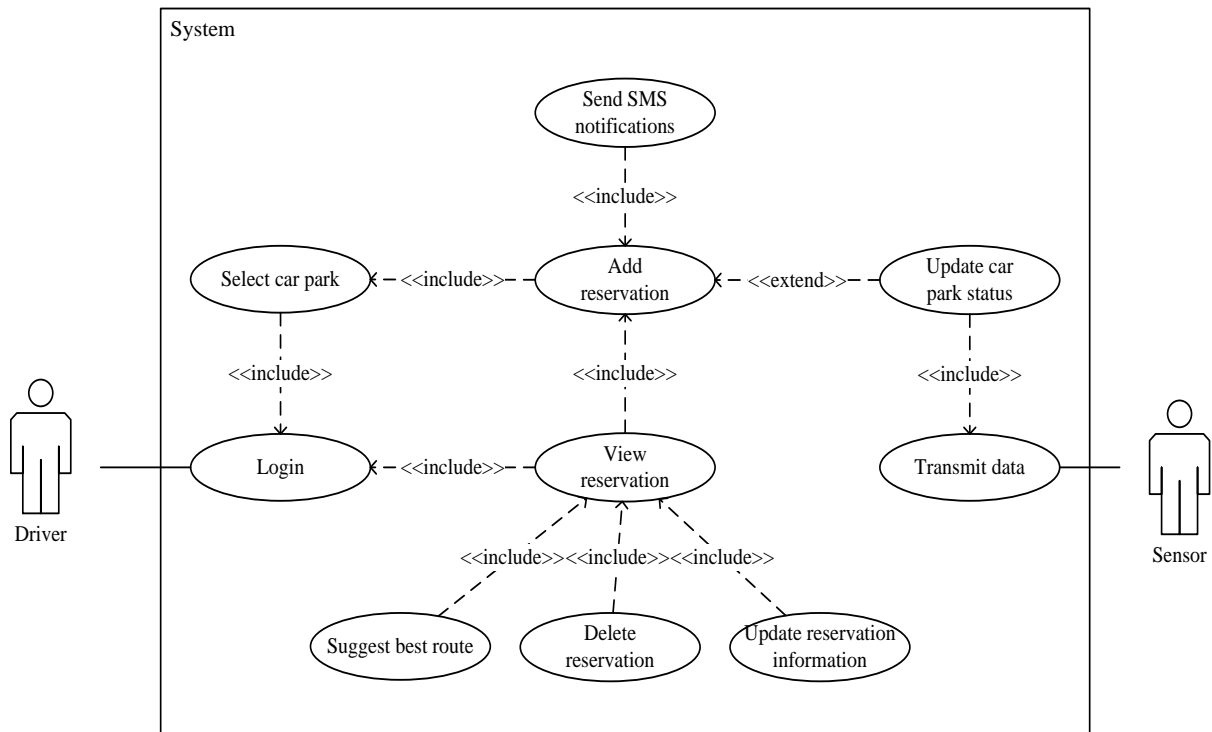


**Figure 3.3 Use case diagram for searching a car park**

Figure 3.3 illustrates the sub use cases related to the main use case – search car park. When selecting the car park, the available parking spaces in a car park need to be known by a driver. For that, timely updates about the status of the car parks need to be provided regularly by the sensors.

### 3.1.3 Reserving parking space use case

The figure 3.4 illustrates the granular use case diagram for “Reserving parking space” use case in the overall use case that involved by the driver.

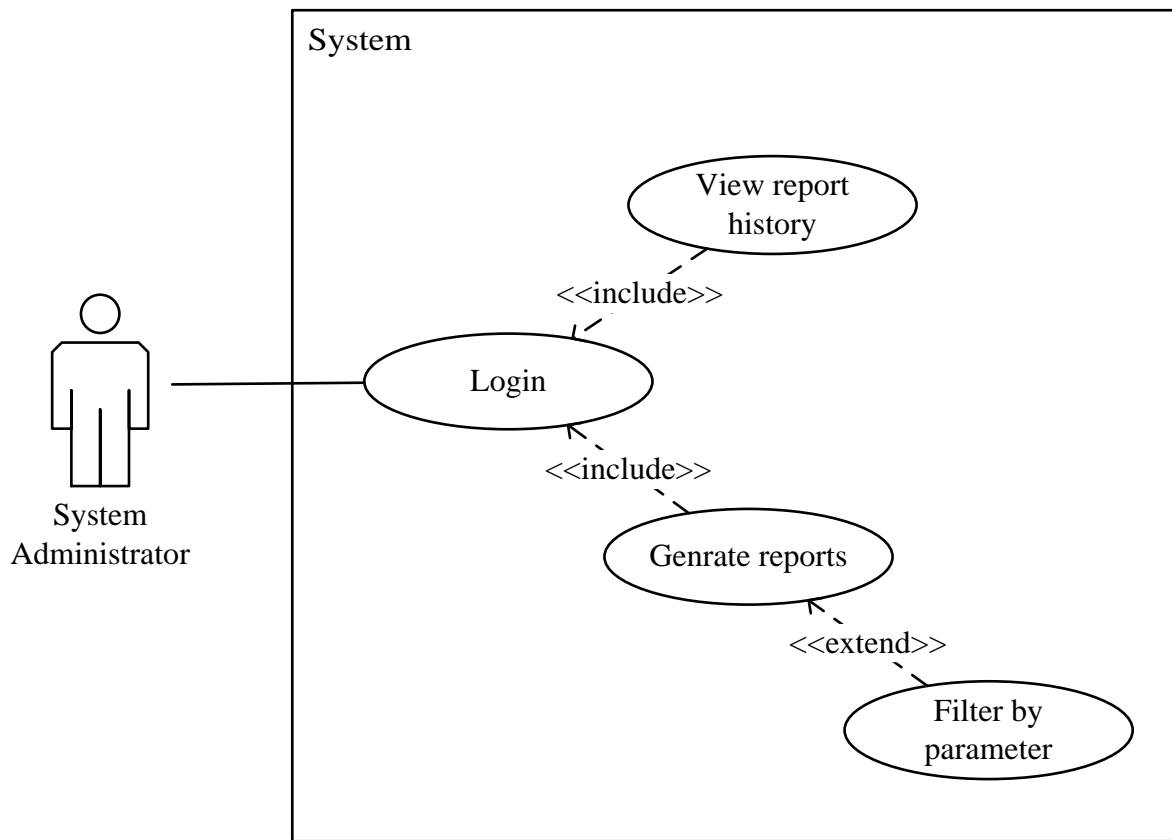


**Figure 3.4 Use case diagram for reserving a parking space**

Figure 3.4 illustrates the sub use cases related to the main use case – reserving a parking space. After selecting a car park, the driver is able to add a reservation or view a previous reservation. A viewed reservation can be updated or deleted. The SMS notification service is enabled after reserving a parking space which will notify the user about the current status of the parking space.

### 3.1.4 Generate reports use case

The figure 3.5 illustrates the granular use case diagram for “Generate reports” use case in the overall use case that involved by the system administrator.



**Figure 3.5 Use case diagram for generating reports**

Figure 3.5 illustrates the sub use cases related to the main use case – generate reports. The system administrator has the authority to view report history. When generating reports, the parameter on which the report needs to be generated should be selected by the administrator.

## 3.2 Use Case Descriptions for the Proposed System

### 3.2.1 User login use case description

The table 3.1 illustrates the Use case description for user login use case.

**Table 3.1 User login use case description**

Use Case ID	UC 01
Use Case	Login
Actors	User
Description	This use case describes the process of logging to the system.
Pre-conditions	User should have created an account in the system.
Basic Course	<ol style="list-style-type: none"><li>1. User request to log in</li><li>2. Enters username and password</li><li>3. Validate login details [if valid] 3.1 Login [if invalid] 3.2 Don't allow login</li></ol>
Alternative course/ Exceptions	2.1 If in the basic course, the actor enters an invalid username and/or password, the system displays an error message. The actor can choose to either return to the beginning of the basic course or cancel the login, at which the use case ends.
Post-condition	Successfully logged in to the system and accesses the allowed level of access.

The use case description of user login illustrated in table 3.1 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.2 Creating user account use case description

The table 3.2 illustrates the use case description for create account use case.

**Table 3.2 Create account use case description**

Use Case ID	UC 02
Use Case	Create account
Actors	User
Description	This use case describes the process of creating an account.
Pre-conditions	-
Basic Course	<ol style="list-style-type: none"><li>1. The system requests that the actor to enter his/her credentials.</li><li>2. Fill the registration form and submit the form.</li><li>3. Verify the submitted details.</li><li>4. Create user account.</li></ol>
Alternative course/ Exceptions	<p>2.1 If in the basic course the actor enters an invalid information, the system displays an error message. The actor can chose to either return to the beginning of the basic course or cancel the registration, at which point the use case ends.</p> <p>2.2 If all necessary fields are not filled, system displays an error message.</p>
Post-condition	Successfully register into the system

The use case description of create account illustrated in table 3.2 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.3 Search car park use case description

The table 3.3 illustrates the use case description for search car park use case.

**Table 3.3 Search car park use case description**

Use Case ID	UC 03
Use Case	Search car park
Actors	Driver
Description	This use case describes how a user can search a car park according to the preferred destination.
Pre-conditions	The user must login to the system
Basic Course	<ol style="list-style-type: none"><li>1. The actor enters the destination where he/she wants to go.</li><li>2. The system will search for the car parks around the destination that are stored in its database.</li><li>3. Actor selects the relevant car park that fulfills the requirement.</li><li>4. System suggests the best route to go to the car park.</li></ol>
Alternative course/ Exceptions	<ol style="list-style-type: none"><li>2.1 If in the basic flow, there are no results found, the actor can chose to return to the beginning of the basic flow, or cancel the searching, at which point the use case ends.</li><li>2.2 If there are no parking space available in the results found, the actor can chose to return to the beginning of the basic flow, or cancel the searching, at which point the use case ends.</li></ol>
Post-condition	Search has successfully completed.

The use case description of search car park illustrated in table 3.3 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.



### 3.2.4 Reserve parking space use case description

The table 3.4 illustrates the use case description for reserve parking space use case.

**Table 3.4 Reserve parking space use case description**

Use Case ID	UC 04
Use Case	Reserving parking space
Actors	Driver
Description	This use case describes how a driver reserves a parking space.
Pre-conditions	The actor must be login to the system. The actor must have selected a car park.
Basic Course	<ol style="list-style-type: none"><li>1. Go to the selected car park.</li><li>2. The actor clicks on add reservation button.</li><li>3. The system requests that the actor to enter details.</li><li>4. Fill and submit the form.</li><li>5. The system saves the reservation.</li><li>6. The actor views the reservation.</li><li>7. The system suggests the best route.</li></ol>
Alternative course/ Exceptions	<p>4.1 If the actor enters invalid information, the system will display an error message. The actor can chose to either return to the beginning of the basic course or cancel the reservation functionality, at which point the use case ends.</p> <p>4.2 If all the necessary fields are not filled, the system displays an error message.</p>
Post-condition	Reservation added successfully. SMS notifications sent to the user to remind the reservation.

The use case description of reserving a parking space illustrated in table 3.4 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.5 Handle payment use case description

The table 3.5 illustrates the use case description for handle payment use case.

**Table 3.5 Handle payments use case description**

Use Case ID	UC 05
Use Case	Handle payments
Actors	Driver
Description	This use case describes how the driver can pay the parking charges.
Pre-conditions	The user must have create an account and/or has a RFID card
Basic Course	<ol style="list-style-type: none"> <li>1. Actor introduce the RFID card</li> <li>2. The system reads the card</li> <li>3. The system checks the time and calculate the payment</li> <li>4. The system issues the invoice</li> <li>5. The system deduct the charges from the subscription in user account</li> </ol>
Alternative course/ Exceptions	<ol style="list-style-type: none"> <li>2.1 If the driver doesn't have the RFID card, he can give his/her NIC</li> <li>2.2 If the subscription has exceeded the minimum limit, the driver can pay manually</li> </ol>
Post-condition	Handle payment has done successfully.

The use case description of handle payments illustrated in table 3.5 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.6 Generate reports use case description

The table 3.6 illustrates the use case description for generate reports use case.

**Table 3.6 Generate reports use case description**

Use Case ID	UC 06
Use Case	Generate reports
Actors	System administrator
Description	This use case describes how the administrator can generate reports through the system
Pre-conditions	The administrator need to be logged in to the system
Basic Course	<ol style="list-style-type: none"> <li>1. The system requests actor to enter the required report type</li> <li>2. System imports data</li> <li>3. The actor selects the parameter</li> <li>4. System generates the report.</li> </ol>
Alternative course/ Exceptions	3.1 If the required parameters are not available system, The actor can chose to either return to the beginning of the basic course or cancel the reservation functionality, at which point the use case ends.
Post-condition	Report generated successfully.

The use case description of generate reports illustrated in table 3.6 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.7 Update user information use case description

The table 3.7 illustrates the use case description for update user information use case.

**Table 3.7 Update user information use case description**

Use Case ID	UC 07
Use Case	Update user information
Actors	User
Description	This use case describes how users can change their details in user profile and delete user profile.
Pre-conditions	The user must be login to the system
Basic Course	<ol style="list-style-type: none"><li>1. The system requests that the actor to edit his/her credentials.</li><li>2. The actor edits the information and save.</li><li>3. The system updates the user details.</li><li>4. The user clicks 'delete account' to delete the account.</li><li>5. The system deletes the account.</li></ol>
Alternative course/ Exceptions	<p>2.1 If the actor enters an invalid information while editing, the system displays an error message. The actor can chose to either return to the beginning of the basic course or cancel the editing information/deleting account, at which point the use case ends.</p> <p>2.2 If all necessary fields are not filled, the system displays an error message.</p>
Post-condition	Successfully updated/deleted user account

The use case description of update user information illustrated in table 3.7 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.8 Update car park status use case description

The table 3.8 illustrates the Use case description for update car park status use case.

**Table 3.8 Update car park status use case description**

Use Case ID	UC 08
Use Case	Update car park status
Actors	Sensor
Description	This use case describes how the car park status is updated by the sensor
Pre-conditions	The sensor must be installed into the system by creating an account for the sensor.
Basic Course	<ol style="list-style-type: none"><li>1. Actor checks whether the parking space is available</li><li>2. If available, the system checks whether the parking space is reserved for the particular time</li><li>3. The system displays the parking space status</li></ol>
Alternative course/ Exceptions	
Post-condition	Update the car park status successfully.

The use case description of update car park status illustrated in table 3.8 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.9 Create sensor account use case description

The table 3.10 illustrates the use case description for create sensor account use case.

**Table 3.9 Create sensor account use case description**

Use Case ID	UC 09
Use Case	Create sensor account
Actors	Car park
Description	This use case describes the process of creating a sensor account
Pre-conditions	The car park must be logged in to the system
Basic Course	<ol style="list-style-type: none"><li>1. The system requests the actor to enter sensor credentials.</li><li>2. Fill the registration form and submit the form.</li><li>3. Verify the submitted details.</li><li>4. Create sensor account.</li></ol>
Alternative course/ Exceptions	<p>2.1 If in the basic course the actor enters an invalid information, the system displays an error message. The actor can chose to either return to the beginning of the basic course or cancel the registration, at which point the use case ends.</p> <p>2.2 If all necessary fields are not filled, system displays an error message.</p>
Post-condition	The sensor account created successfully.

The use case description of create sensor account illustrated in table 3.9 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.10 Updating reservation use case description

The table 3.11 illustrates the use case description for updating reservation use case.

**Table 3.10 Updating reservation use case description**

Use Case ID	UC 10
Use Case	Updating reservation
Actors	Driver
Description	This use case describes how drivers can change the reservation details and delete a reservation
Pre-conditions	The driver must be logged in to the system
Basic Course	<ol style="list-style-type: none"><li>1. The system requests that the actor to edit the reservation details</li><li>2. The actor edits the information and save.</li><li>3. The system updates the reservation details.</li><li>4. The user clicks 'delete reservation' to delete the reservation</li><li>5. The system deletes the reservation.</li></ol>
Alternative course/ Exceptions	<ol style="list-style-type: none"><li>2.1 If the actor enters an invalid information while editing, the system displays an error message. The actor can chose to either return to the beginning of the basic course or cancel the editing information/deleting account, at which point the use case ends.</li><li>2.2 If all necessary fields are not filled, the system displays an error message.</li></ol>
Post-condition	The reservation updated/deleted successfully

The use case description of updating reservation illustrated in table 3.10 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.

### 3.2.11 Sending SMS notification use case description

The table 3.12 illustrates the use case description for sending SMS notification use case.

**Table 3.11 Sending SMS notification use case description**

Use Case ID	UC 11
Use Case	Sending SMS notifications
Actors	Car park
Description	This use case describes how the SMS notifications about a particular reservation are sent to the driver from the car park.
Pre-conditions	The driver has made a reservation
Basic Course	<ol style="list-style-type: none"><li>1. The system checks the upcoming reservations</li><li>2. The system checks the date and time of the reservation</li><li>3. Send SMS notifications to the driver</li></ol>
Alternative course/ Exceptions	3.1 If the SMS notification does not received by the driver, the mobile app will display the notifications about the reservation in the driver's account.
Post-condition	Send SMS notifications successfully.

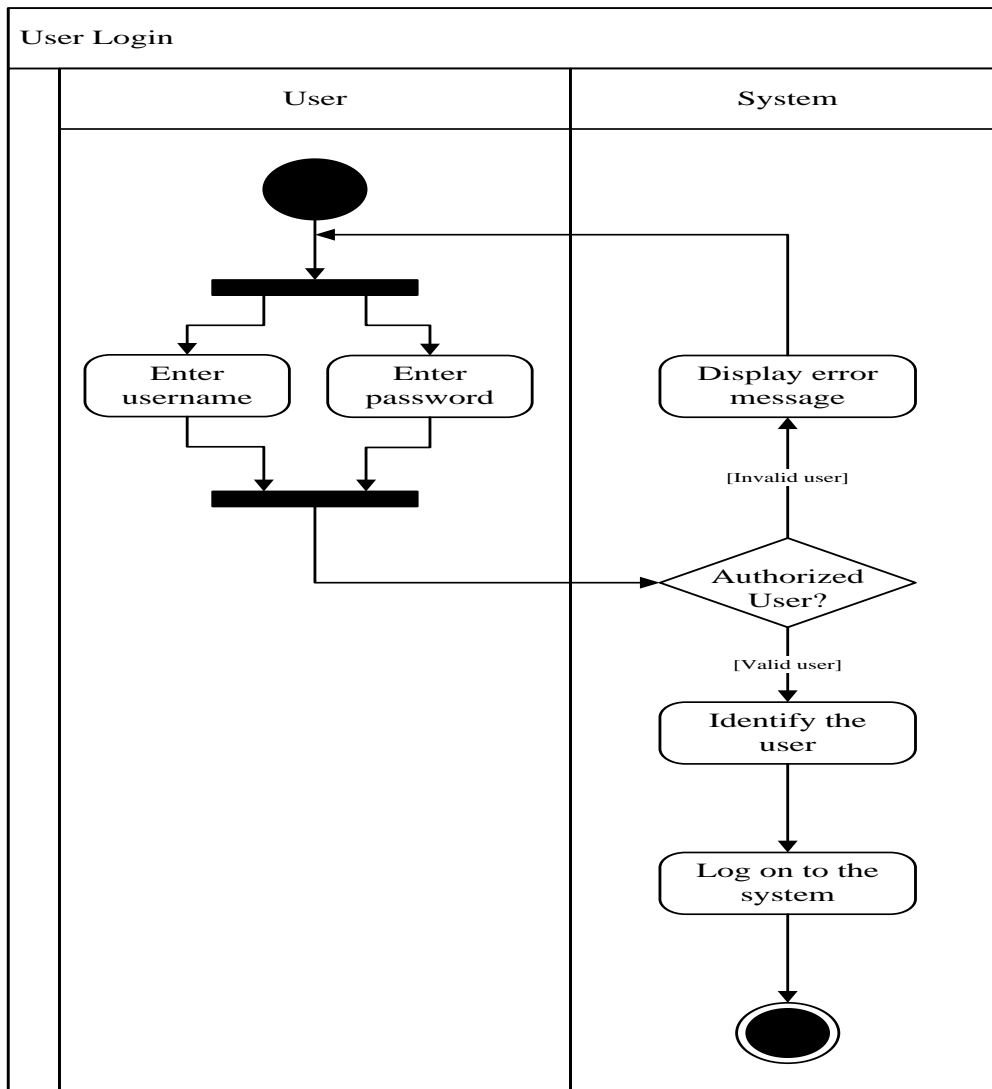
The use case description of sending SMS notifications illustrated in table 3.11 describes the actors, the pre-conditions, the basic course, alternative courses or exceptions if the process deviates from the basic course and the post conditions that need to be followed and achieved in response to the actions of the actor.



### 3.3 Activity Diagram for the Proposed System

#### 3.3.1 Activity diagram for user login

Figure 3.6 further illustrates the actions occurred in login use case in the use case diagram.

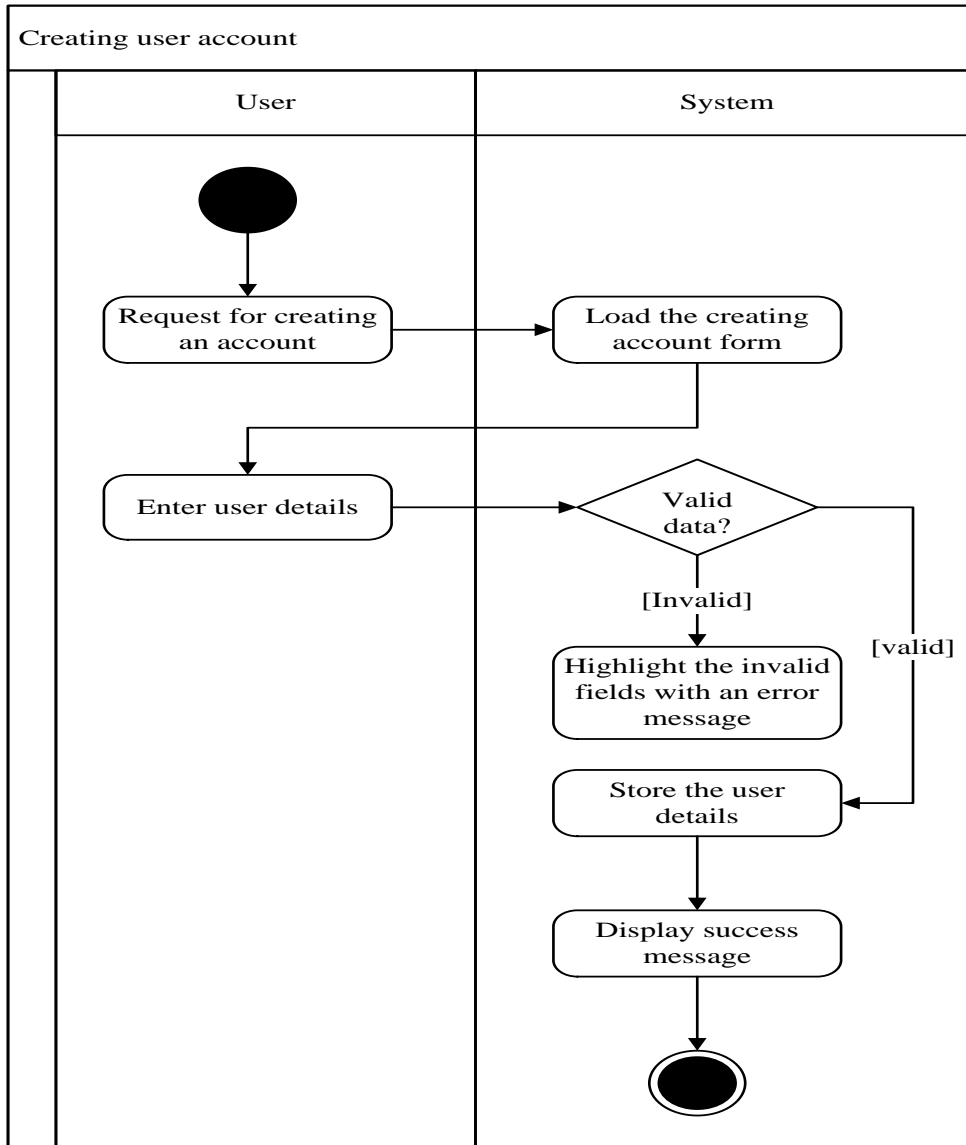


**Figure 3.6 Activity diagram for user login**

The activity diagram for user login illustrated in figure 3.6 describes the flow of actions for user login activity by the relevant actor – the users and the system in an elaborated manner. To login to the system, the user (car park or the driver) needs to enter their username and passwords, and if the user is not authorized, specific actions will be taken by the system.

