

**CLOUD BASED PUBLIC BUS LOCATION  
TRACKING AND NOTIFICATION APPLICATION  
FOR MOBILE DEVICES**

**CHEE XIAN YANG**

**Faculty of Computer Science & Software Engineering**

**UNIVERSITI MALAYSIA PAHANG**

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : CHEE XIAN YANG

Date of Birth : 28-8-1995

Title : CLOUD BASED PUBLIC BUS LOCATION TRACKING AND  
NOTIFICATION APPLICATION FOR MOBILE DEVICES

Academic Session : 2018/2019

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Student's Signature)

\_\_\_\_\_  
(Supervisor's Signature)

\_\_\_\_\_  
950828-01-5413  
Date: 10 January 2019

\_\_\_\_\_  
DR SYAFIQ FAUZI BIN  
KAMARULZAMAN  
Date: 10 January 2019

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



## **SUPERVISOR'S DECLARATION**

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Computer System and Networking) with Honors.

---

(Supervisor's Signature)

Full Name : DR SYAFIQ FAUZI BIN KAMARULZAMAN

Position : SUPERVISOR

Date : 10 January 2019



## **STUDENT'S DECLARATION**

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

---

(Student's Signature)

Full Name : CHEE XIAN YANG

ID Number : CA15094

Date : 10 January 2019

CLOUD BASED PUBLIC BUS LOCATION TRACKING AND NOTIFICATION  
APPLICATION FOR MOBILE DEVICES

CHEE XIAN YANG

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Bachelor of Computer Science (Computer System and Networking)

Faculty of Computer Systems and Software Engineering

UNIVERSITI MALAYSIA PAHANG

JANUARY 2019

## **ACKNOWLEDGEMENTS**

Firstly, I would like to express my deepest appreciation to my parents for always giving me encouragement and moral support for me to continue my study up to the degree level. Also, I would like to thank all the lecturers who educate me during my 4 years of study in Universiti Malaysia Pahang.

I must express my greatest gratitude to my supervisor, Dr. Syafiq Fauzi bin Kamarulzaman who always give me guidance through out the project development. It is a great honor to do this project under his supervision. He provided insight comment and outstanding technical advice as well as moral support to me all the time.

Lastly, I would like to thank to all my friends who had given me idea, guidance and sharing of knowledge throughout the project development.

## ABSTRAK

Pada masa kini, penggunaan pengangkutan bas awam adalah salah satu fenomena yang biasa di Malaysia. Walau bagaimanapun, tanpa sistem pengurusan yang baik, perkhidmatan bas mungkin tidak dapat dilaksanakan secara efisien. Penumpang masih perlu menunggu lama untuk ketibaan bas di perhentian bas bersama dengan cuaca yang cukup panas dan mempunyai peluang yang tinggi terlepas bas. Oleh itu, ini adalah sistem pengesanan bas yang dibuat untuk menyelesaikan masalah tersebut. Aplikasi mudah alih berasaskan Android yang diintegrasikan dengan Google Map akan dicipta untuk menjejak lokasi bas awam dan memberikan anggaran masa ketibaan setiap bas kepada pengguna. Oleh itu, pengguna boleh pergi ke perhentian bas berdasarkan masa yang diberikan daripada menunggu di perhentian untuk bas datang. Pada masa yang sama, aplikasi mudah alih akan dicipta untuk pemandu bagi memberikan lokasi mereka kepada pengguna secara langsung dan aplikasi berasaskan web akan diwujudkan untuk syarikat bas untuk mendaftar laluan bas mereka, menguruskan pemandu mereka serta menguruskan bas yang dimiliki oleh syarikat tersebut. Keseluruhan sistem akan dibangunkan melalui kitaran RAD dan mempunyai empat fasa iaitu perancangan, perekaan sistem, pembinaan dan pemotongan fasa. Prototaip sistem dijangka dapat disiapkan setiap ulangan fasa pembinaan. Sistem ini bertujuan untuk memberikan manfaat kepada masyarakat atau orang yang sering menggunakan pengangkutan bas awam. Walau bagaimanapun, peningkatan masa depan sistem diperlukan untuk memastikan bateri telefon tidak habis begitu cepat disebabkan servis lokasi GPS dan sistem ini mungkin tidak dapat memberi manfaat kepada pengguna iOS pada peringkat awal.