### 3.3.2 Activity diagram for creating user accounts

Figure 3.7 further illustrates the actions occurred in create user accounts use case in the use case diagram.

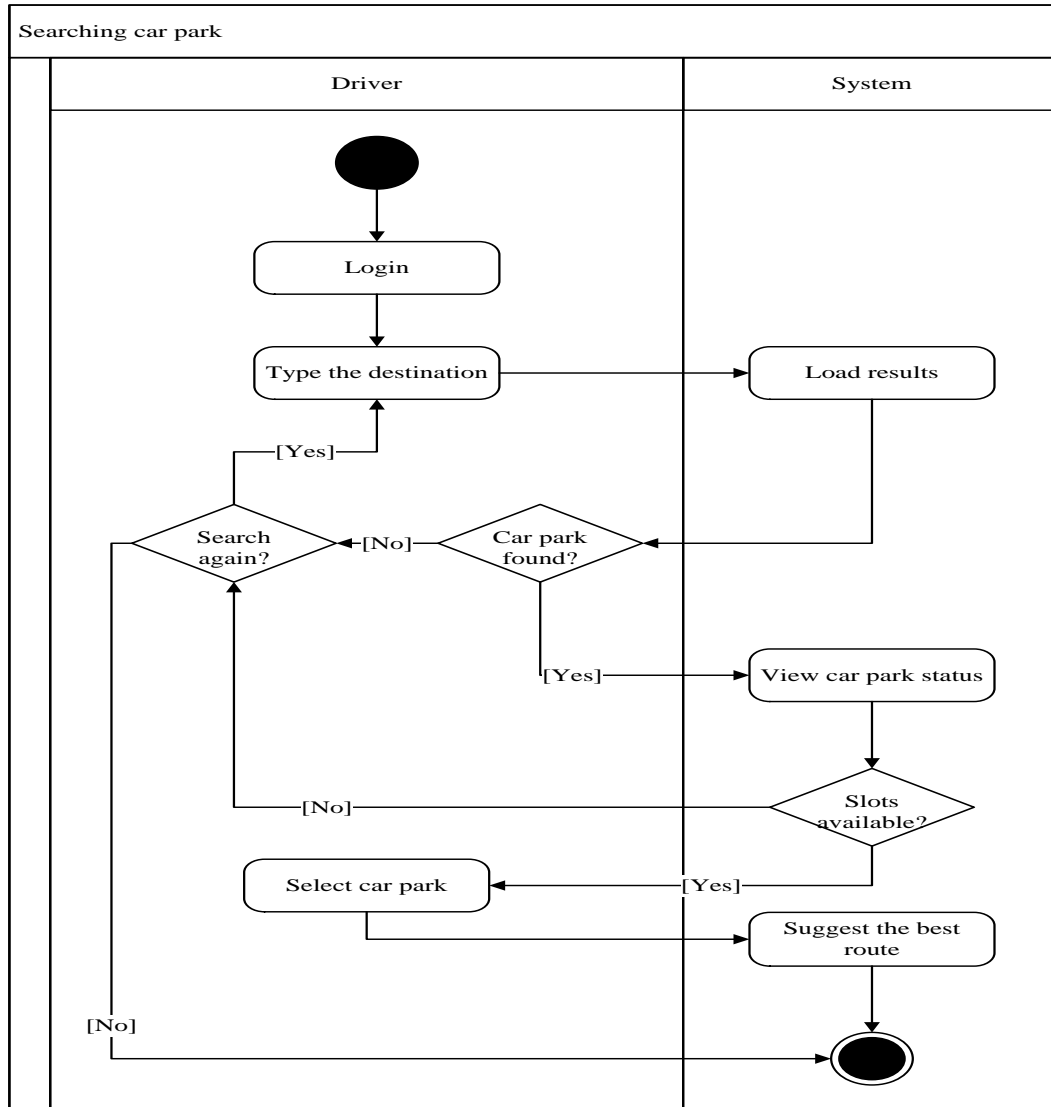


**Figure 3.7** Activity diagram for creating user account

The activity diagram for creating user account illustrated in figure 3.7 describes the flow of actions for creating account activity by the relevant actor – the users and the system in an elaborated manner. To create the account, the users need to submit the requested details and if the details are not valid, specific actions will be taken by the system.

### 3.3.3 Activity diagram for searching a car park

Figure 3.8 further illustrates the actions occurred in search car park use case in the use case diagram.



**Figure 3.8 Activity diagram for searching a car park**

The activity diagram for searching a car park illustrated in figure 3.8 describes the flow of actions for searching a car park activity by the relevant actor – the driver and the system in an elaborated manner. To search the car park, the destination need to be entered by the driver, and if a car park is found he can check the status (whether parking slots are available). The system will suggest the best route for the car park. If slots are not available, the driver can search for another car park.

### 3.3.4 Activity diagram for reserving a parking space

Figure 3.9 further illustrates the actions occurred in reserving parking space use case in the use case diagram.

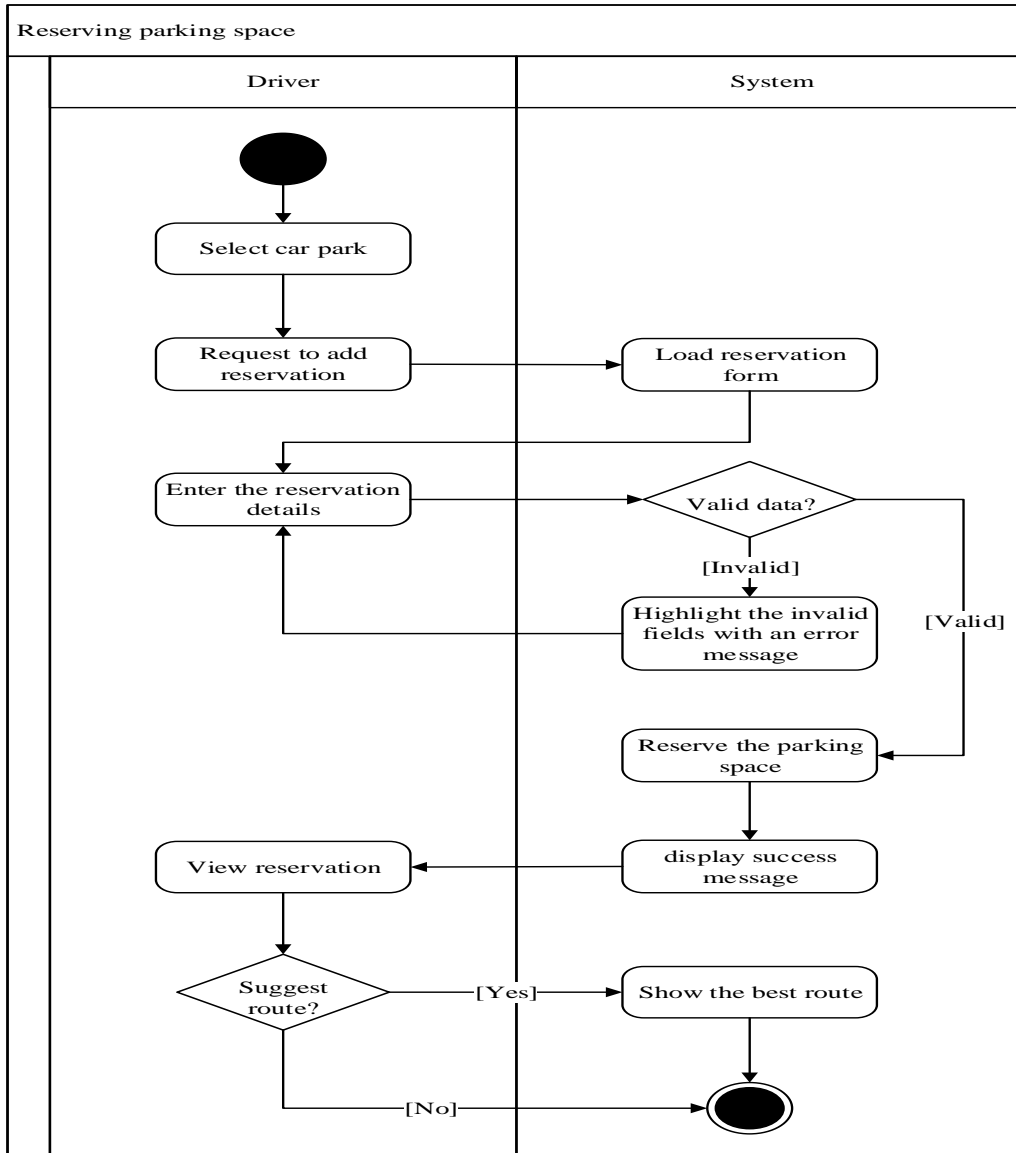
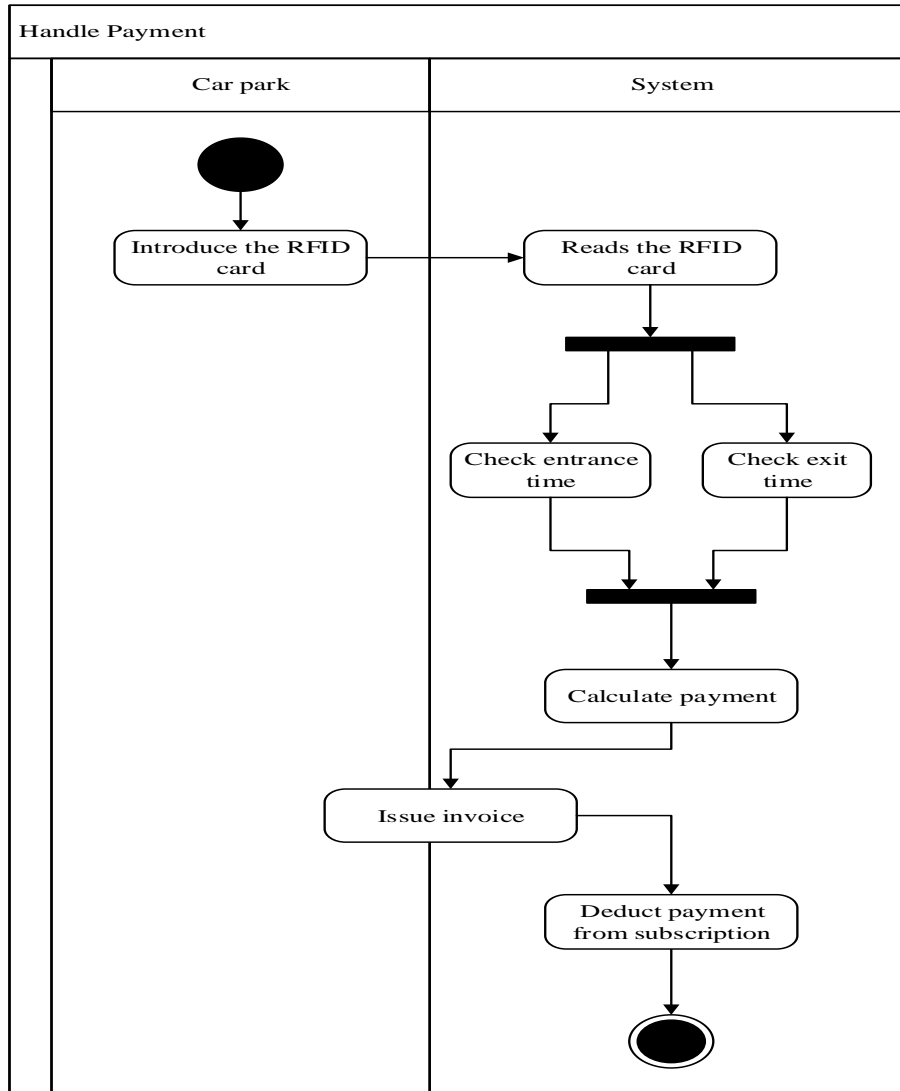


Figure 3.9 Activity diagram for reserving a parking space

The activity diagram for reserving a parking space illustrated in figure 3.9 describes the flow of actions for reserving a parking space activity by the relevant actor – the driver and the system in an elaborated manner. To reserve a parking space, the driver needs to submit the requested details such as time in, time out and the date. Specific actions will be taken by the system if the details are not valid. The best route will be shown to the driver by the system.

### 3.3.5 Activity diagram for payment handling

Figure 3.10 further illustrates the actions occurred in handle payment use case in the use case diagram.

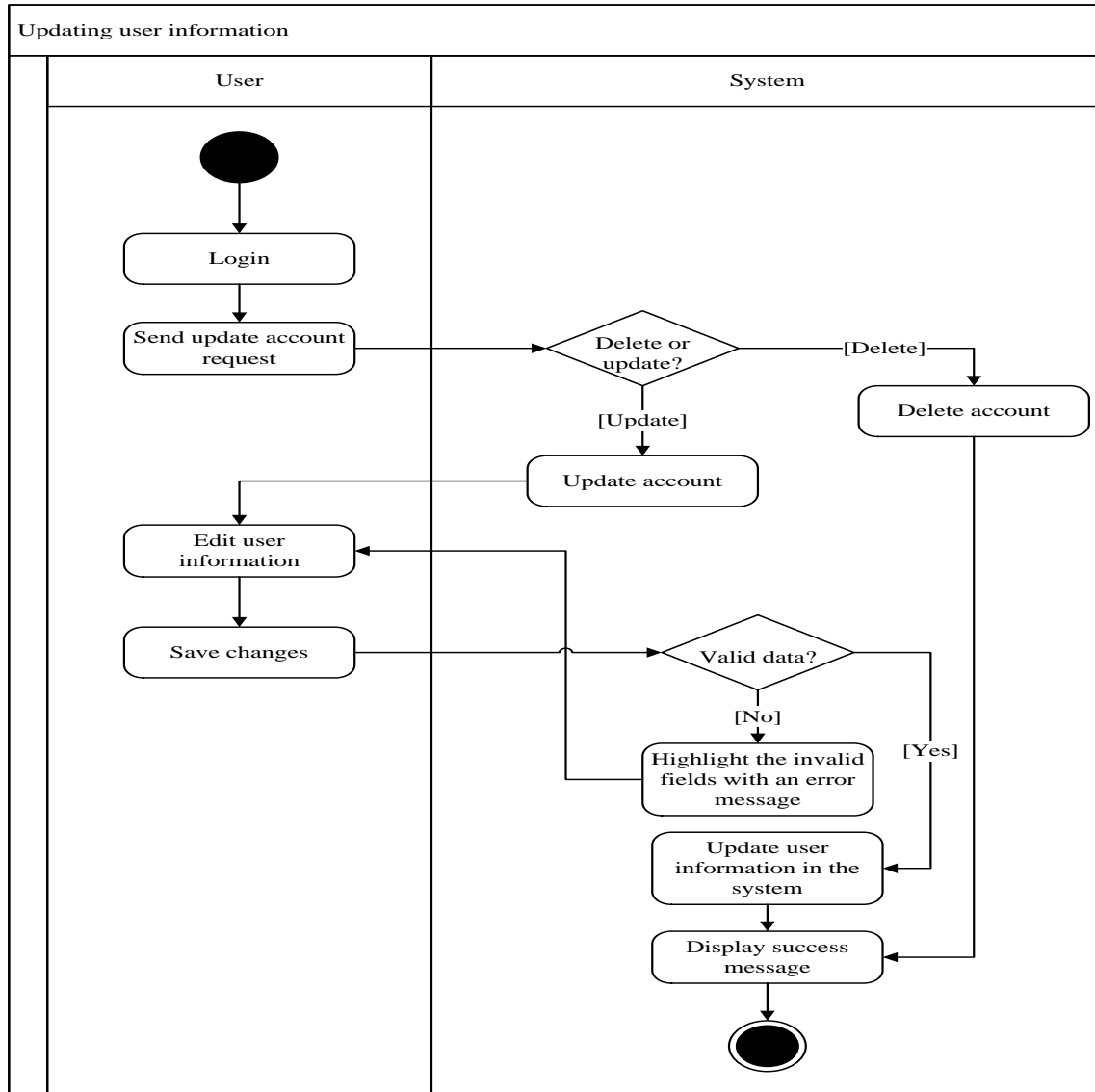


**Figure 3.10 Activity diagram for handle payment**

The activity diagram for handling payment illustrated in figure 3.10 describes the flow of actions for payment handling activity by the relevant actor – the car park and the system in an elaborated manner. The RFID card will be read by the system and the duration will be calculated by checking the entrance and exit time of the vehicle to calculate the payment. The payment amount will be credited to the driver's account and will be deducted from the subscription.

### 3.3.6 Activity diagram for updating user information

Figure 3.11 further illustrates the actions occurred in update user information use case in the use case diagram.



**Figure 3.11 Activity diagram for updating user information**

The activity diagram for updating user information illustrated in figure 3.11 describes the flow of actions for updating user information activity by the relevant actor – the user and the system in an elaborated manner. The user can choose whether to delete or update account. When updating, the details need to be entered by the user and the validity of the data is checked by the system. The updated information will be saved by the system.

### 3.3.7 Activity diagram for generating reports

Figure 3.12 further illustrates the actions occurred in generating reports use case in the use case diagram.

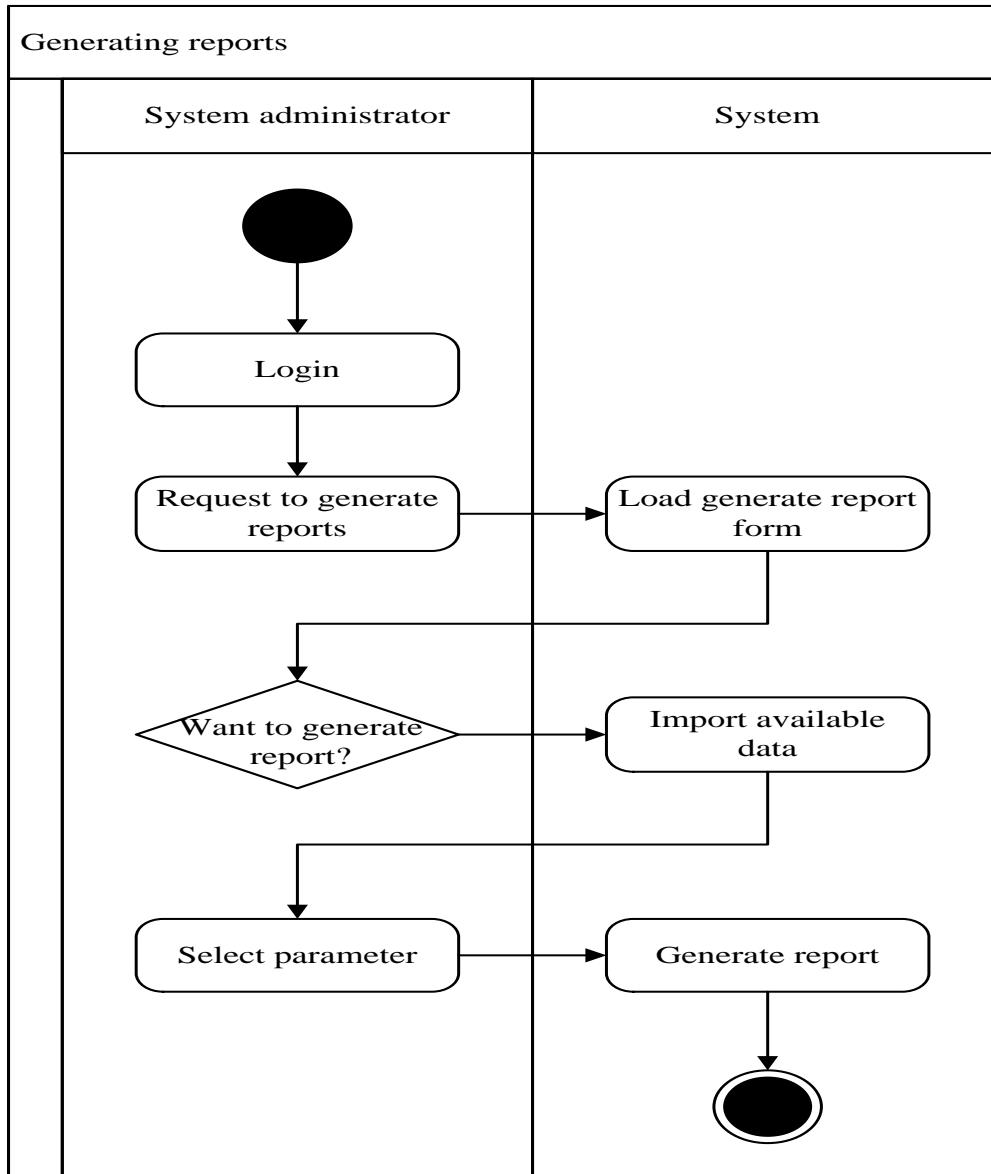
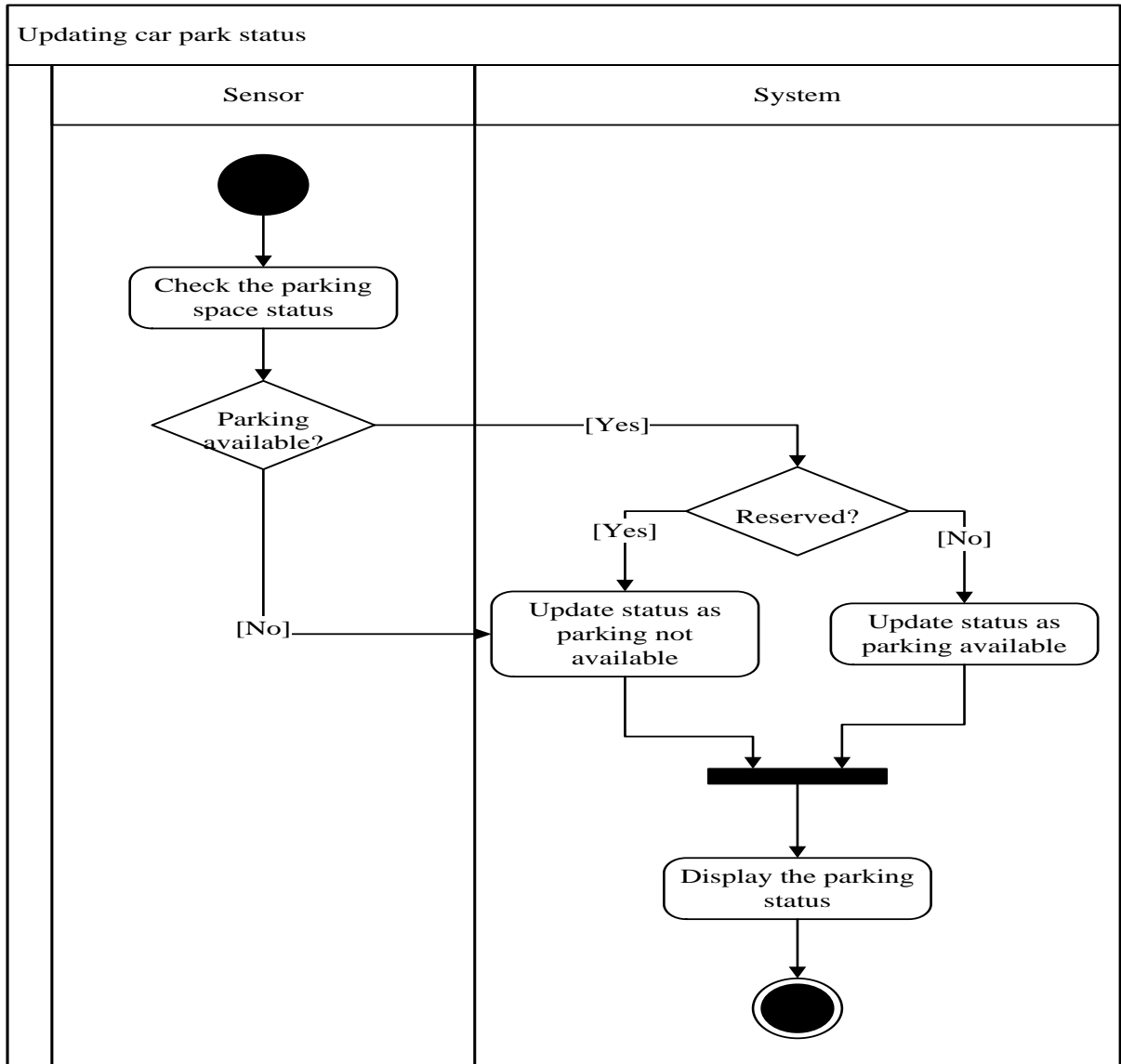


Figure 3.12 Activity diagram for generating reports

The activity diagram for generating reports illustrated in figure 3.12 describes the flow of actions for generating reports activity by the relevant actor – the system administrator and the system in an elaborated manner. A parameter need to be selected by the administrator to generate the report.