## **ABSTRACT**

Nowadays, public bus transportation had become more common in Malaysia. However, without a good management system, the bus service might not be fully utilized. Passenger still have to wait for long time under the hot weather for the bus to come and might have higher chances to miss the bus. Therefore, this is a real time bus tracking system developed to solve the problem. An Android based mobile application integrated with Google Map is created to stream the location of public buses in real time and provide estimated arrival time of each buses to the users. Hence, user can go to the bus stop based on the time given instead of waiting at the bus stop under the hot weather. At the same time, the mobile application will be created for the driver to provide their location to the user in real time and a web-based application will be created for the bus company to register their route, manage their driver as well as manage the buses that belong to the company. The whole system will be developed through the RAD cycle and having four phases which is requirement planning, user design, construction and cutover phase. The prototype of system is expected to deliver every loop of the construction phase. They system is aimed to give advantage to the community or people who often using public bus transportation. However, future enhancement to the system is needed to make sure that the phone battery does not drain so fast due to the GPS location service and the current constrain of this system is that it might not benefit to iOS users.



## TABLE OF CONTENT

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 BACKGROUND	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVE	3
1.4 SCOPE	4
1.5 SIGNIFICANCE	4
1.6 THESIS ORGANIZATION	4
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>6</b>
2.1 INTRODUCTION	6
2.2 STUDIES ON TECHNOLOGY INVOLVED	6
2.2.1 GLOBAL SYSTEM FOR MOBILE COMMUNICATION	6
2.2.2 GLOBAL POSITIONING SYSTEM	7

2.2.3	MOBILE APPLICATION PLATFORM – ANDROID OPERATING SYSTEM	9
2.3	COMPARISION OF MOBILE APPLICATION FRAMEWORK	11
2.3.1	XAMARIN	11
2.3.2	IONIC FRAMEWORK	12
2.3.3	INTEL XDK	13
2.4	COMPARISON OF MAPS API	13
2.4.1	GOOGLE MAPS API	13
2.4.2	BING MAPS API	14
2.4.3	APPLE MAPS API	16
2.5	REVIEW OF EXISTING SYSTEM	16
2.5.1	ASIA PASIFIC UNNIVERSITY (APU) BUS TRACKING SYSTEM	17
2.5.2	UiTM CAMPUS BUS TRACKING SYSTEM USING ARDUINO BASED AND SMARTPHONE APPLICATION (UiTM BUS CHECKER)	18
2.5.3	IoT BASED SCHOOL BUS TRACKING AND ARRIVAL TIME PREDICTION SYSTEM.	20
2.5.4	COMPARISON OF EXISTING SYSTEM	22
2.6	Summary	23
	<b>CHAPTER 3 METHODOLOGY</b>	<b>24</b>
3.1	INTRODUCTION	24
3.2	RAPID APPLICATION DEVELOPMENT (RAD)	24
3.2.1	REQUIREMENT PLANNING	25
3.2.2	USER DESIGN	25
3.2.2.1	Use Case Diagram	26
3.2.2.2	Context Diagram	27

3.2.2.3	General Flowchart of The Prototype	27
3.2.2.3.1	BusSeeker (Passenger Apps)	27
3.2.2.3.2	BusSeeker (Driver Apps)	29
3.2.2.3.3	Bus Seeker Management System	30
3.2.2.4	USER INTERFACES	31
3.2.3	CONSTRUCTION PHASE	38
3.2.4	CUTOVER PHASE	39
3.3	GENERAL ARCHITECTURE AND PACKAGE MODULES	39
3.3.1	GENERAL ARCHITECTURE	39
3.3.2	PACKAGE MODULE	40
3.3.2.1	Tracking Module	41
3.3.2.2	Notify Driver	41
3.3.2.3	Driver Login	41
3.3.2.4	Activate Route	41
3.3.2.5	Company Registration	41
3.3.2.6	Company Login	41
3.3.2.7	Manage Bus	42
3.3.2.8	Manage Route	42
3.3.2.9	Manage Driver	42
3.3.2.10	View Feedback	42
3.4	HARDWARE AND SOFTWARE	42
3.4.1	HARDWARE REQUIREMENT	42
3.4.2	SOFTWARE REQUIREMENT	43
<b>CHAPTER 4 RESULTS AND DISCUSSION</b>		<b>44</b>
4.1	INTRODUCTION	44
4.2	IMPLEMENTATION	44
4.2.1	TOOLS AND TECHNOLOGY	44
4.2.2	DATABASE IMPLEMENTATION	46
4.2.3	INTERFACE DESIGN DESCRIPTION	47