### 3.3.8 Activity diagram for updating car park status

Figure 3.13 further illustrates the actions occurred in updating car park status use case in the use case diagram.



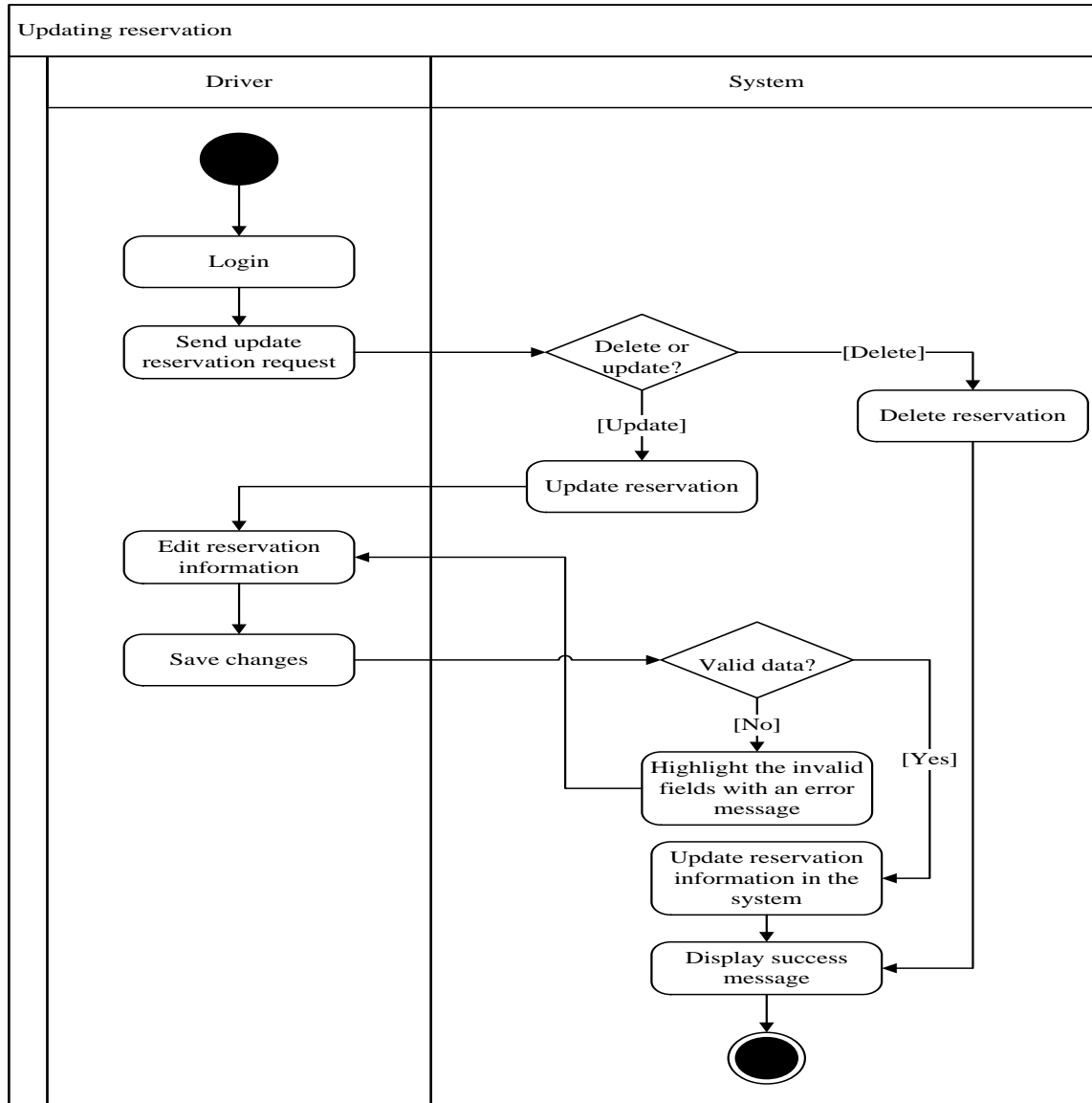
**Figure 3.13 Activity diagram for updating car park status**

The activity diagram for updating car park status illustrated in figure 3.13 describes the flow of actions for updating car park status activity by the relevant actor – the sensor and the system in an elaborated manner. Data will be transmitted by the sensor stating whether the slot is available or not. The system will then check whether the particular slot is reserved or not. Then the system will display the parking status.



### 3.3.9 Activity diagram for updating reservation

Figure 3.14 further illustrates the actions occurred in update reservation use case in the use case diagram.



**Figure 3.14 Activity diagram for updating reservation**

The activity diagram for updating reservation illustrated in figure 3.14 describes the flow of actions for updating reservation activity by the relevant actor – the driver and the system in an elaborated manner. The driver can choose whether to delete or update the reservation. When updating, the details need to be entered by the drivers and the validity of the data is checked by the system. The updated information will be saved by the system.

### 3.3.10 Activity diagram for sending SMS notification

Figure 3.15 further illustrates the actions occurred in sending SMS notifications use case in the use case diagram.

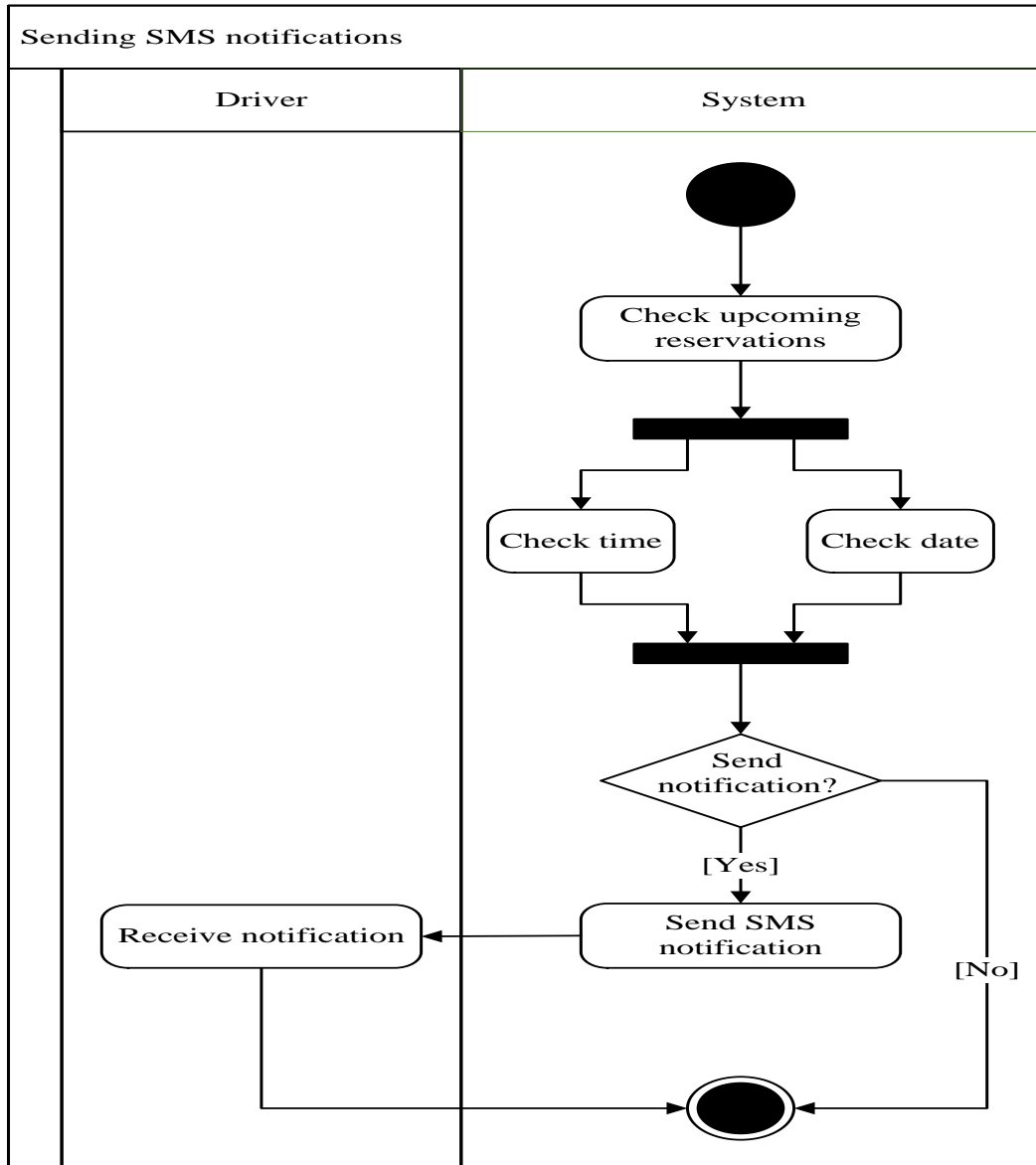


Figure 3.15 Activity diagram for sending SMS notifications

The activity diagram for sending SMS notifications illustrated in figure 3.15 describes the flow of actions for sending SMS notification activity by the relevant actor – the driver and the system in an elaborated manner. The time and date of the reservation will be checked by the system. If it is nearly the time, an SMS notification will be sent to the driver from the system.

## 3.4 Sequence Diagrams for the Proposed System

Sequence diagrams show how the system interacts with the actors in a use case functionality. Each actor is represented with a horizontal lifeline and the data transactions are drawn from one life line to another or within one lifeline. Following sequence diagrams describe some of the main use cases which are a bit difficult to understand with only having use case descriptions.

### 3.4.1 Sequence diagram for user login

Figure 3.16 below illustrates the sequence of actions for user login.

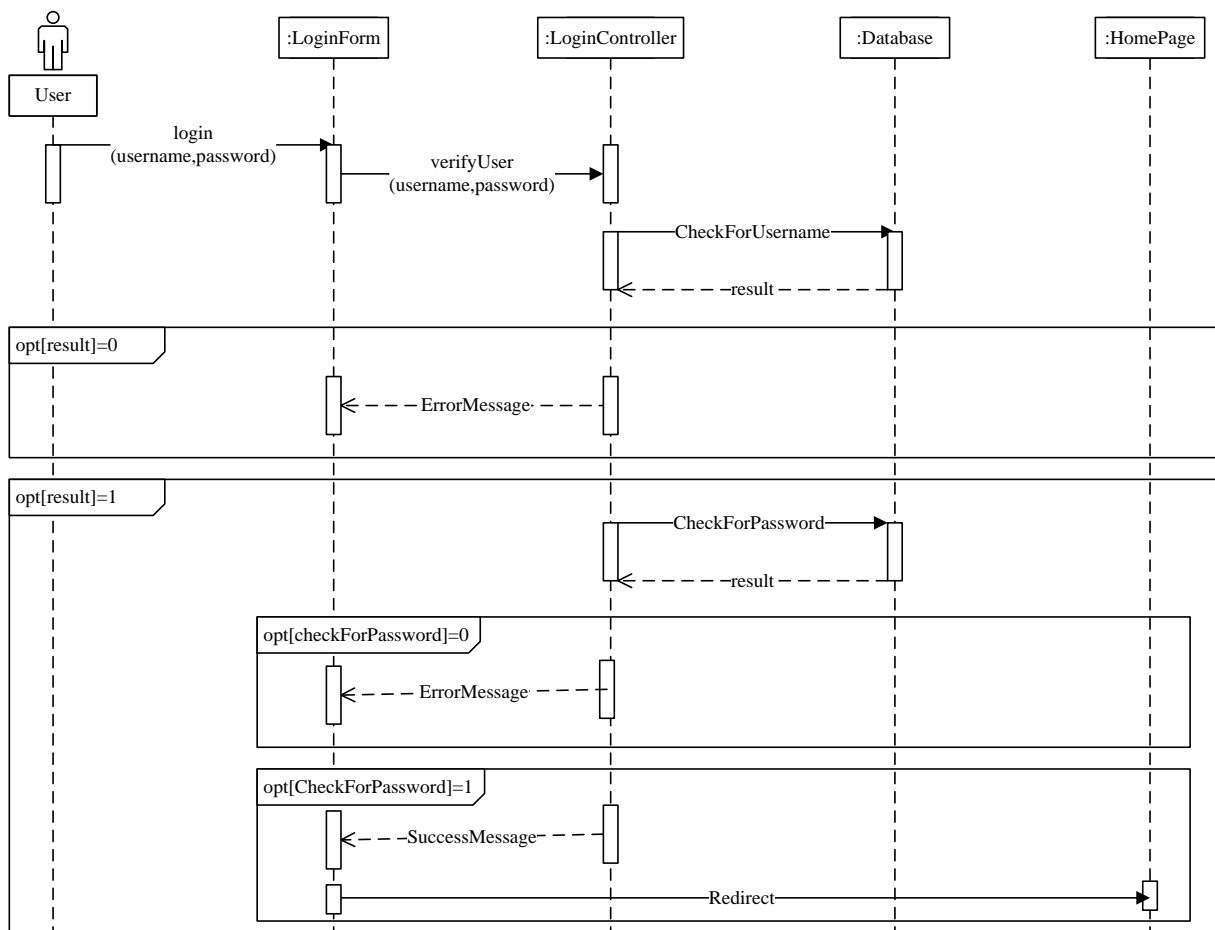


Figure 3.16 Sequence diagram for user login

The sequence diagram for user login illustrated in figure 3.16 describes the objects and classes involved in the user login scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

### 3.4.2 Sequence diagram for creating user account

Figure 3.17 below illustrates the sequence of actions for creating user account.

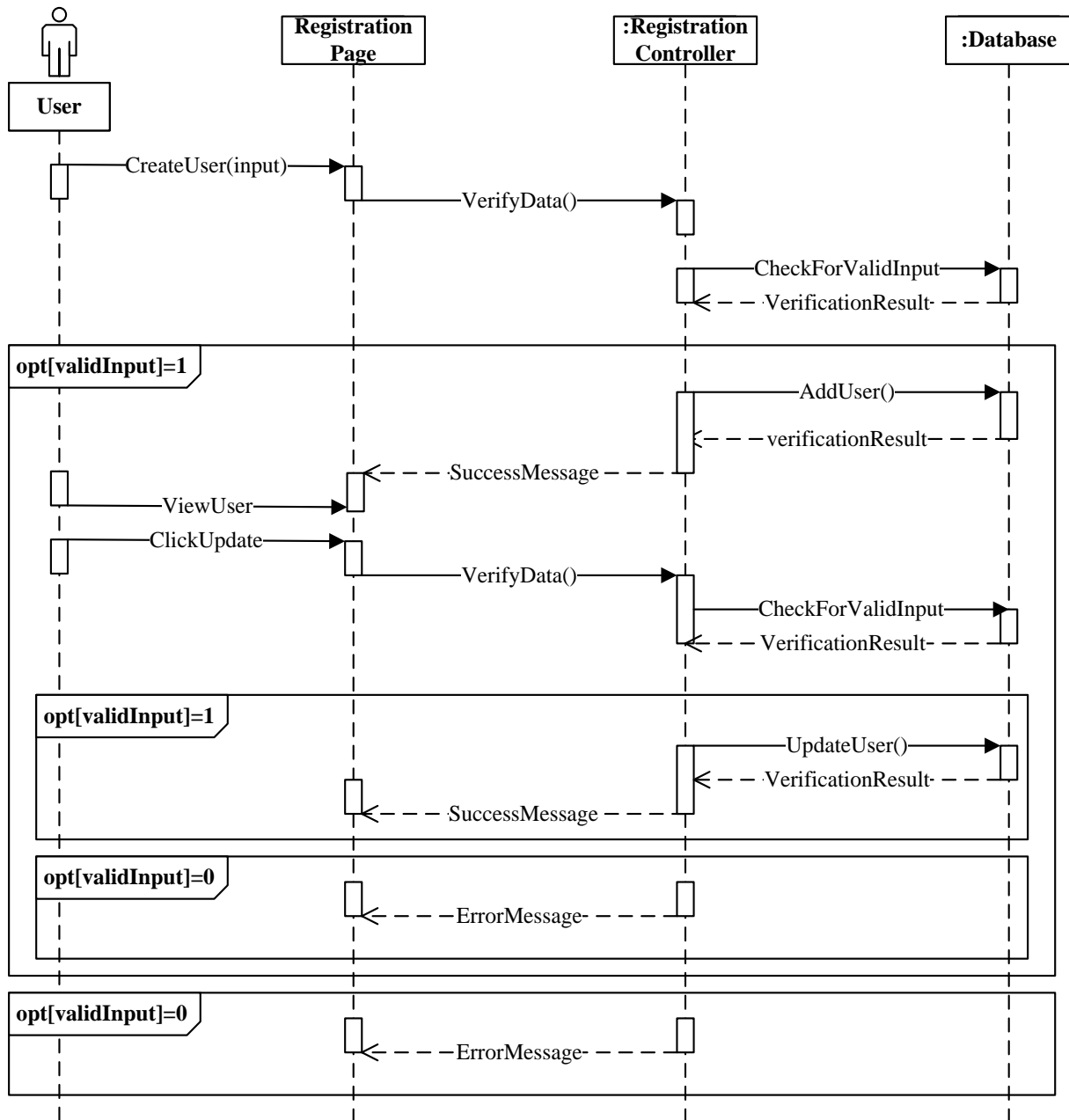
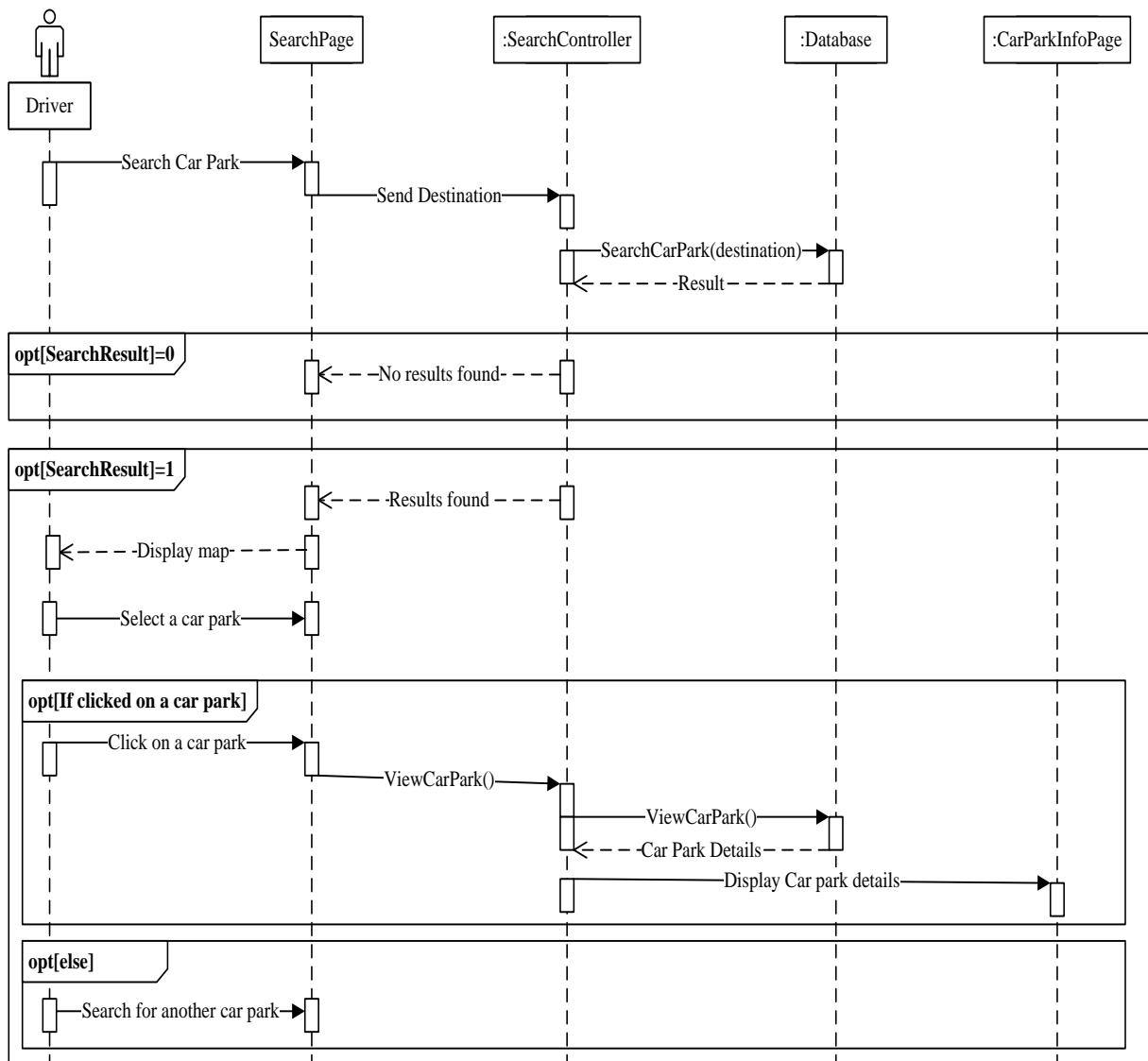


Figure 3.17 Sequence diagram for creating user account

The sequence diagram for creating user account illustrated in figure 3.17 describes the objects and classes involved in the creating user account scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

### 3.4.3 Sequence diagram for searching a car park

Figure 3.18 below illustrates the sequence of actions for searching a car park.

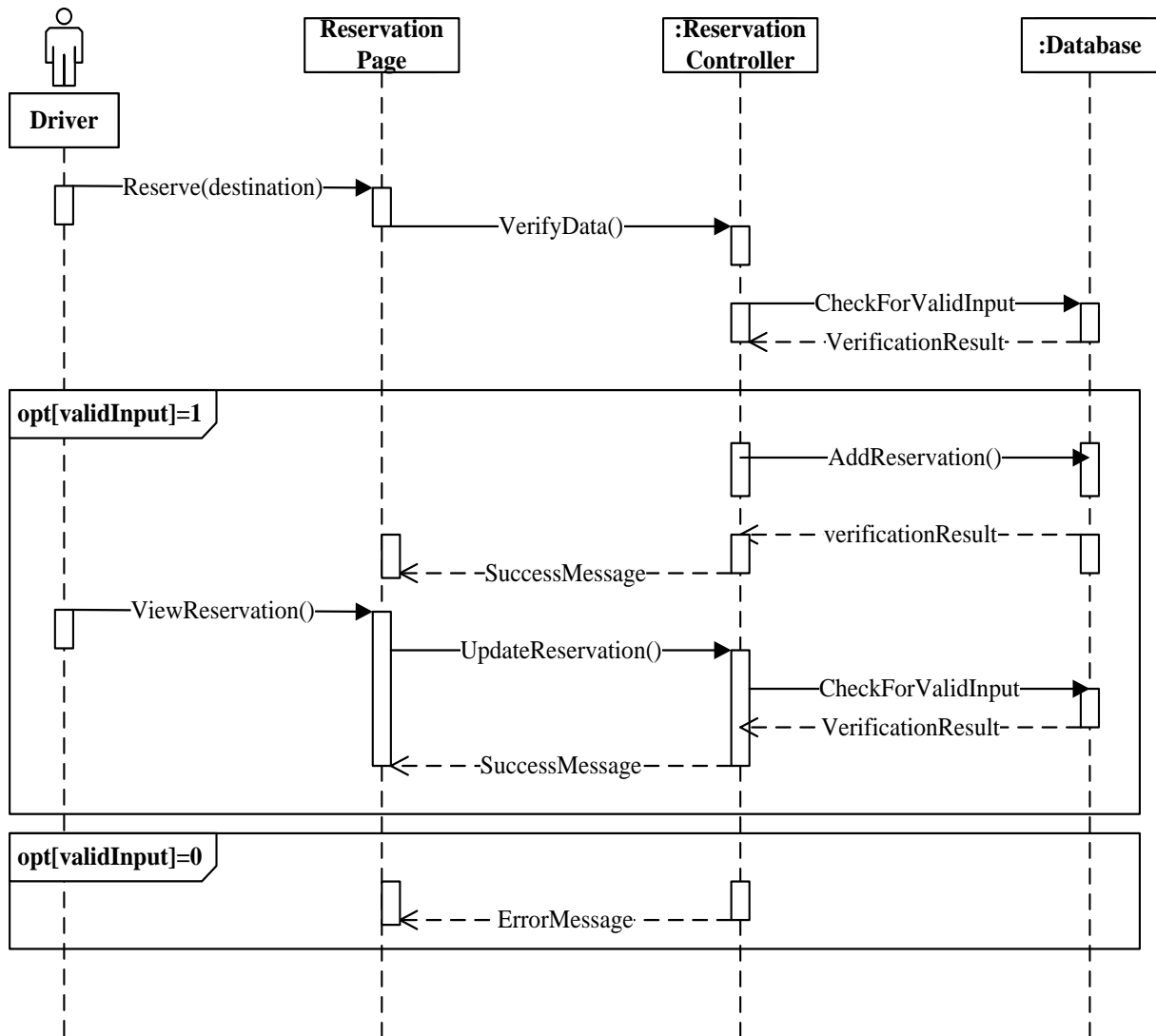


**Figure 3.18 Sequence diagram for searching a car park**

The sequence diagram for searching a car park illustrated in figure 3.18 describes the objects and classes involved in the searching a car park scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

### 3.4.4 Sequence diagram for reserving a parking space

Figure 3.19 below illustrates the sequence of actions for reserving a car park.



**Figure 3.19** Sequence diagram for reserving a parking space

The sequence diagram for reserving a parking space illustrated in figure 3.19 describes the objects and classes involved in the reserving a parking space scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

### 3.4.5 Sequence diagram for payment handling

Figure 3.20 below illustrates the sequence of actions for payment handling.

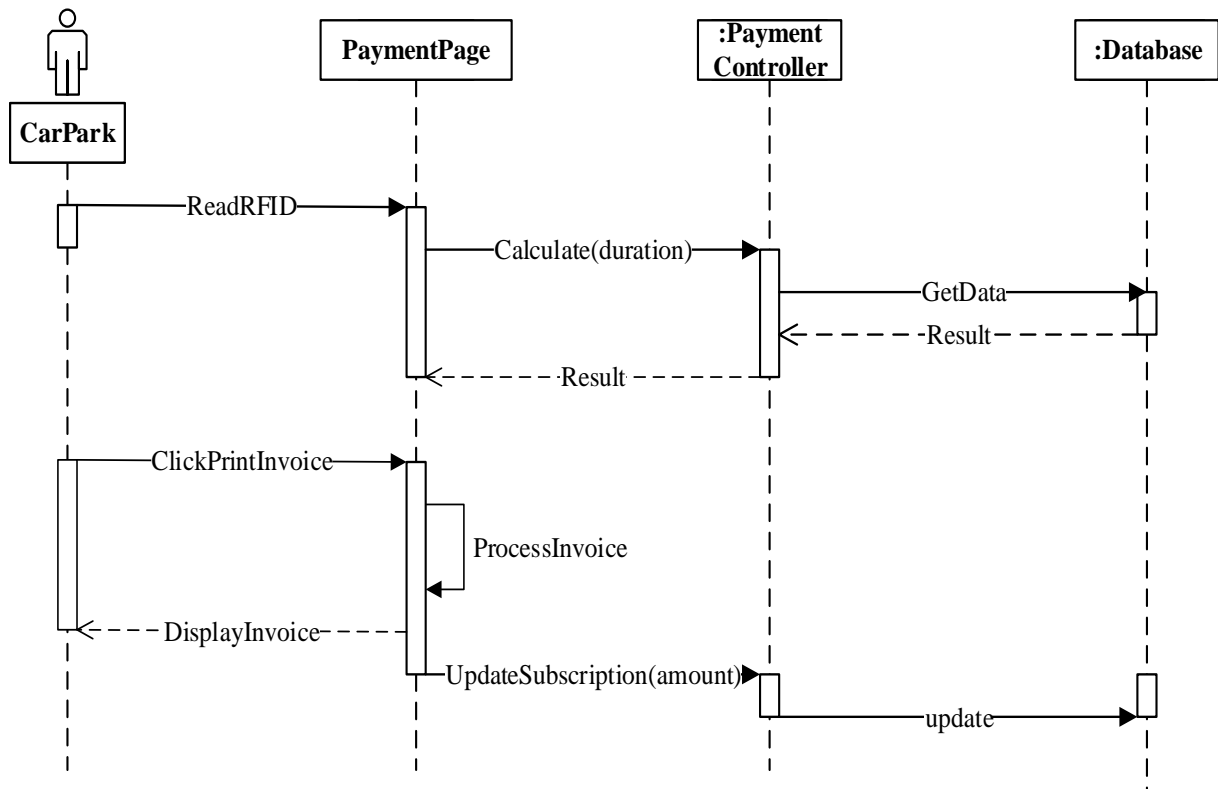


Figure 3.20 Sequence diagram for payment handling

The sequence diagram for payment handling illustrated in figure 3.20 describes the objects and classes involved in the payment handling scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.



### 3.4.6 Sequence diagram for generating reports

Figure 3.21 below illustrates the sequence of actions for generating reports.

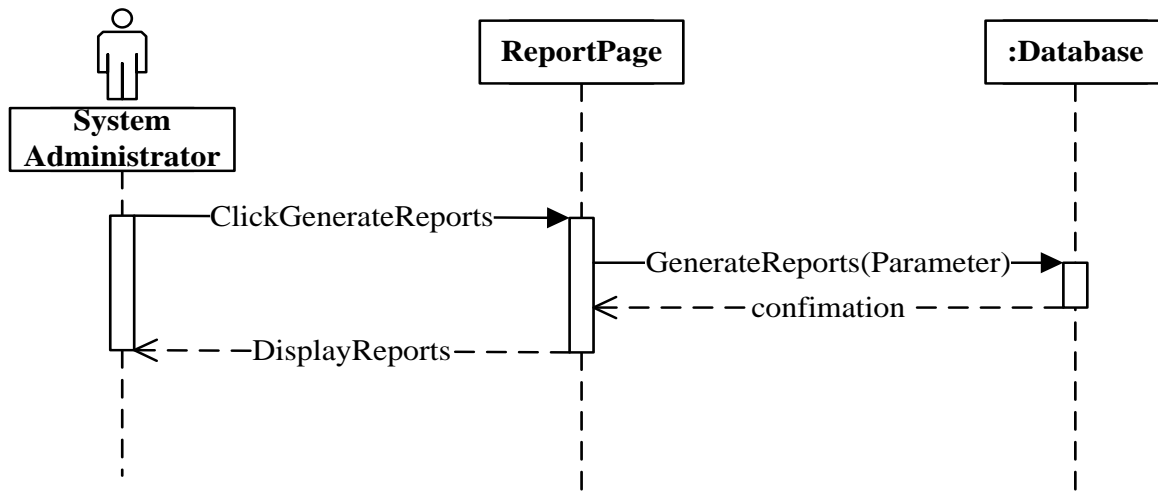
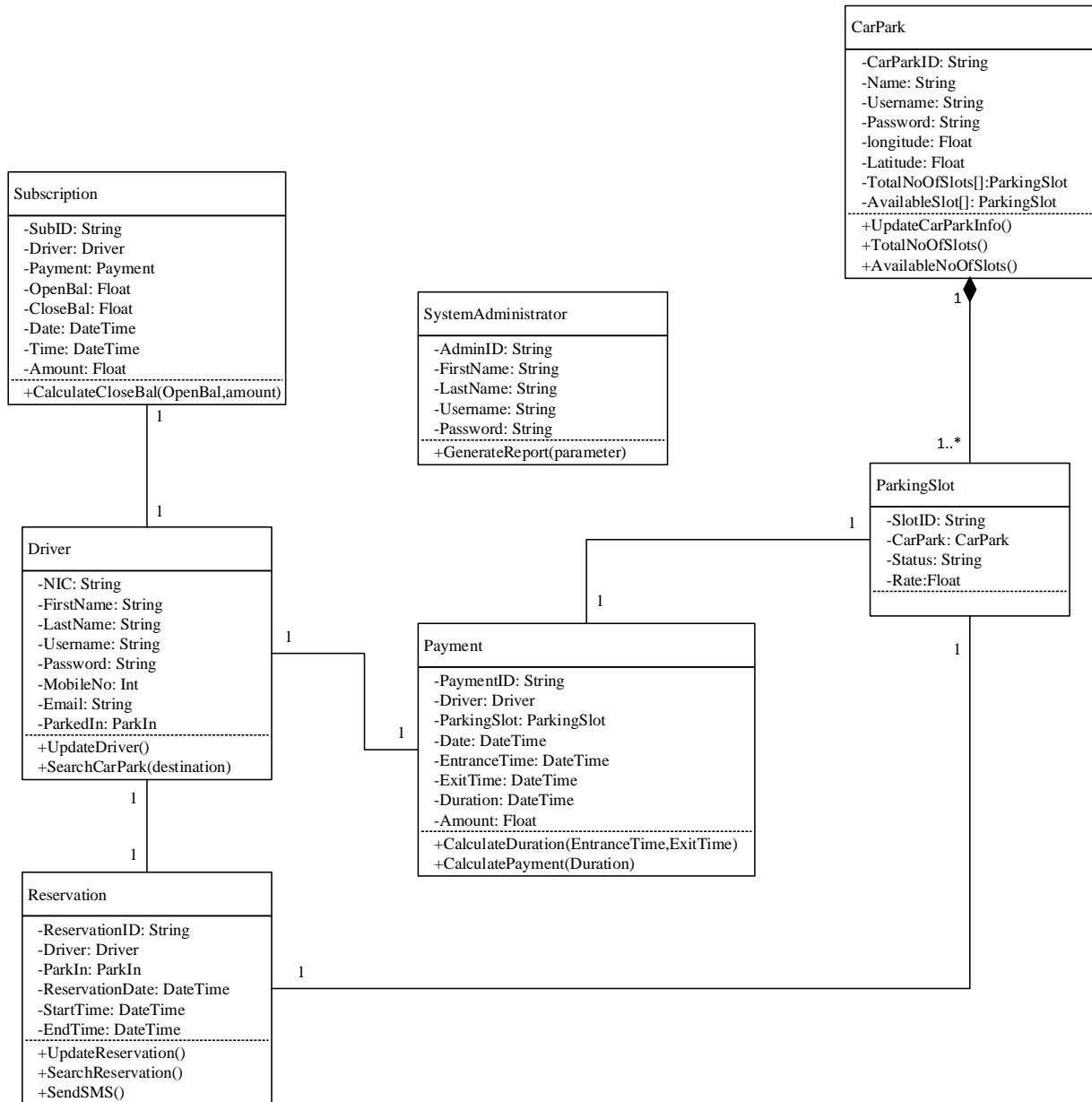


Figure 3.21 Sequence diagram for generating reports

The sequence diagram for generating reports illustrated in figure 3.21 describes the objects and classes involved in the generating reports scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

### 3.5 Class Diagram for Proposed System

Class diagram in figure 3.22 describes the structure of a proposed system by showing the system's classes their attributes, operations and the relationships among the classes.



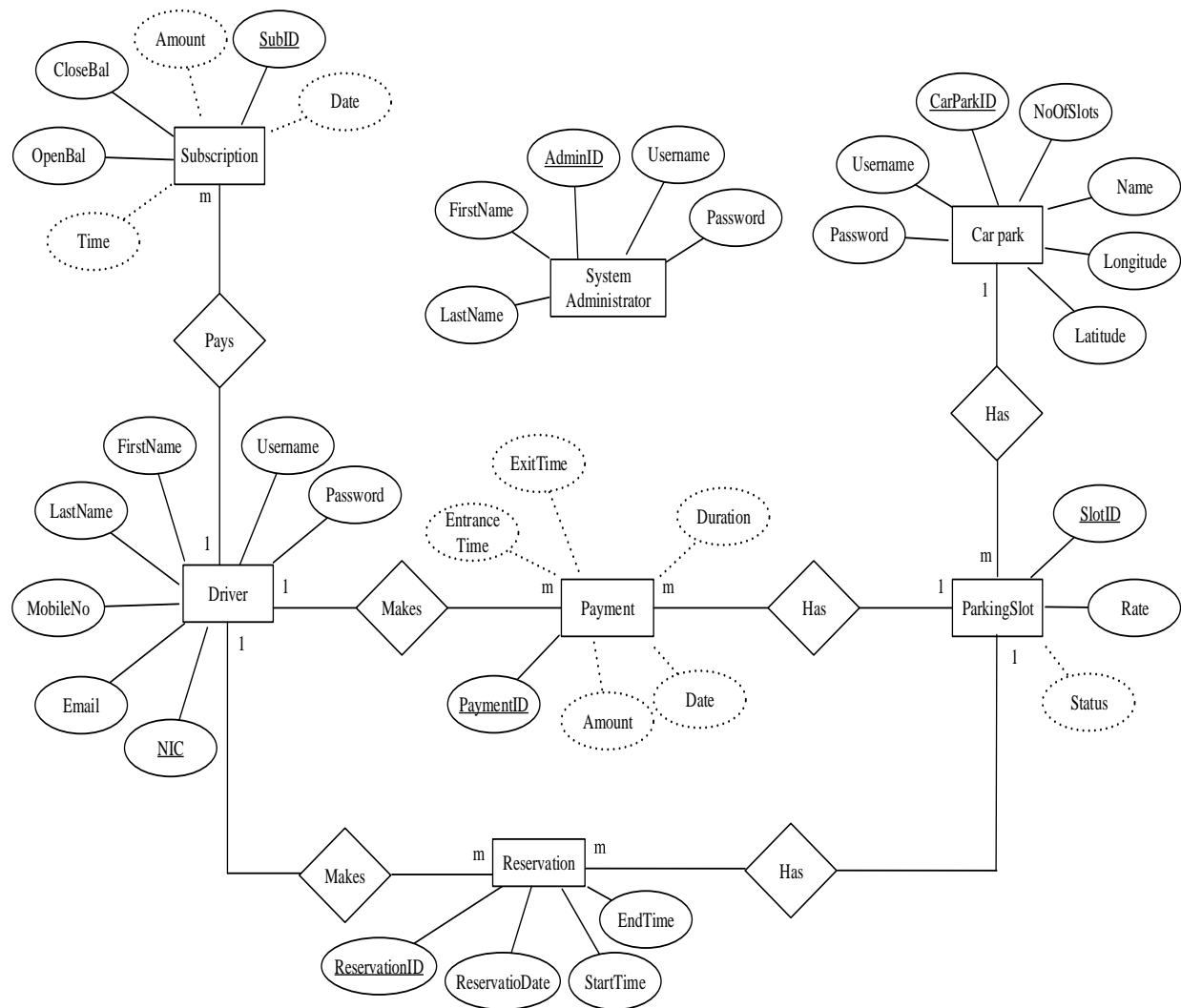
**Figure 3.22 Class diagram for the proposed system**

The class diagram for the proposed system illustrated in figure 3.22 describes the attributes, methods and the relationships among the 7 classes – Driver class, Reservation class, Payment class, CarPark class, Subscription class, ParkingSlot class and the SystemAdministrator class.

## 3.6 Database Design

### 3.6.1 Entity - Relationship model

The entity relationship diagram shown below in figure 3.23 describe the relationship between entities.



**Figure 3.23 Entity – Relationship model**

The Entity – Relationship model illustrated in the figure 3.23 describes the relationships of entity sets stored in a database. The entities identified are Driver, Payment, CarPark, ParkingSlot, Reservation, Subscription, and SystemAdministrator. The relevant attributes of each entity are shown around the entity.



### 3.6.2 Normalized database design

Data base relationship diagram shown in figure 3.24 illustrates the relationship between the data tables in the database.

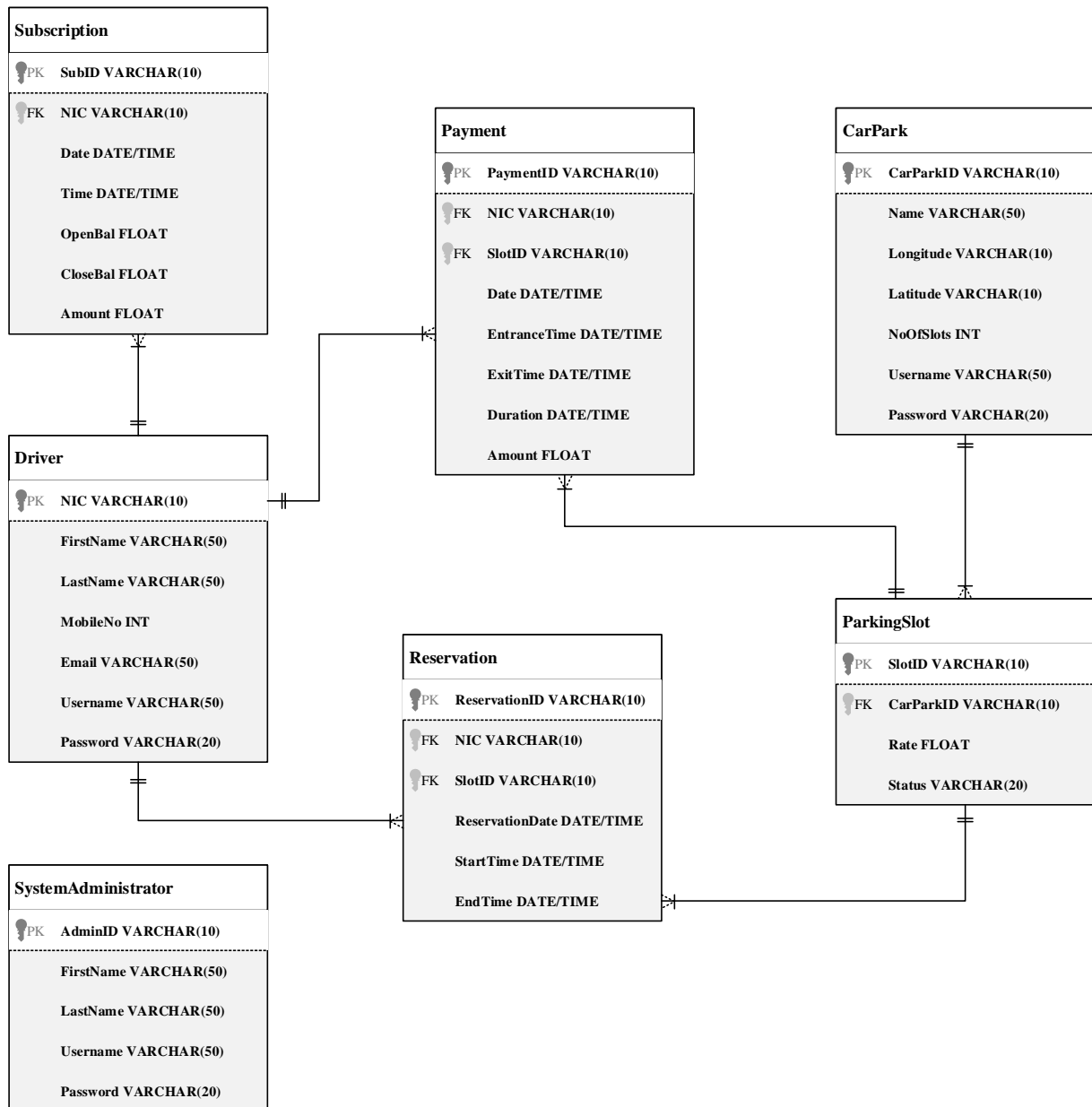


Figure 3.24 Normalized database design

The normalized database illustrated in figure 3.24 describes the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity.