4.2.3.1	BusSeeker Driver Apps	47
4.2.3.2	BusSeeker Passenger Apps	50
4.2.3.3	BusSeeker Management System	54
4.3	TESTING AND RESULT DISCUSSION	58
4.4	USER MANUAL	59
<b>CHAPTER 5 CONCLUSION</b>		<b>60</b>
5.1	INTRODUCTION	60
5.2	PROJECT CONSTRAINTS	61
5.3	FUTURE WORK	62
<b>REFERENCES</b>		<b>63</b>
<b>APPENDIX A GANTT CHART</b>		<b>65</b>
<b>APPENDIX B USER ACCEPTANCE TEST</b>		<b>66</b>
<b>APPENDIX C USER MANUAL</b>		<b>76</b>
1.0	GENERAL INFORMATION	76
1.1	System Overview	76
1.2	Organization of the Manual	76
2.0	SYSTEM SUMMARY	77
2.1	System Configuration	77
2.2	User Access Level	77
2.3	Contingencies and Alternate Mode of Operation	77
3.0	GET STARTED	78
3.1	BusSeeker (Driver Apps)	78
3.1.1	Driver Login	78

3.1.2 Activate Route	78
3.1.3 Location Tracking	79
3.2 BusSeeker (Passenger Apps)	80
3.2.1 Bus Stop and Route Selection	80
3.2.2 Search Bus	81
3.2.3 Provide Feedback	83
3.3 BusSeeker Management System	83
3.3.1 Management System Login Page	83
3.3.2 Management System Sign Up Page	84
3.3.3 Manage Bus	85
3.3.4 Manage Route	85
3.3.5 Manage Driver	87
3.3.6 Manage Feedback	89

## LIST OF TABLES

Table 1.1 Problem statements in the project	2
Table 2.1 Comparison of features between maps provided by three company.	16
Table 3.1 Hardware requirement	42
Table 3.2 Software requirement	43

## LIST OF FIGURES

Figure 2.1 GSM location tracking through the process called triangulation	7
Figure 2.2 GPS localization through trilateration.	8
Figure 2.3 AGPS provide assisting position information to the smartphone.	9
Figure 2.4 Difference between traditional Xamarin with Xamarin Forms	12
Figure 2.5 Bing's Isoshrone API	15
Figure 2.6 Bing's Snap-to-Road API	15
Figure 2.7 Asia Pacific University (APU) Bus Tracking System webpage.	18
Figure 2.8 Overall system architecture of UiTM Bus Checker.	19
Figure 2.9 Overall UiTM Bus Tracking System	19
Figure 2.10 Overall system architecture diagram	21
Figure 2.11 Travelling path of school bus, scheduled and predicted arrival time of bus.	21
Figure 3.1 RAD phases.	25
Figure 3.2 Use Case Diagram	26
Figure 3.3 Context Diagram	27
Figure 3.4 Flow Chart of BusSeeker (Passenger Apps)	28
Figure 3.5 Flow Chart of BusSeeker (Driver Apps)	29
Figure 3.6 Flow Chart of BusSeeker Management System	30
Figure 3.7 GUI of BusSeeker mobile apps for passengers.	31
Figure 3.8 GUI of BusSeeker mobile apps for driver.	32
Figure 3.9 Sign up page	32
Figure 3.10 Login page.	33
Figure 3.11 Bus adding and deleting page.	33
Figure 3.12 Adding bus route.	34
Figure 3.13 Delete bus route.	35
Figure 3.14 Showing detail of each route.	35
Figure 3.15 Register driver to the system.	36
Figure 3.16 Update driver information.	37
Figure 3.17 User feedback	37
Figure 3.18 Update company information	38
Figure 3.19 General architecture of the system.	40
Figure 3.20 System package module.	40
Figure 4.1 Visual Studio	44