## 3.7 GUI Design

Graphical User Interface (GUI) is one of the key component in a mobile application that communicates with the users of the system. User friendly GUI is the one of the major non-functional requirement of this system.

The main design consideration related with GUI's (Graphical User Interface) are listed below where these factors are considered in the system GUI designs.

- Attractive user interfaces
- User friendly interfaces, easy to use and easily learnable user interfaces
- Easy to navigate forward and backward and keeping the process flow of the actions
- Give good error messages with information to recover from the error occurred
- Prevent errors as much as possible and use client side validations to give immediate feedback
- Provide feedback of all the actions if succeeded or not

### 3.7.1 Welcome Page GUI

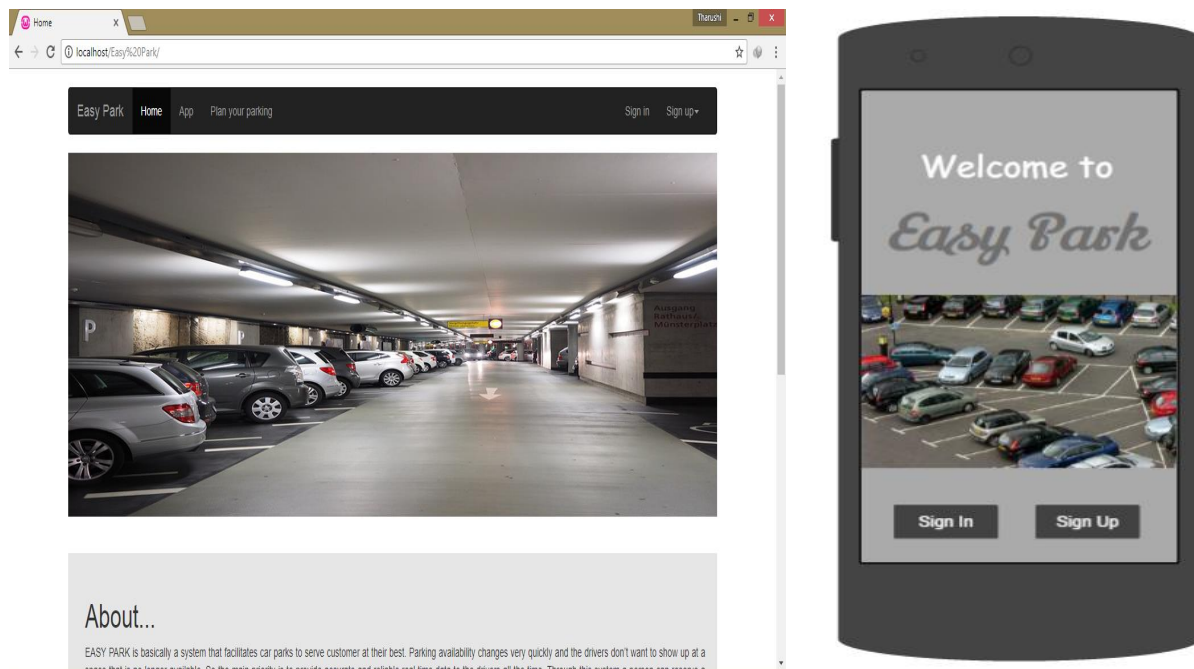


Figure 3.25 Welcome Page GUI

### 3.7.2 User login GUI

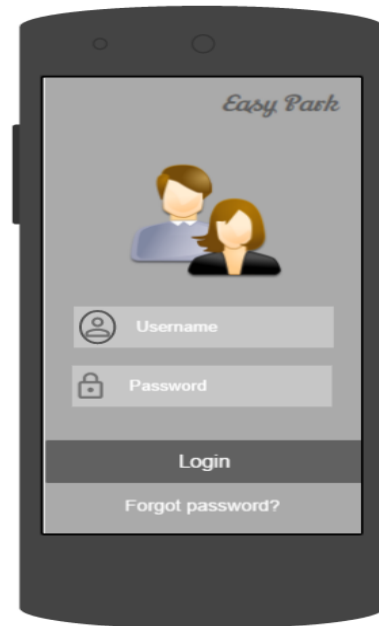
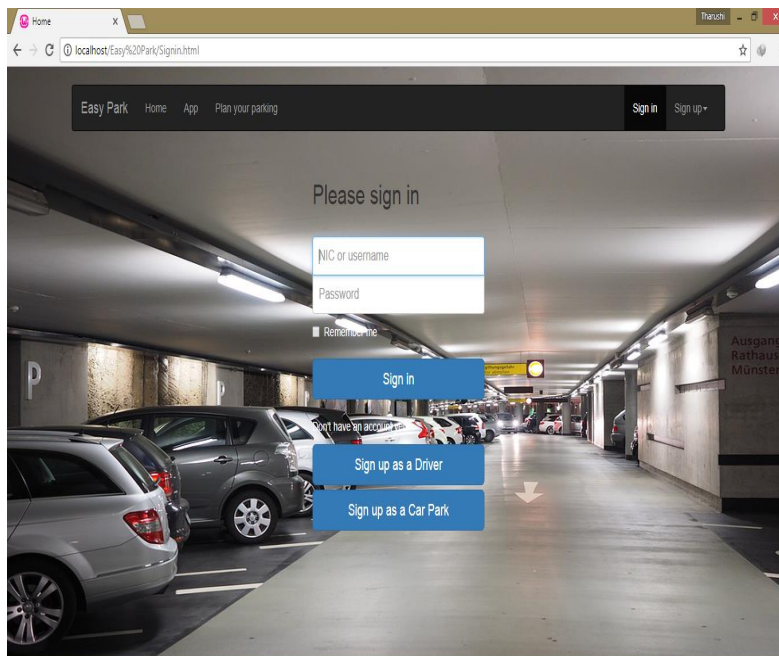


Figure 3.26 User login GUI

### 3.7.3 Create user GUI

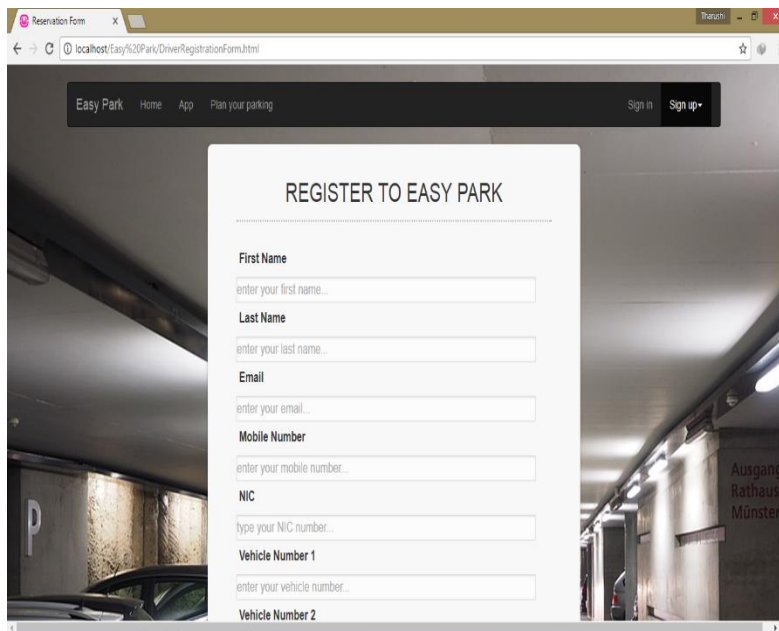


Figure 3.27 Create user GUI

### 3.7.4 Search car park GUI

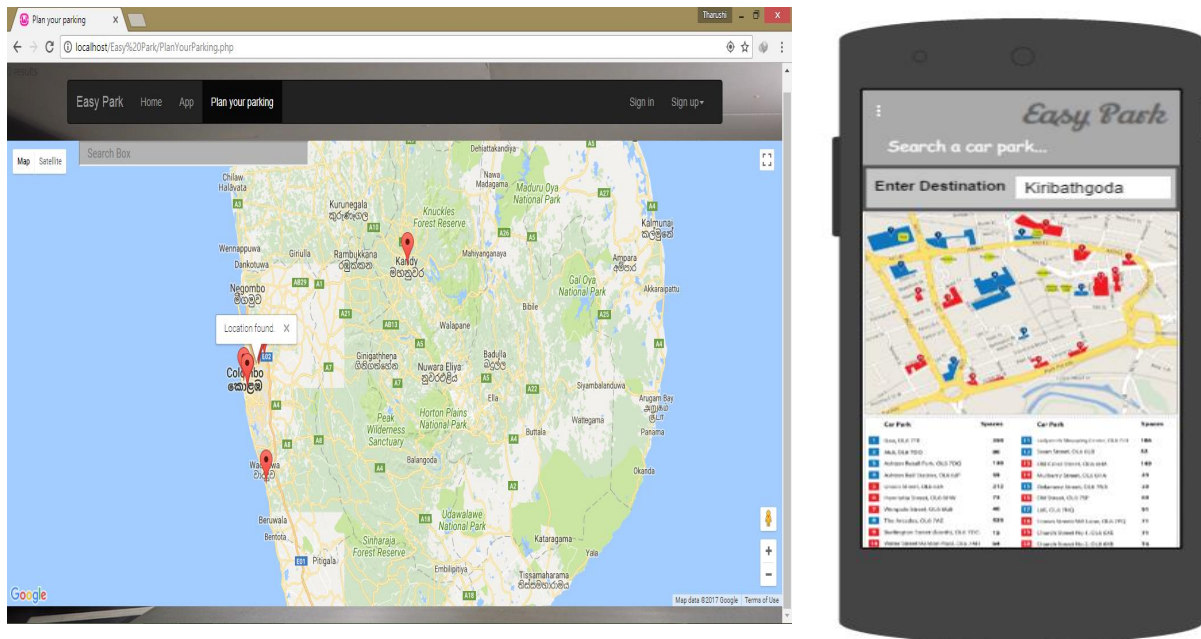


Figure 3.28 Search Car Park GUI

### 3.7.5 View car park GUI

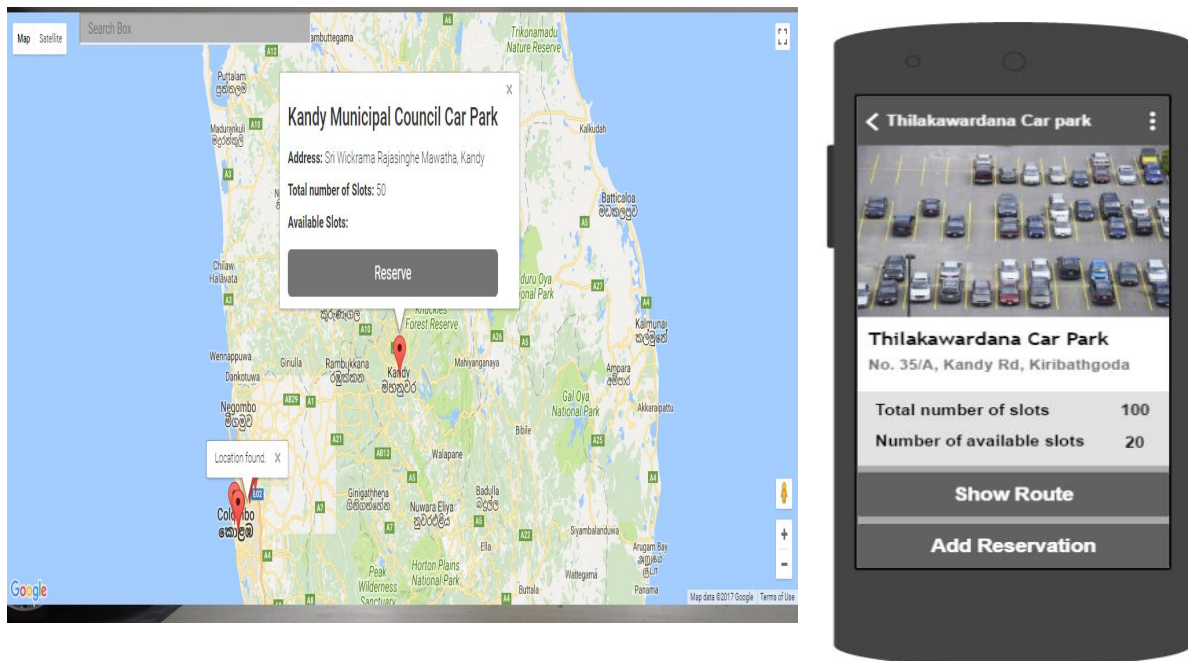


Figure 3.29 View Car Park GUI



### 3.7.6 Report

The figure 3.30 illustrates a report which represents the number of vehicles parked in the car parks of a particular city within a given week.

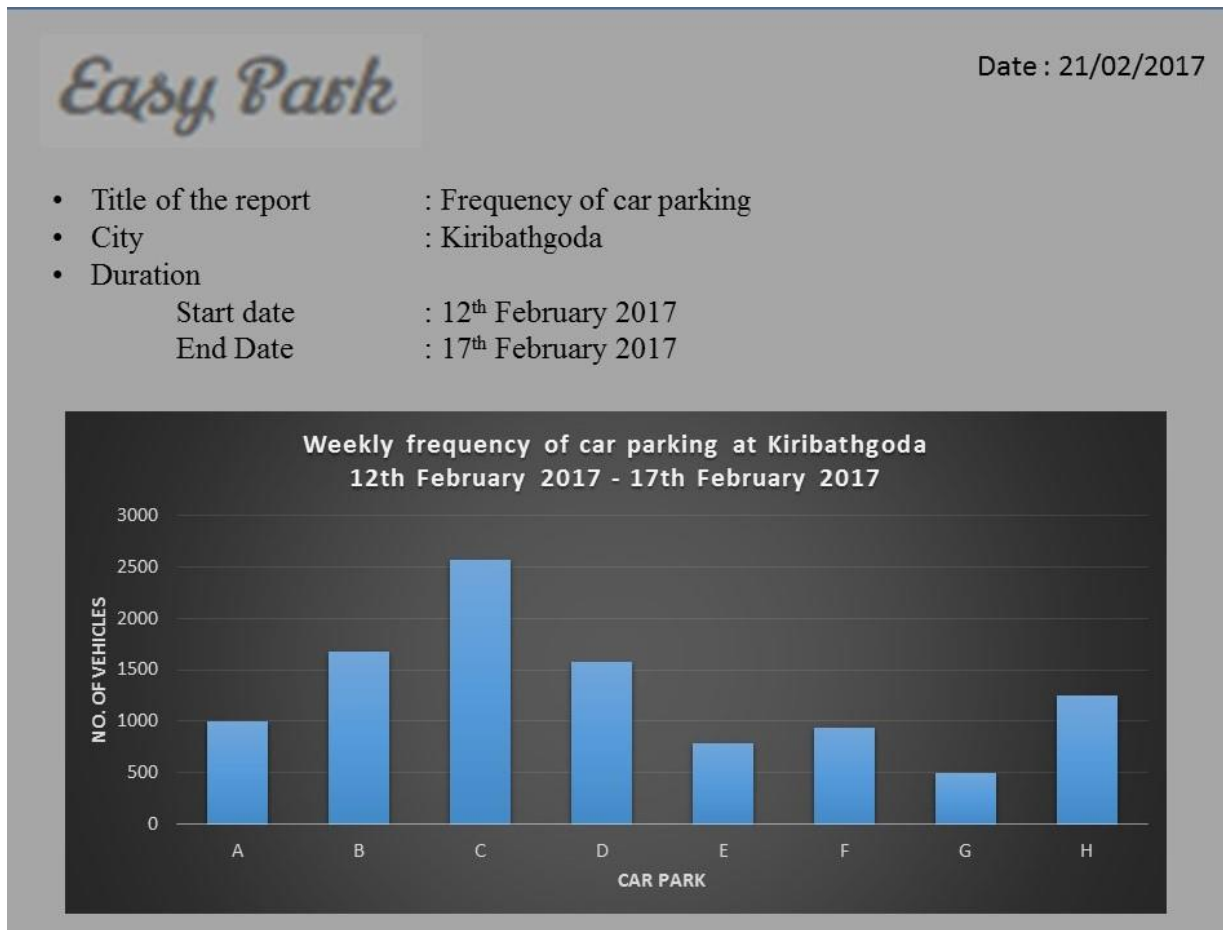


Figure 3.30 Report

### 3.7.7 Screen of the car park

The figure 3.31 illustrates the screen which represents the number of vehicles parked, available slots and the details of the parked vehicles in a particular car park as shown on the desktop at the car park.

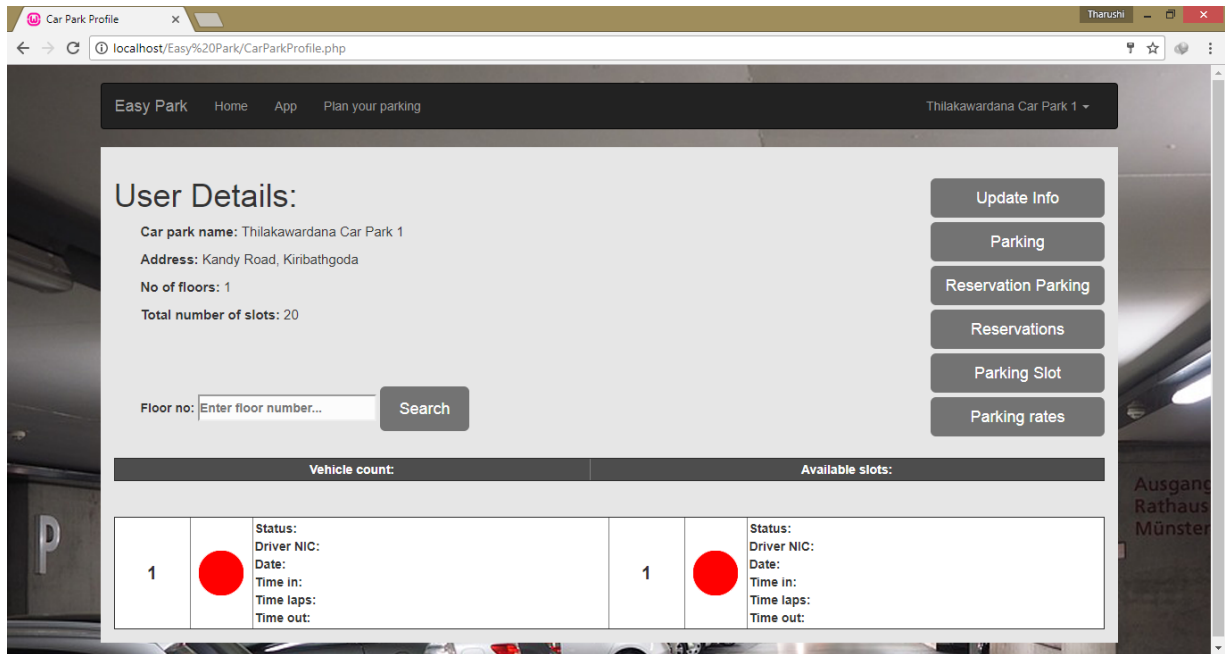


Figure 3.31 Screen of the car park

## 3.8 Summary

This chapter depicted the system design where it showed how the functionality is achieved. This was depicted through Objected Oriented Approach. Furthermore the Database Design and the Graphical User Interfaces were elaborated.

## CHAPTER 4

### 4. DEVELOPMENT

This chapter consist of the development details of the project. It gives a brief introduction to the programing language properties required to implement the Class Diagrams described in a previous chapter and select programming languages that have used in the Easy Park Car Park Management System development. And also it will give brief description of third party components/libraries used. Finally it is consisted of examples of algorithms used in the system.

## **4.1 Programming Languages and Development Tools**

When selecting the tools and technologies for the development of the Easy Park Car Park Management System, the main factors that have considered is the matching of those technologies with the hardware, software and human technological capabilities. In addition, support for attractive Graphical User Interface design was considered.

Easy Park Car Park Management System is a combination of web based application, a native android application with IoT included in the system.

In its web application it includes html view. Modules of the system capture user data and various database interactions and form submission parts are included in the system. GUI design should be attractive and support form filling controls for most of the GUIs. In the native mobile application, also the modules are interacting with database through web service.

Hence considering all these facts, for the development environment of web solution PHP scripting language has been used. Apart from this, JQuery, JavaScript, AJAX technologies have been used in the client side to give better flexibility, increase interactions with the user and minimize the response time of the system.

In the mobile application development, Java standard language for Android application development within the Android Studio IDE.

Since all these technologies are open source, could minimize the cost as well.

For the development of IoT sector of the system, Raspberry Pie 3 was used.

### **4.1.1 Programming Language**

#### **Web Solution**

##### **1) JavaScript**

A scripting language developed by Netscape to enable Web authors to design interactive sites. Although it shares many of the features and structures of the full Java language, it was developed independently. JavaScript can interact with HTML source code, enabling Web authors to spice up their sites with dynamic content. JavaScript is endorsed by a number of

software companies and is an open language that anyone can use without purchasing a license. Above mentioned features and functionalities are mainly affected to the selection of this language. Within web solution JavaScript used as client side validation scripting language.

## **2) PHP**

PHP stands for PHP: Hypertext Preprocessor. Hypertext refers to files linked together using hyperlinks, such as HTML (Hyper Text Markup Language) files. Preprocessing is executing instructions that modify the output. Taken directly from PHP's home, PHP.net, "PHP is an HTML-embedded scripting language. Web solution server side implementations are written on PHP scripting language.

Within web solution PHP used as server side validation scripting language.

## **3) CSS**

CSS is the abbreviation for Cascading Style Sheets, new feature being added to HTML that gives both Web site developers and users more control over how pages are displayed. By using CSS one could separate HTML content from its appearance, distinguishing style from structure. CSS give the following advantages.

- Control layout of many documents from one single style sheet
- More precise control of layout
- Apply different layout to different media-types (screen, print, etc.)

## **Mobile Solution**

### **1) Java**

Android applications are developed using the Java language. That is the only option for native applications. Java is a very popular programming language developed by Sun Microsystems (now owned by Oracle). Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks. Still, programming languages are only as powerful as their libraries. These libraries exist to help developers build applications.

Some of the Java's important core features are:

- It is easy to learn and understand
- It is designed to be platform-independent secure
- Use of virtual machines
- Object-oriented

Android relies heavily on these Java fundamentals. The Android SDK includes many standard Java libraries (data structure libraries, math libraries, graphics libraries, networking libraries and everything else you could want) as well as special Android libraries that will help you develop awesome Android applications.

#### **4.1.2 Development Tools and Technologies**

##### **Web Solution**

##### **1) Notepad++**

Notepad++ is a free source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GPL License.

##### **2) Wamp**

WAMPs are packages of independently created programs installed on computers that use a Microsoft Windows operating system.

WAMP is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as PhpMyAdmin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

### **3) MySQL**

MySQL is widely used open-source relational database management system (RDBMS) which was owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software.

### **4) JQuery**

This is cross browser Javascript library designed to simplify client side scripting of HTML web pages. JQuery is free, open source software, JQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. JQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developer create more dynamic web applications. JQuery is used to implement image gallery, and interactive web pages with toggle option, etc

### **5) AJAX**

AJAX is a latest web development technology which allows dynamic web page contents with scripting and data manipulation with JSON/XML objects. Asynchronous JavaScript and XML is a group of interrelated web development methods used on the client-side to create asynchronous web applications. With Ajax, this web application can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data is retrieved using the *HttpRequest* object using JSON (JavaScript Object Notation).

Reference: [www.ajax.com](http://www.ajax.com)

## **6) Apache**

Apache is web server software notable for playing a key role in the initial growth of the World Wide Web. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, and PHP.

## **Mobile Solution**

### **1) Android Studio 2.3.2**

Android Studio is the official integrated development environment(IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It provide the interface for you to create your apps and to handle much of the complicated file-management behind the scenes. The programming language used is Java. Android Studio will give you access to the Android SDK or 'Software Development Kit'. Think of this as an extension to the Java code that allows it to run smoothly on Android devices and take advantage of the native hardware. Java is needed to write the programs, the Android SDK is needed to make those programs run on Android and Android Studio has the job of putting it all together. At the same time, Android Studio also enables you to run your code, either through an emulator or through a piece of hardware connected to your machine. You'll then also be able to 'debug' the program as it runs and get feedback explaining crashes etc. so that you can more quickly solve the problem.

## **4.2 Third Party Components and Libraries**

### **1) Raspberry Pi 3 Model B**

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. Several generations of Raspberry Pis have been released. The first generation (Raspberry Pi 1 Model B) was released in February 2012. Raspberry Pi 3 Model B released in February 2016 and is bundled with on-board WiFi, Bluetooth and USB boot capabilities.



The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B.

Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processor, 10x faster than the first generation Raspberry Pi.

Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.

## **2) Google Map API**

The Google Maps API allow for the embedding of Google Maps onto web pages of outside developers, using a simple JavaScript interface or a Flash interface. It is designed to work on both mobile devices as well as traditional desktop browser applications. The API includes language localization for over 50 languages, region localization and geocoding, and has mechanisms for enterprise developers who want to utilize the Google Maps API within an intranet. The API HTTP services can be accessed over a secure (HTTPS) connection by Google Maps API Premier customers.

## **4.3 Algorithms used in the System**

### **1) Calculating the duration of the parking**

```
$sql = "SELECT * FROM payment WHERE VehicleNum='".$user2->vnum.'" AND  
CheckOutStatus = '0'";
```

```
$result = mysqli_query($connection, $sql);
```

```
if (mysqli_num_rows($result)>0){
```

```
    //output data of each row
```

```
    while($row = mysqli_fetch_assoc($result)){
```

```
        //checkin date and time
```

```
        $inDate = $row['CheckInDate'];
```

```
        $inTime = $row['CheckInTime'];
```

```
        $strDT = $inDate." ".$inTime;
```

```

        $temp1 = strtotime($strDT);
        $ckInDateTime = date("Y-d-m H:i:s", $temp1);
    }
} else {
    echo "0 results";
}

//Checkout date and time

$outTime = strtotime($user2->checkouttime);
$ckOutDateTime = date("Y-m-d H:i:s", $outTime);

$ckin = new DateTime($ckInDateTime);
$ckout = new DateTime($ckOutDateTime);
$interval = date_diff($ckin,$ckout);

$hour1 = 0; $hour2 = 0; $hour3 = 0;

if($interval->format('%D') > 0){
    $hour1 = $interval->format('%a')*24;
}

if($interval->format('%h') > 0){
    $hour2 = $interval->format('%h');
}

if($interval->format('%i') > 0){
    $hour3 = 1;
}

$duration = $hour1 + $hour2 + $hour3;

```

## 4.4 Hashing of the Confidential Data

When storing confidential and private data of a user, it should be made sure that a third party would not be able to retrieve those. For example a password should not be stored in a database as a plain text.

## **4.5 Summary**

In this chapter, the development aspects of the system has been described. Under that initially the technologies used have been explained and justified and finally the constraints faced during the development process have been described.

## CHAPTER 5

### 5. TESTING

Software testing aspect of the “Easy Park Car Park Management System” is discussed in this chapter. Under this chapter, the strategies and types of testing used and why and how they have been employed to test the system have been discussed. Then it will discuss about the sample test cases, followed by the test reports of the system and results illustrating the severity of the bugs identified and possible solutions.

## **5.1 Test Plan and Test Strategy**

Testing is done in order to align the product with the user. The functionalities and functions are tested in order to improve the quality of the product and improve the functionalities of the product.

The testing plan defines the items to be tested and the functions which are selected based on the importance of the functions, and the risk of the functions on the user's view point.

Then the test cases were designed corresponding with the use case descriptions. They were executed manually and the results were recorded. The bugs identified were corrected and tested again. In this testing process of this system both black box and white box testing is being used.

### **5.1.1 White Box Testing**

The testing is mostly carried out while the coding is taking place. Each statement, conditions and decision structures test soon after finishing the coded. The purpose of using white box testing for this system is that white box testing enables to identify small bugs that could be created by the mistakes of coding and due to the errors in the decision logics, and it enables the developers to prevent those errors before integrating the recently developed contents together with the system.

### **5.1.2 Black Box Testing**

This has been used once after a particular functionality is completely developed. A set of random values used for the testing and the errors and bugs which are generated is identified and fixed. The gessoes over user errors can be tested at this stage.

Under these testing approaches four main test types has used to test this system.

- **Concept Testing**

This is used to check the compatibility of the concepts used in the system and the concepts in the actual business process.

- **Unit Testing**

This type of testing has used white box testing approach to test each and every critical units (i.e. functions, algorithms etc.) built in the system individually. Only after the following tests are being completed, units are integrated with each other and tested in integration testing.

- **Integration Testing**

After integrating the components together, which are tested using the unit testing, the entire integrated component is tested again. Both black box and white box testing approaches have used for this testing type.

- **Interface Testing**

Interface testing is being used to test each and every component in the GUIs against their intended purpose.

## 5.2 Testing Environment

Operating System: Windows 8.1

## 5.3 Sample Test Cases

Test cases for main system:

**Table 5.1 User Login Test Case**

1. User Login				
ID	Test Case Description	Input data	Expected Output	Status
1.1	User Login Steps: 1. Input user name 2. Input password 3. Press signin	Username Password	Accept the username and password, and check the type of the user profile (whether it is admin username or driver username or car park username). Display the user home page.	Pass

1.2	Validate user login Steps: 1. Input username 2. Input password 3. Press signin button	Incorrect username, Incorrect password	Display error message and clear username and password fields.	Pass
-----	---	---	---	------

**Table 5.2 Add User Test Case**

2. Add driver or admin				
ID	Test Case Description	Input data	Expected Output	Status
2.1	Add users Steps: 1. Input NIC 2. Input other requested personal details 3. Input username 4. Input password 5. Press Register button	NIC Requested personal details Username Password	Display the user home page	Pass
2.2	Validate user entry	Already existing username and password	Display error message	Pass

**Table 5.3 Add Car Park Test Case**

3. Add car park				
ID	Test Case Description	Input data	Expected Output	Status
3.1	Add car parks Steps: 1. Input other requested details 2. Input username 3. Input password 4. Press Register button	Requested details Username Password	Automatically generate the car park ID Display the car park home page	Pass
3.2	Validate user entry	Already existing username and password	Display error message	Pass

**Table 5.4 Parking Space Reservation by Driver Test Case**

4. Reserve parking space by the driver				
ID	Test Case Description	Input data	Expected Output	Status
4.1	Select a car park Steps: 1. Select a car park 2. Click on the car park 3. Click “Reserve” button	Click on the car park Click on the reserve button	When the driver click on the car park an Info window will be displayed showing the public information of the car park. When the driver click on the “Reserve” button, the reservation form will be displayed. Display the driver home page.	Pass



4.2	Reserve a parking space Steps: 1. Input the data requested 2. Click “Reserve” button	Requested details	Display the driver’s home page Display the new reservation under the Reservation details in the driver’s home page.	
4.3	Validate user entry	Invalid user entry Blank text boxes	Display error message highlighting the empty field label.	Pass

**Table 5.5 Parking Space Reservation by the Car Park Test Cases**

5. Reserve parking space by the car park				
ID	Test Case Description	Input data	Expected Output	Status
5.1	Add Reservation Steps: 1. Input the data requested 2. Click on the “reserve” button	Requested details	Display the car park home page.	Pass
5.2	Validate user entry	Invalid user entry Blank text boxes	Display error message highlighting the empty field label.	Pass

**Table 5.6 Parking Process – Checkin Test Case**

6. Parking Process - Checkin				
ID	Test Case Description	Input data	Expected Output	Status
6.1	Add Checkin Steps: 1. Input vehicle number 2. Select Checkin 3. Click “OK”	Vehicle number	Display the Checkin Interface. The checkin time, date, driver NIC, vehicle number will be automatically displayed in the relevant text boxes.	Pass
6.2	Submit Checkin Steps: 1. Select an available slot 2. Click “ Submit” button	Slot ID	Display the home page of the car park.	Pass

**Table 5.7 Parking Process – Checkout Test Case**

7. Parking Process - Checkout				
ID	Test Case Description	Input data	Expected Output	Status
7.1	Add Checkout Steps: 1. Input vehicle number 2. Select Checkout 3. Click “OK”	Vehicle number	Display the Checkout Interface. The Checkin information, checkout time and date will be displayed in the relevant text boxes. The duration of the parking and the amount payable will be automatically calculated and displayed.	Pass
7.2	Submit Checkout Steps: 1. Click “Submit” button		Display the home page of the car park. The amount payable will be automatically deducted from the driver’s account.	Pass

**Table 5.8 Generate Reports Test Case**

8. Generate reports				
ID	Test Case Description	Input data	Expected Output	Status
8.1	Open generate report interface Steps: 1. Login as the admin 2. Click on “Generate Reports” button	Username password	When signin, display the admin home page. After clicking the generate reports button, the report interface will be displayed.	Pass
8.2	Generate reports	Select the required parameters	Display the report in the form of a chart.	Pass

## 5.4 Test Report

Testing is done as part of the development process, and not at the end of the whole process. If it is not done throughout the development process then the testing is done at the end. If so the system will take many changes even in the interface design and process of interface navigations too. Thus changing these at the end is costly and less effective.

The number of defects that is uncovered by the test cases and test data decrease gradually with the number of testing iterations performed. In the initial iteration large number of bugs uncovered and most of those errors are moderate interims of the importance.

Unit testing, integrated testing, functional testing was done by the developer with the help of peers. Peer evaluation aid in the achieving test goals and achieving functional requirements of the system greatly since they will comment on the usability of the system which will be a help in improving the usability of the system. UI testing and the Concept testing was done in the same environment with involvement of peers at particular times.

Peers: Miss Gaya Madhumali, Mr. Rajitha Abesekara, Undergraduates of MIT

- Concept Testing: Passed
- All Units Testing: Passed
- All integrated Testing: Passed
- UI Testing: Average
- All Functional Testing: Partially Completed
- System Testing: Partially Completed

System Completion:

1. Parking Process – Fully implemented  
Result: Completed
2. Reservation Process – Fully implemented  
Result: Completed
3. IoT – Fully implemented  
Result: Completed

Concept Testing, Unit Testing, Integrated testing, Functional testing test the functionality of the system and compares with the requirements of the system which increases the reliability and quality of the system. UI testing increases the usability of the system, which increases the quality of the system as well. Errors were found in Unit, Integrated and Functional testing. But they were not catastrophic or serious errors and were tolerable. Developer was able to handle the exceptions caught when debugging the system. Errors found in UI and Concept testing were taken into consideration and re-modified the system according to them in the design phase and development phase respectively.

## **5.5 Summary**

In this chapter, the developer has mentioned the test types, which the developer has followed when implementing the test procedure and test cases of the system. The developer has further described test strategies used, test plan, sample test cases and presents the report of the overall test phase.

# CHAPTER 6

## 6. IMPLEMENTATION

This chapter consists of the implementation details of the project. It will demonstrate the implementation requirements of the Easy park Car Park management System. In addition, this chapter mainly focuses on the implementation specifications such as installation guide and security procedures.

## **6.1 System Requirements**

The minimum hardware and software requirement for the installation of the system as follows.

### **6.1.1 Hardware Requirements**

Minimum hardware requirements for the system installation.

- CPU – Pentium 4 2.8 GHz
- RAM – 4GB or above
- Router with firewall to be always connected to the Internet
- LDR Sensors
- Raspberry Pie 3

### **6.1.2 Software Requirements**

Operating System

- Windows 8.1, Android

Antivirus Software

- Norton, McAfee, Avira, Kaperskey

Backend Software

- Wamp Server

## **6.2 User Manual**

This user manual will guide you through the system functionality enabling the user, easily manage activities and processes of the main system. This software is created by M. T. S. Ilangakoon, with special customized features to satisfy the requirements identified in the normal car parking process. In addition to the administrator, mainly two user categories are authorized to access the system. They are,

- 1). Driver
- 2). Car Park

These users have the authority to access the system according to their user type.

## 6.2.1 Home Page of the System

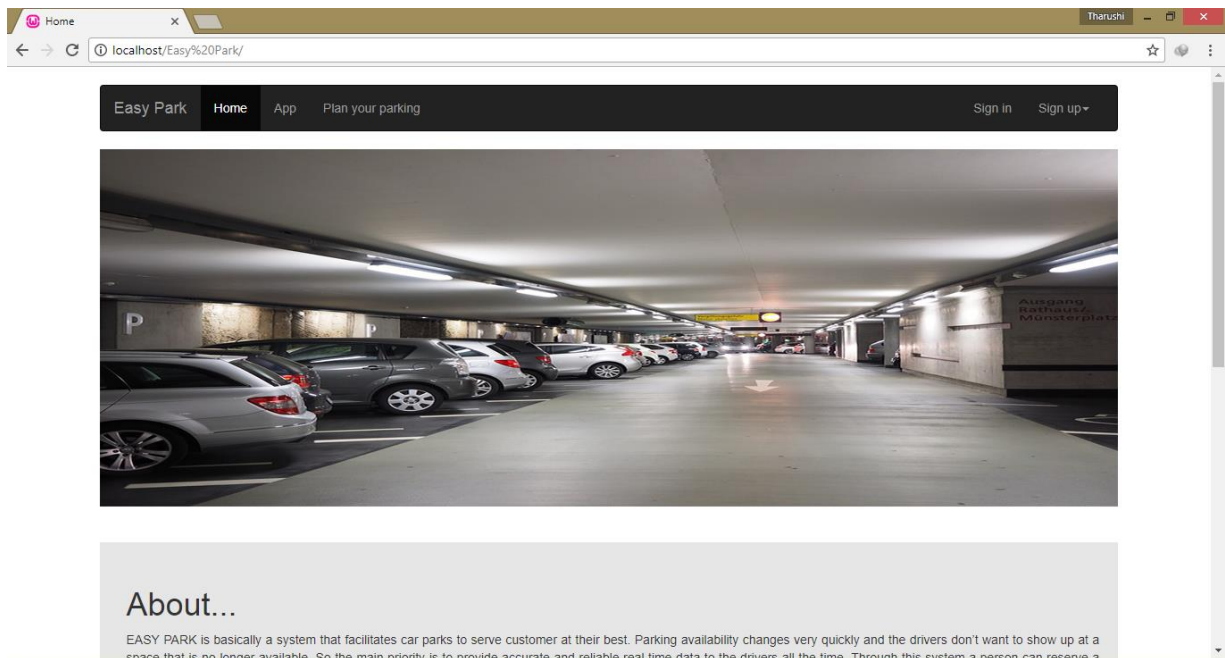


Figure 6.1 Home Page

The main interface contains the information about the system and its benefits.

## 6.2.2 Plan your Parking Interface

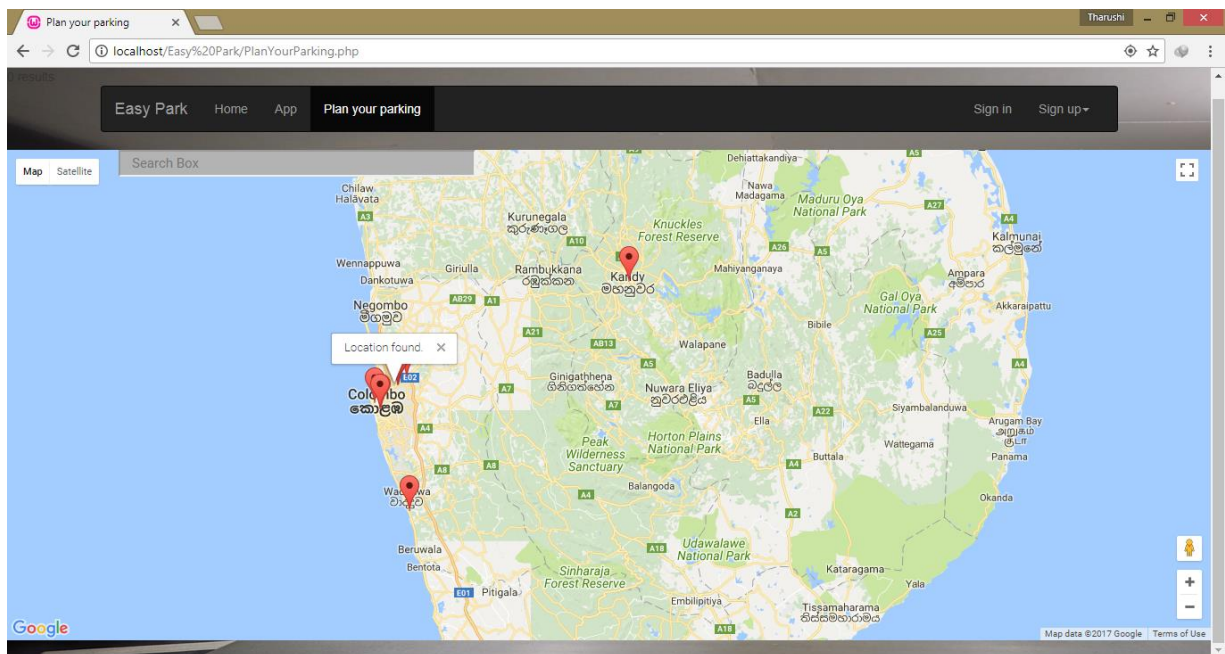
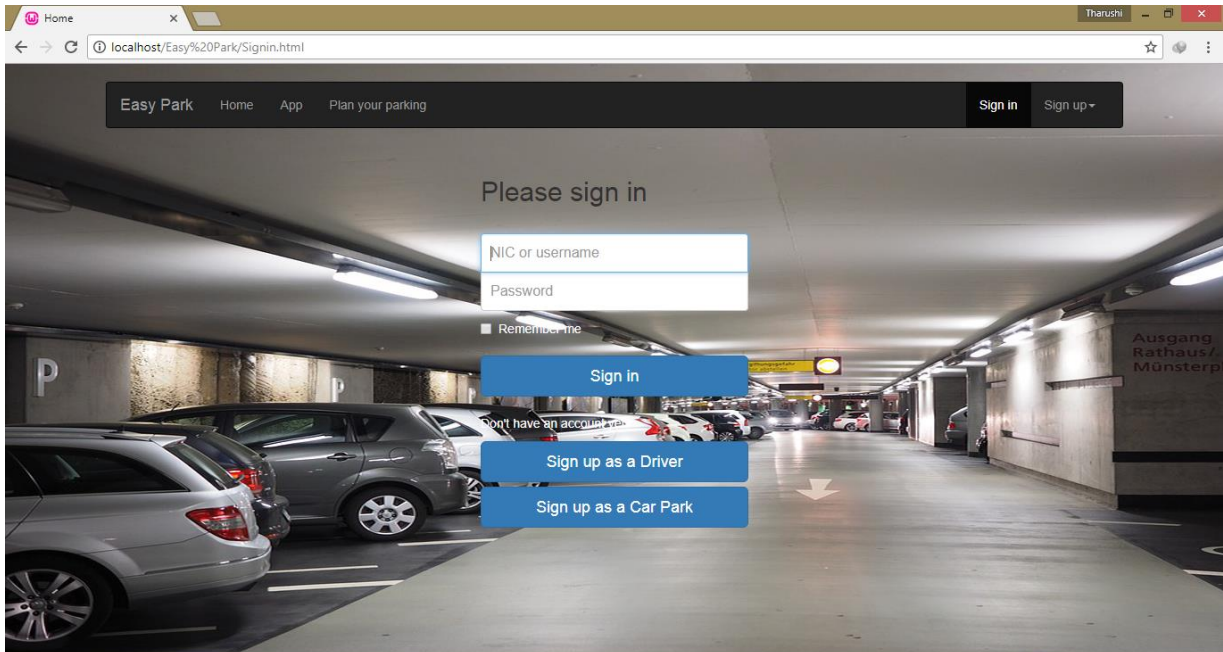


Figure 6.2 Plan Your Parking Interface

This interface shows the map of the car parks registered in the system. You can view the available parking slots in a particular car park. But first you have to signin to reserve a parking slot.

### 6.2.3 Signin Interface



**Figure 6.3 Signin Interface**

Steps:

1. Enter the username/NIC
2. Enter the password
3. Click Signin button

After inserting the correct data, the user will be directed to the home page of their profile.



## 6.2.4 Home Page of the user profile

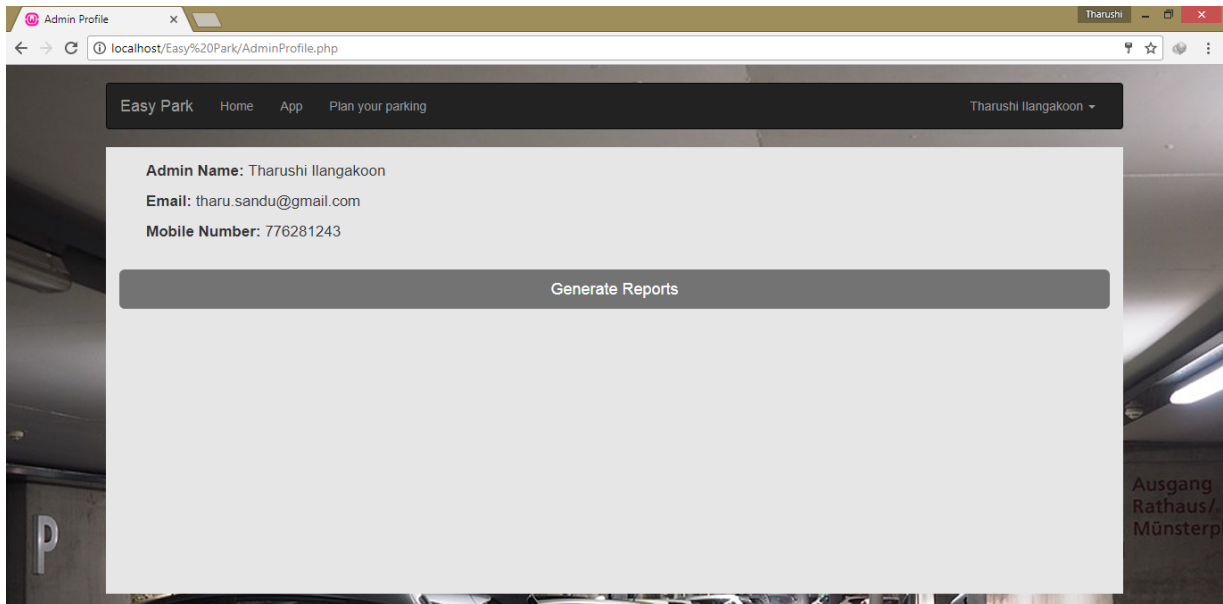


Figure 6.4 Admin User Profile Interface

This holds buttons to direct the administrator to generate the reports.

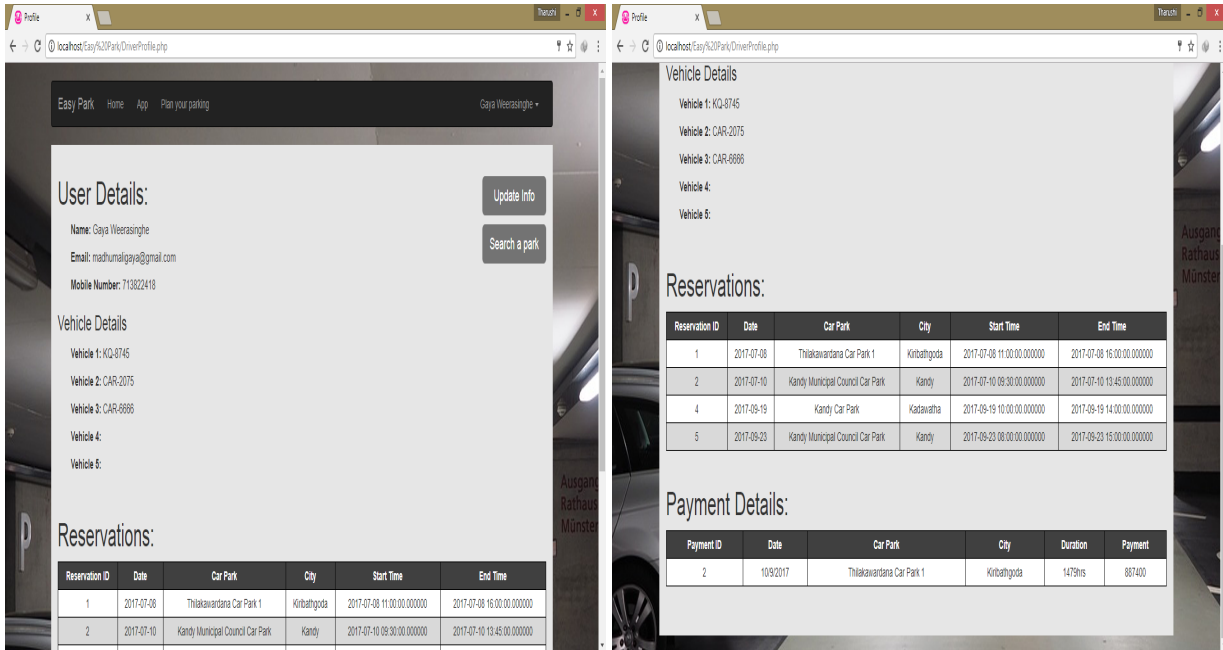


Figure 6.5 Driver User Profile Interface

This interface displays the personal information of the driver, reservation information in different car parks, and their payment details for the parkings they have done. The driver can update his/her information selecting the “Update Info” Button.

The screenshot shows a web browser window with the URL `localhost/Easy%20Park/CarParkProfile.php`. The page has a navigation bar with links: Easy Park, Home, App, Plan your parking. The current page is titled "Thilakawardana Car Park 1".

**User Details:**

- Car park name: Thilakawardana Car Park 1
- Address: Kandy Road, Kiribathgoda
- No of floors: 1
- Total number of slots: 20

On the right side, there are buttons: Update Info, Parking, Reservation Parking, Reservations, Parking Slot, and Parking rates.

Below the user details, there is a search section:
   
Floor no:

Below the search section, there are two columns: "Vehicle count:" and "Available slots:". Each column contains a table with two rows. The first row in each table has a red circle icon and the number "1". The second row in each table has a red circle icon and the number "1".

Vehicle count:		Available slots:	
1		1	
Status:	Driver NIC:	Status:	Driver NIC:
Date:	Time in:	Date:	Time in:
Time laps:	Time out:	Time laps:	Time out:

**Figure 6.6 Car park Profile Interface**

This interface displays the details of the car park and tools to help the car park attendant to manage the vehicle parking process. A parking attendant can,

1. Update car park information
2. Manage normal check in and check out process of vehicles
3. Manage check in and check out of the reservations
4. Search reservation information, add/update/delete reservations
5. Add/Update parking rates
6. Add/Update/Delete parking slots

## 6.2.5 User Signup

The following button can be viewed in the menu bar of all the interfaces

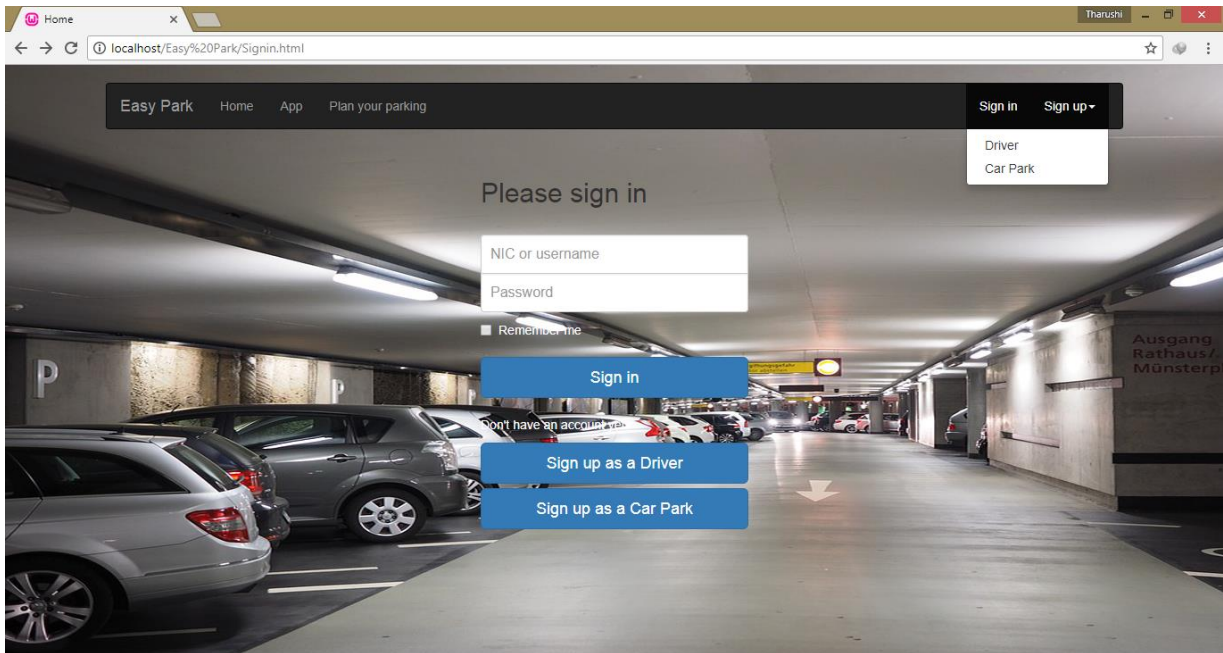


Figure 6.7 Select User Type

Select the required user type to sign up in to the system.

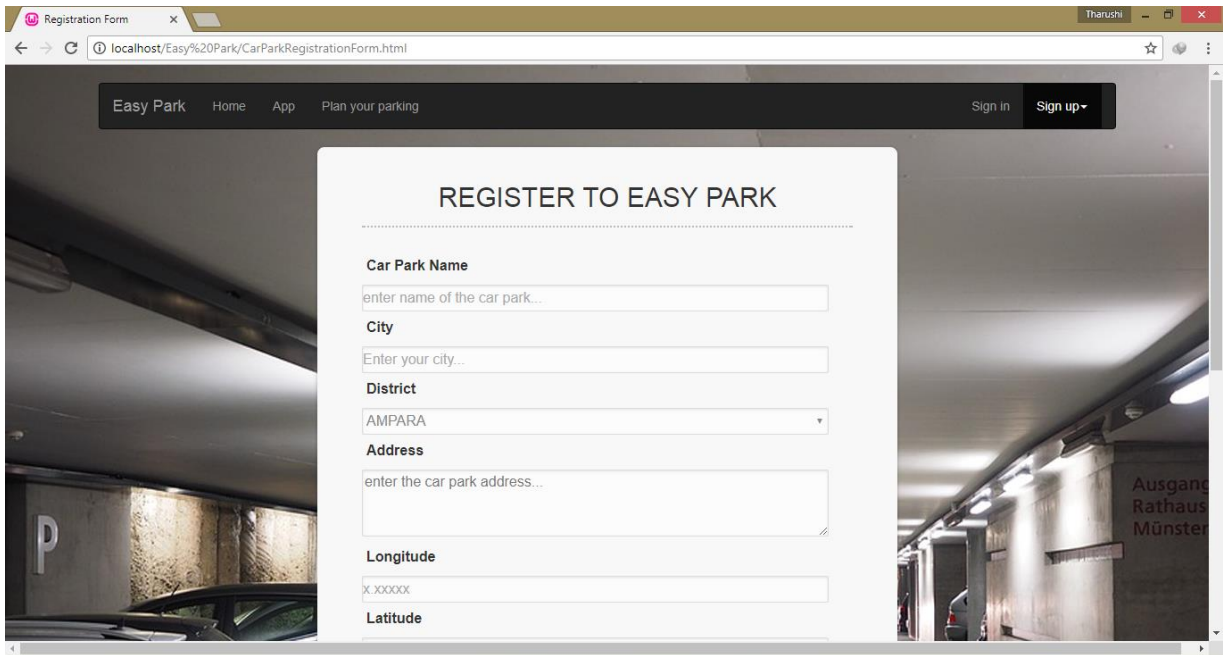
A screenshot of a web browser displaying the 'Easy Park DriverRegistrationForm.html' page. The page features a dark navigation bar at the top with links for 'Easy Park', 'Home', 'App', and 'Plan your parking'. On the right side of the navigation bar, there are 'Sign in' and 'Sign up' buttons. The main content area has a background image of a parking garage. It contains a white registration form titled 'REGISTER TO EASY PARK'. The form has the following fields: 'First Name' (with placeholder 'enter your first name...'), 'Last Name' (with placeholder 'enter your last name...'), 'Email' (with placeholder 'enter your email...'), 'Mobile Number' (with placeholder 'enter your mobile number...'), 'NIC' (with placeholder 'type your NIC number...'), 'Vehicle Number 1' (with placeholder 'enter your vehicle number...'), and 'Vehicle Number 2'.

Figure 6.8 Driver Registration Form

To add a new driver to the system, the respective fields must be filled with the following data.

NIC, First Name, Last Name, Email, Mobile Number, Vehicle numbers (Up to 5 vehicles), Username, Password

A driver can use his NIC as the Username and the given password to login to the system.

The image shows a web browser window displaying a registration form titled "REGISTER TO EASY PARK". The browser's address bar shows the URL "localhost/Easy%20Park/CarParkRegistrationForm.html". The form is centered on a background image of a parking garage. The form fields include: "Car Park Name" (text input), "City" (text input), "District" (dropdown menu with "AMPARA" selected), "Address" (text input), "Longitude" (text input with placeholder "X.XXXXX"), and "Latitude" (text input). The browser window has a dark header with navigation links "Easy Park", "Home", "App", and "Plan your parking", along with "Sign in" and "Sign up" buttons.

**Figure 6.9 Car Park registration Form**

To add a new car park to the system, the respective fields must be filled with the following data.

Car Park Name, City, District, Longitude, Latitude, Number of Floors, Number of parking spaces in total, Type of the car park (outdoor, indoor, automates etc.), Username, Password

The car park attendant can use the given Username and password to login to the system.

The Car Park ID will be generated automatically by the system.

### **6.2.6 Reserving a Parking Space by the Driver**

When user click on the “Search Car Park” button in the home page of his profile or the “Plan your Parking” tab in the menu bar, the driver will be directed to the google map which shows all the registered car parks in the system.

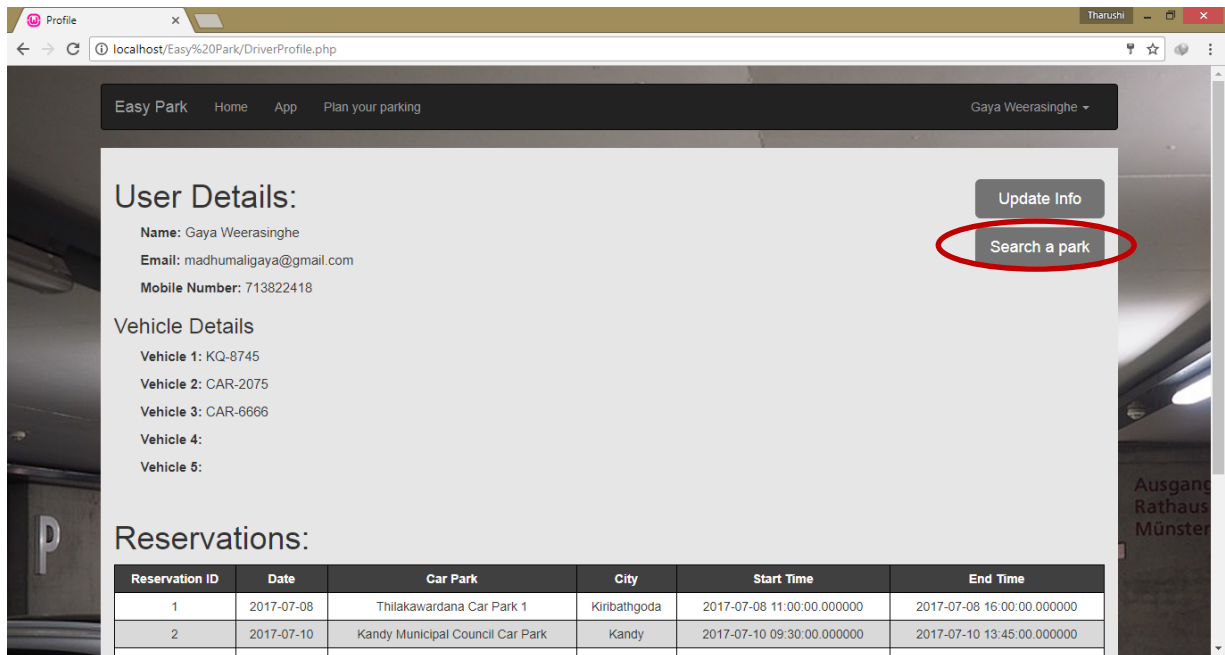


Figure 6.10 Viewing the google map

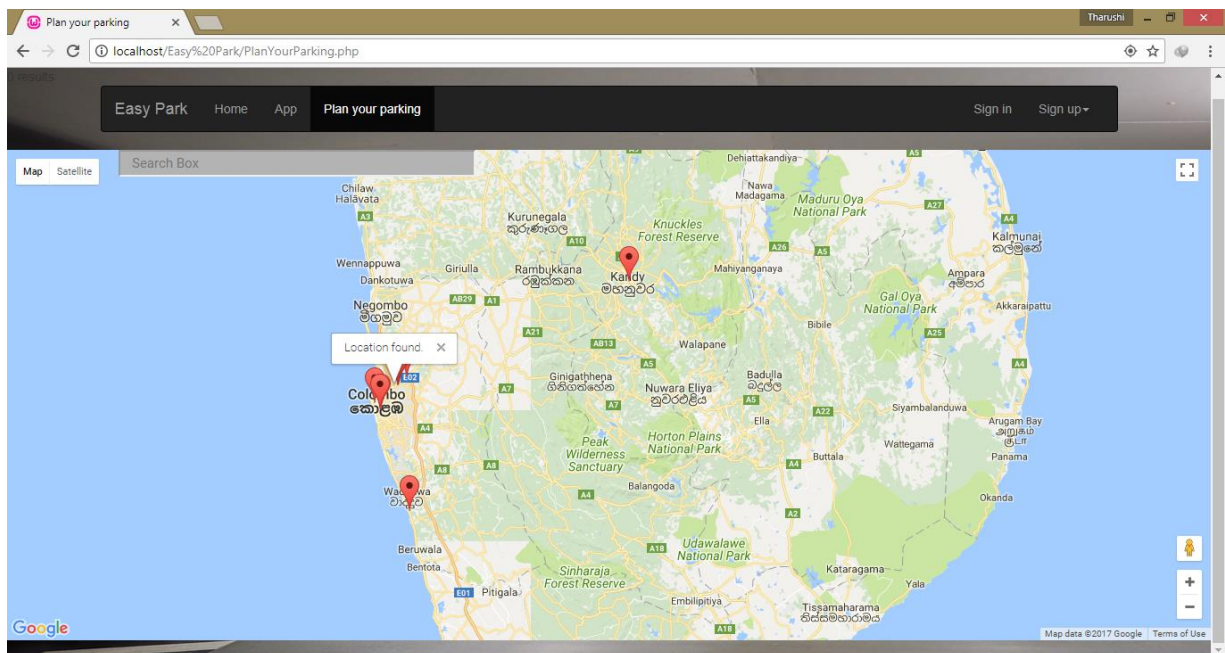
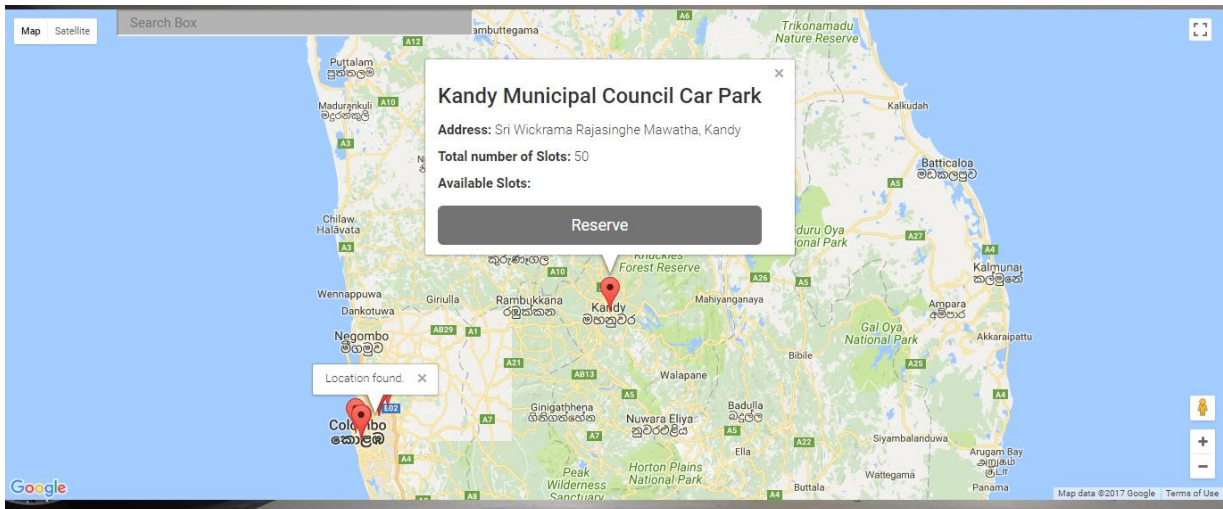


Figure 6.11 Map with Car Parks

When you click on a marker of the car park, it will display an info window which consist of public information of a car park.





**Figure 6.12 Info Window**

When you click on the “Reserve” button, the driver will be directed to the Reservation form

## RESERVE YOUR PARKING SPACE

---

**Vehicle Number**

**Reservation Date**

**Start time**

**End Time**

RESERVE

RESET

CANCEL

\*Your reservation begins from the moment you submit the reservation form.

\*If you don't arrive at your reserved parking space within 15 minutes or you didn't cancel your reservation, the normal parking charges will be charged for the reserved time duration.

**Figure 6.13 Parking Reservation Form viewed by the driver**

To add a new reservation, the respective fields must be filled by the driver with the following data.

Reservation date, Start Time, End Time.

The reservation date and the start time date given in the form should be the same.

The reservation ID will be automatically generated by the system. The driver can view his reservation details with the Reservation ID in his profile.

The Reservation ID given by the system need to be introduced when check in to the car park.

**Vehicle Details**

Vehicle 1: KQ-8745  
Vehicle 2: CAR-2075  
Vehicle 3: CAR-6666  
Vehicle 4:  
Vehicle 5:

**Reservations:**

Reservation ID	Date	Car Park	City	Start Time	End Time
1	2017-07-08	Thilakawardana Car Park 1	Kiribathgoda	2017-07-08 11:00:00.000000	2017-07-08 16:00:00.000000
2	2017-07-10	Kandy Municipal Council Car Park	Kandy	2017-07-10 09:30:00.000000	2017-07-10 13:45:00.000000
4	2017-09-19	Kandy Car Park	Kadawatha	2017-09-19 10:00:00.000000	2017-09-19 14:00:00.000000
5	2017-09-23	Kandy Municipal Council Car Park	Kandy	2017-09-23 08:00:00.000000	2017-09-23 15:00:00.000000

**Payment Details:**

Payment ID	Date	Car Park	City	Duration	Payment
2	10/9/2017	Thilakawardana Car Park 1	Kiribathgoda	1479hrs	887400

**Figure 6.14 Reservation Details**

### 6.2.7 Reserving a Parking Space by the Car Park

The car park can also add reservations by clicking the “Reservations” button and then select the “Add Reservation” from the given choices.

The screenshot shows a web interface for managing car park details. The main form, titled 'User Details:', contains the following information:

- Car park name: Thilakawardana Car Park 1
- Address: Kandy Road, Kiribathgoda
- No of floors: 1
- Total number of slots: 20

On the right side of the form, there are four buttons: 'Update Info', 'Parking', 'Reservation Parking', and 'Reservations'. The 'Reservations' button is highlighted.

A modal window titled 'Reservation Details:' is open in the center. It contains four radio buttons for selection:

- ☒ Search Reservation
- ☐ Add Reservation
- ☐ Update Reservation
- ☐ Delete Reservation

At the bottom of the modal are 'OK' and 'Cancel' buttons.

Below the modal, a table displays reservation data. The table has two columns, each containing a red circle icon and a list of fields: Status, Driver NIC, Date, Time in, Time laps, and Time out.

1	<div>Status:</div> <div>Driver NIC:</div> <div>Date:</div> <div>Time in:</div> <div>Time laps:</div> <div>Time out:</div>	1	<div>Status:</div> <div>Driver NIC:</div> <div>Date:</div> <div>Time in:</div> <div>Time laps:</div> <div>Time out:</div>
---	---	---	---

Figure 6.15 Adding a reservation by the car park

A new reservation can be added by filling the respective fields. The new reservation can be viewed in the driver profile.



**RESERVE YOUR PARKING SPACE**

---

**Driver NIC**

Enter NIC number..

**Vehicle Number**

Enter your Vehicle number

**Reservation Date**

22-Sep-2017

**Start time**

11:00:01

**End Time**

dd-----yyyy --:--

**RESERVE** **RESET** **CANCEL**

Figure 6.16 Parking Reservation form viewed by the car park

### 6.2.8 Parking Process in the Car Park

- **Check in**

When a vehicle come to the car park, the car park attendant has to follow the following process.

1. Click on the “Parking” button
2. Enter the vehicle number
3. Select “Checkin”
4. Click OK
5. The check in date and time will be displayed
6. Select a parking slot
7. Click Submit

## User Details:

Car park name: Thilakawardana Car Park 1

Address: Kandy Road, Kiribathgoda

No of floors: 1

Total number of slots: 20

Update Info

Parking

Reservation Parking

Reservations

### Parking Details:

Vehicle Number:

☒ Check in
 ☐ Check out

OK Cancel

1		Driver NIC: Date: Time in: Time laps: Time out:	1		Driver NIC: Date: Time in: Time laps: Time out:
---	--	---	---	--	---

**Figure 6.17 Insert Checkin Details**

Check In

localhost/Easy%20Park/ParkingCheckin.php?vnum=KW-5000%20%20%20day=22&%20month=9&%20year=2017&%20hour=16&%20minutes=21&%20seconds=14

Therushi

Easy Park Home App Plan your parking

Thilakawardana Car Park 1

### Parking Details:

Vehicle Number:

Driver NIC:

Check in Date:

Check in Time:

Parking Slot:

SUBMIT

CANCEL

**Figure 6.18 Checkin Details**

- **Check out**

When a vehicle check out from the car park, the parking attendant has to follow the following steps.

1. Click on the “Parking” button
2. Enter the vehicle number
3. Select “Checkout”
4. Click OK
5. The checkout date, time, duration of the parking and the parking charges will be displayed
6. Click Submit

**User Details:**

Car park name: Thilakawardana Car Park 1  
Address: Kandy Road, Kiribathgoda  
No of floors: 1  
Total number of slots: 20

**Parking Details:**

Vehicle Number: KW-5000

☐ Check in ☒ Check out

OK Cancel

1		Date:		1		Date:	
		Time in:				Time in:	
		Time laps:				Time laps:	
		Time out:				Time out:	

**Figure 6.19 Insert Checkout details**

Easy Park Home App Plan your parking Thilakawardana Car Park 1

### Check Out Details:

Vehicle Number: KQ-6047

Driver NIC: 948401932v

Driver's Name: Tharushi Ilangakoon

Check in Date: 22/9/2017

Check in Time: 16:55:3

Parking Slot: Thilaka1-2

Check out Date: 22/9/2017

Check out Time: 19:48:53

Duration: 418364hrs

Payment Amount (Rs.): 251018400

SUBMIT

CANCEL

**Figure 6.20 Checkout Details**

The relevant payment will be reduced from the user subscription and it will be displayed in the driver profile under payment details.

Profile

localhost/Easy%20Park/DriverProfile.php

### Vehicle Details

Vehicle 1: KQ-8745

Vehicle 2: CAR-2075

Vehicle 3: CAR-6666

Vehicle 4:

Vehicle 5:

### Reservations:

Reservation ID	Date	Car Park	City	Start Time	End Time
1	2017-07-08	Thilakawardana Car Park 1	Kiribathgoda	2017-07-08 11:00:00.000000	2017-07-08 16:00:00.000000
2	2017-07-10	Kandy Municipal Council Car Park	Kandy	2017-07-10 09:30:00.000000	2017-07-10 13:45:00.000000
4	2017-09-19	Kandy Car Park	Kadawatha	2017-09-19 10:00:00.000000	2017-09-19 14:00:00.000000
5	2017-09-23	Kandy Municipal Council Car Park	Kandy	2017-09-23 08:00:00.000000	2017-09-23 15:00:00.000000

### Payment Details:

Payment ID	Date	Car Park	City	Duration	Payment
2	10/9/2017	Thilakawardana Car Park 1	Kiribathgoda	1479hrs	887400

**Figure 6.21 Payment Details**

## 6.2.10 Reports

To generate the reports the administrator has to click on the “Generate Reports” button

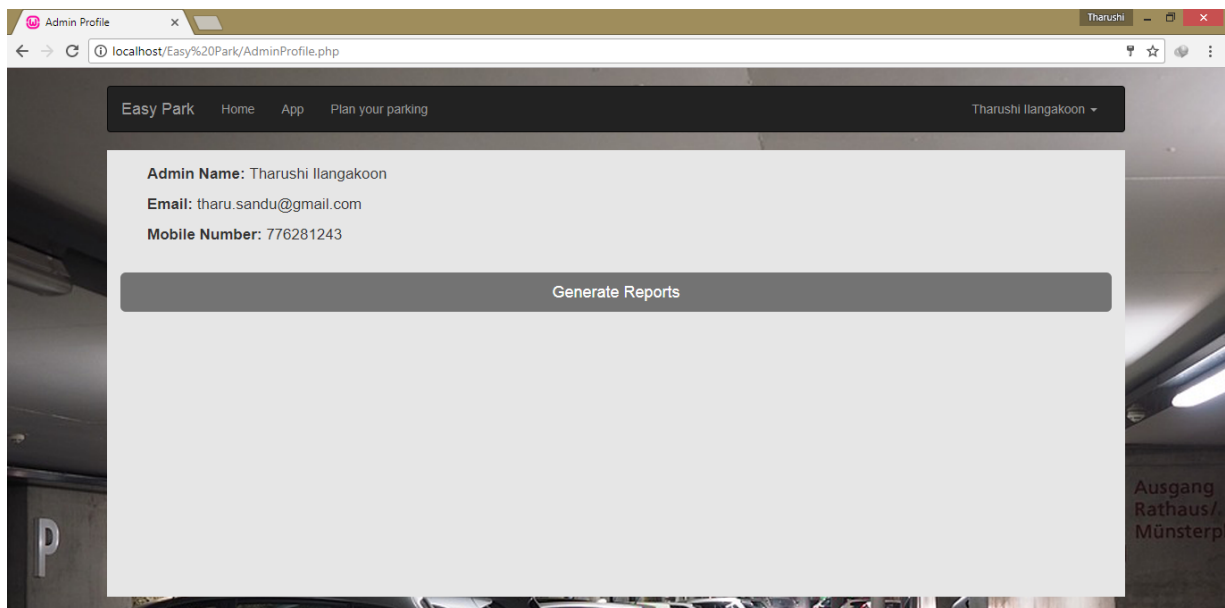


Figure 6.22 Generating Reports

This will direct the administrator to a new page where he can generate different reports.

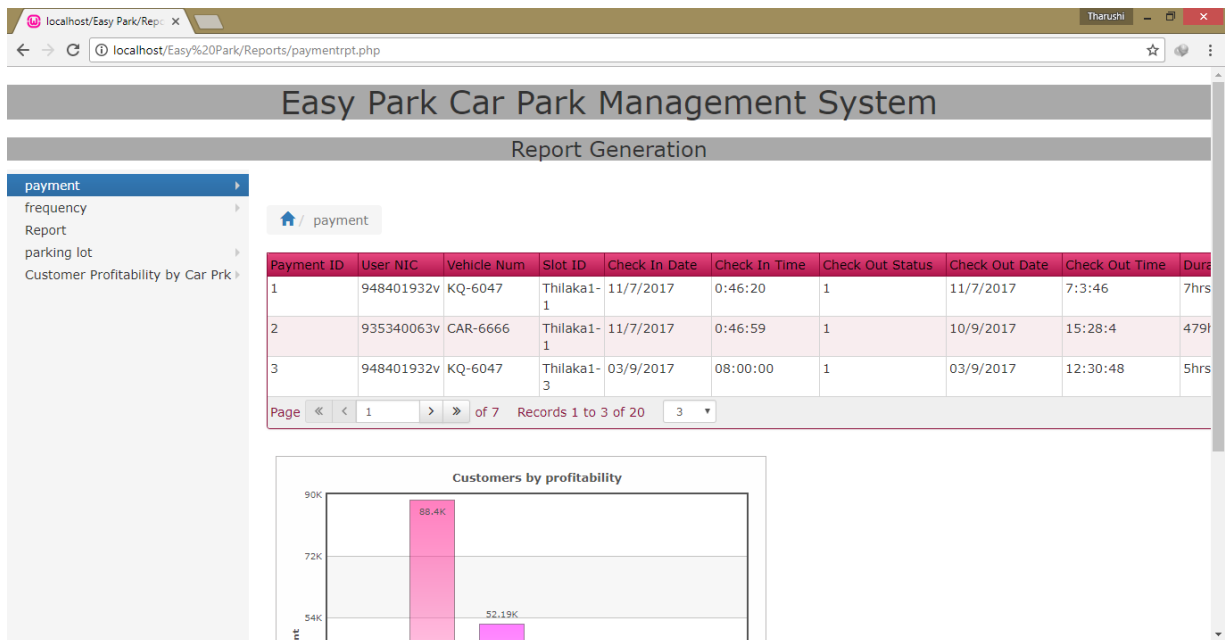


Figure 6.23 Reports

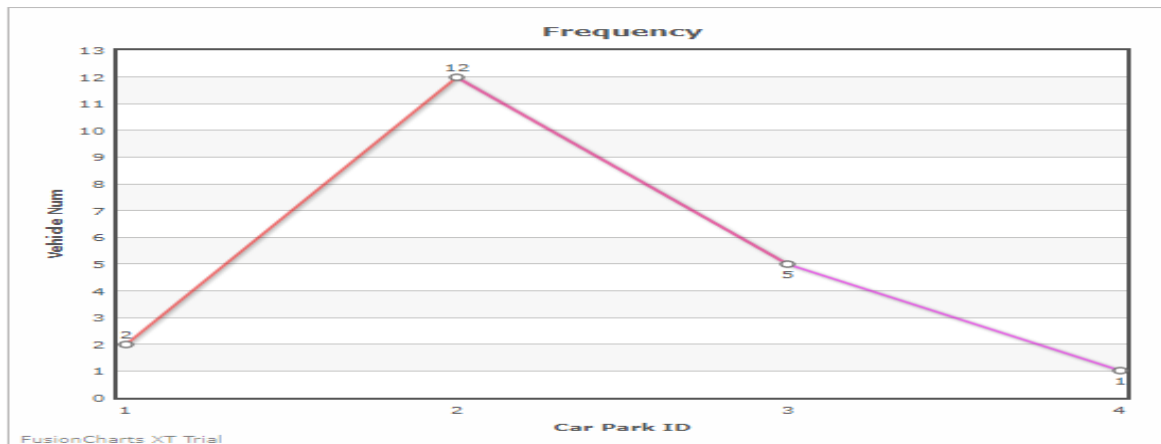


Figure 6.24 Frequency Reports

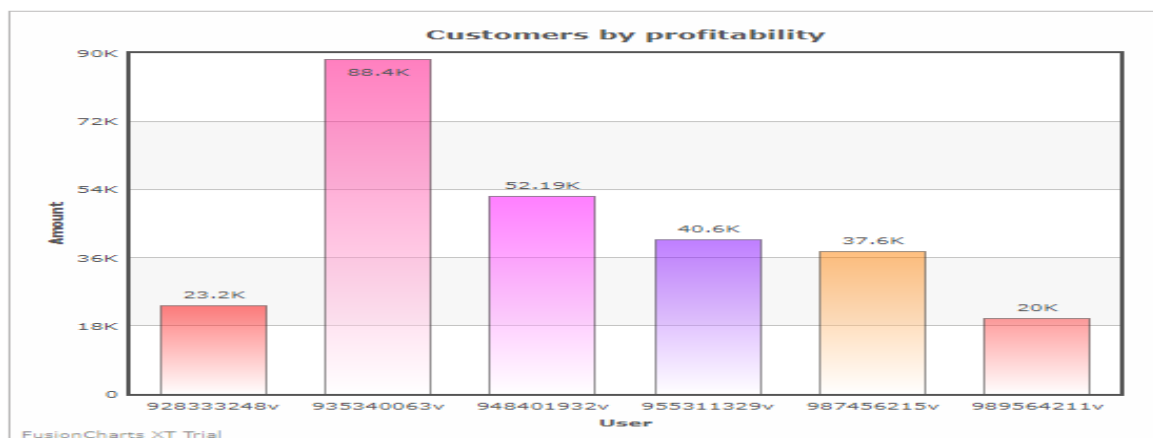


Figure 6.25 Overall Customer Profitability

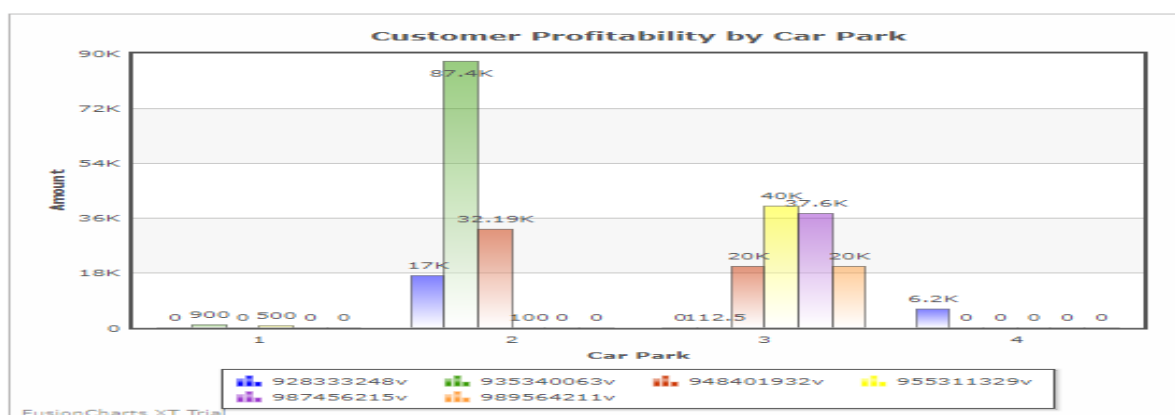
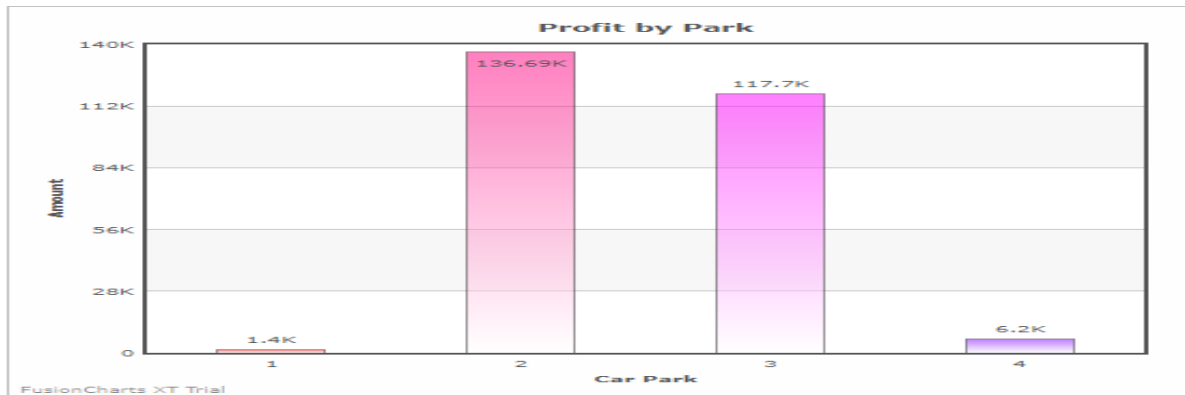


Figure 6.26 Customer Profitability by Car Park Report



**Figure 6.27 Profitability of the Car Parks**

User NIC	Payment ID	Vehicle Num	Slot ID	Check In Date	Check In Time	Check Out Status	Check Out Date	Check Out Time	Duration	Amount
928333248v	9	KW-5000	Thilaka1-2	22/9/2017	16:21:14	1	22/9/2017	16:46:52	361hrs	17000.00
	16	KW-5000	Thilaka1-2	23/9/2017	12:25:16	0				
	17	KW-5000	NavalokaCMB-1	14/9/2017	01:00:00	1	16/9/2017	14:00:00	62hrs	6200.00
COUNT=3										
935340063v	2	CAR-6666	Thilaka1-1	11/7/2017	0:46:59	1	10/9/2017	15:28:4	479hrs	87400.00
	8	CAR-2075	Thilaka1-1	19/9/2017	20:50:38	0				
	11	KQ-8745	KandyMC-2	23/9/2017	6:16:58	1	23/9/2017	6:27:23	5hrs	112.50
	13	KQ-8745	KandyKadawatha - 3	01/9/2017	14:00:00	1	01/9/2017	16:00:00	2hrs	900.00
	15	CAR-6666	Thilaka1-3	23/9/2017	12:23:55	0				
COUNT=5										
948401932v	1	KQ-6047	Thilaka1-1	11/7/2017	0:46:20	1	11/7/2017	7:3:46	7hrs	140.00
	3	KQ-6047	Thilaka1-3	03/9/2017	08:00:00	1	03/9/2017	12:30:48	5hrs	50.00
	5	KQ-6047	Thilaka1-1	10/9/2017	15:14:9	1	10/9/2017	15:26:22	1hrs	10000.00
	7	KQ-6047	Thilaka1-3	18/9/2017	20:43:1	1	18/9/2017	21:31:46	8270hrs	22000.00
	10	KQ-6047	Thilaka1-2	22/9/2017	16:55:3	0				
	14	KQ-6047	KandyMC-2	15/9/2017	08:00:00	1	15/9/2017	22:00:00	14hrs	20000.00
COUNT=6										
Page « < 1 > » of 2 Records 1 to 3 of 6 3 ▼										

**Figure 6.28 Customer Information Report**

The above types of reports can be obtained from the system. These reports can be used to gain additional revenue by selling them to different companies. For an example, the companies can put up stalls for their products and do a promotional campaign for their products. The car park can charge some fee from those stalls.

Another benefit of these reports is that a car park can identify their most profitable customers and introduce some benefits for them such as loyalty scheme or extra reservation time etc.

## 6.3 Summary

In this chapter, the developer has discussed about the implementation of the Easy Park Car Park Management System. It includes the implementation specifications such as software requirements and hardware requirements installation guide, User manual which guides the user to use the system easily.

## CHAPTER 7

### 7. EVALUATION AND CONCLUSION

This chapter evaluates the attempt taken in the project and provides a conclusion of the project. It includes degree of objectives met, limitations and drawbacks, and future modifications, improvements and extensions possible.



## **7.1 Degree of objectives met**

The main objective of “Easy Park” car park management system is to go beyond just helping the drivers to locate a parking space. The system enables the drivers to locate the right parking space. It provides a solution for the problems identified in the current process by improving efficiency and accuracy, minimizing time consumed with finding a parking space, providing timely data and improving the decision making ability.

### **1. Minimize the time consumed with finding a parking space**

The system provides reservation facility by which the driver can reserve a parking space before going to the car park. The location of the car park and the route will be shown through the navigation facility provided by the system.

### **2. Improve efficiency and accuracy**

The entrance time and exit time will be extracted from the system for the calculation of the parking charges. The payment amount will be automatically deducted from the subscription.

### **3. Provide timely data**

Data is transmitted through the sensors and other physical devices in the car park. Based on the transmitted data SMS notifications will be sent to the user to notify and remind them about the updates in reservation facility.

### **4. Improve decision making**

The management decision making process is provided by the new system to develop strategic plans to increase the revenue from the car parks.

## **7.2 Usability, Accessibility, Reliability and Friendliness**

In the development of Easy Park Car Park Management System, discount usability engineering factors were used in a scale of particularity.

- **Visibility of system status**

The system always keeps users informed about what is going on, through appropriate feedback within reasonable time. Alert messages, validation messages, Error Messages with natural Language are always assisting user with the status of the system while performing a task.

- **Match between system and the real world**

The system speaks the users' language, with words, phrases and concepts familiar to the user. System follows the natural and logical order of precedence according to the users' behavior.

- **Consistency and standards**

System maintains the consistency throughout, use standard styles in all pages, to avoid surprises for the users

- **Error prevention**

Prevention is better than Cure. So use of regular expression validation and scripts are preventing users to enter invalid data. Enabling and disabling options when necessary, use of drop down list to select from menus, are some of error prevention methods used.

- **Helps users reorganize, diagnose and recover from errors**

Error messages are expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution when possible.

- **Help and documentation**

System provides a comprehensive user guide with step by step description with pictures

- **Clearly marked exits**

All the pages contain breadcrumbs where the navigational path is indicated with link buttons. User can use breadcrumbs to go back to the previous page visited, or use the sign out link button to sign out from the system at any given time. Main menu bar is always present on the top where users can navigate easily between pages.

### **7.3 Limitations and Drawbacks**

With the time and resource limitations, “Easy Park” – Car Park management System has its limitations and drawbacks with the current completion state.

The main drawback of the system is the inability to know the last minute reservations of a parking space. For an example, the driver may have gone to the car park based on the information provided by the system in the previous minute and within that minute another car may have parked in the parking space.

Another drawback is inability to remove a previously parked vehicle from a parking space if the parking space is reserved for the next hour or so. The next vehicle will face problems if the previous vehicle hasn't been removed on time.

### **7.4 Future Modifications, Improvements and Extensions**

There are several further modifications to the system, which will enhance the quality of the system in terms of usability, reliability and accessibility.

- System can be developed to process the vehicle number plate to automatically add data about the vehicle owner to the system
- Improve the system to be used in other platforms
- Develop a web application integrating the APIs
- Link the system with PayPal

### **7.5 Summary**

In this chapter, the author has described about the degree of objectives met after implementing the project, limitations and drawbacks, and further development of the project.

## REFERENCES

1. Stack overflow : <https://stackoverflow.com>
2. Raspberry Pi Learning resources :  
<https://www.raspberrypi.org/learning/physical-computing-with-python/ldr/>
3. Google Maps APIs:  
<https://developers.google.com/maps/documentation/javascript/get-api-key>
4. Innovative Parking Solutions for Smart Cities: <http://www.iemgroup.com/>
5. W3 Schools : <https://www.w3schools.com/html/>