Figure 4.2 Microsoft Azure	45
Figure 4.3 Microsoft Azure portal interface for app service.	45
Figure 4.4 Entity Relationship Diagram (ERD)	46
Figure 4.5 Driver login interface	47
Figure 4.6 Route activation interface	47
Figure 4.7 Tracking interface	48
Figure 4.8 User interface of BusSeeker passenger apps.	50
Figure 4.9 Search bus	51
Figure 4.10 Notification	52
Figure 4.11 Feedback	52
Figure 4.12 Exit confirmation dialog	53
Figure 4.13 Management system login page	54
Figure 4.14 Management system signup page	54
Figure 4.15 Bus management page	55
Figure 4.16 Route management page (add new route)	55
Figure 4.17 Search route	56
Figure 4.18 Route details	56
Figure 4.19 Driver management page.	57
Figure 4.20 Update driver information	57
Figure 4.21 Feedback management page	58
Figure 4.22 Update company information	58



## LIST OF ABBREVIATIONS

A-GPS	Assisted Global Positioning System
API	Application Programming Interface
AXML	Article XML Markup Language
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DBaaS	Database-as-a-service
DFD	Data Flow Diagrams
ERD	Entity Relationship Diagram
ETSI	European Telecommunication Standards
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile
HDD	Hard Drive Disk
HTML	Hyper-Text Markup Language
HTTP	Hypertext Transfer Protocol
IC	Identity Card
ID	Identity
iOS	Iphone Operating System
IoT	Internet of Thing
IP	Internet Protocol
KML	Keyhole Markup Language
LOS	Level of Service
LRT	Light Rail Transit
LTE	Long Term Evolution
PHP	Personal Home Pages
RFID	Radio Frequency Identification
SDLC	System Development Life Cycle
SDM	System Development Methodology
SIM	Subscriber Identity Module
SMS	Short Message Service
SSD	Solid State Drives
TCQSM	Transit Capacity and Quality of Service Manual
TDMA	Time Division Multiple Access
TIFF	Time to First Fix

UI	User Interface
UML	Unified Modeling Language
Wi-Fi	Wireless Fidelity
XAML	Extensible Application Markup Language

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 BACKGROUND**

Around the world, public bus services have undoubtedly play a very important role to provide transportation and mobility for community especially for those who can't afford for the ownership of car. However, public bus transportation services are not the first choice of urban travel in Malaysia. According to the Nielsen Global Survey of Automotive Demand, Malaysia ranked third in the world with 93% of car ownership. The reasons behind this is because public transport is not as attractive as private vehicle such as inflexibility, no direct access, longer travel and waiting time and unsafe when travelling (Kamarudin Ambak & Rozmi Ismail, 2012). In fact, public bus transportation does bring a lot of benefit to the community. It can help to reduce congestion, reduce air pollution as well as saving transportation fee.

The Global Positioning System (GPS) is a navigation system that use satellite technology to pinpoint or determine the geographical location of ground object. The GPS technology was initially used by United State military in early 1970s and is available for general use nowadays. GPS is currently one of the most popular navigation system which integrated in many vehicle, and mobile devices. The geographical location obtained can be manipulated to derive useful information such as best route between two points, instantaneous speed, monitoring movement and locating lost person or stolen subjects (Vermaat, 2014).

Therefore, a mobile application can be created by applying the GPS technology to solve the schedule punctuality problem and long waiting time of public buses. From the application, user will be able to get the current location and the estimated arrival time of the buses in real time, so that they can plan their journey earlier and take their time to reach the bus stop instead of spending more time waiting for the bus. At the same time

another mobile application is needed for the driver to share the bus location in real time. User may also alert bus driver that he or she is waiting at particular bus stop so that the driver is ready to stop for the fetching purpose.

According to the survey done by the Malaysian Communication and Multimedia Commission (MCMC) based on the feedback from the poll's 2787 respondents, the smartphone ownership among the internet user had reached 90.7% and 87.3% of them preferred to use mobile broadband as choice of internet access. Hence the mobile application is useful to track the location of buses anywhere, anytime and expected to give benefit to the user or passenger who often using public bus transportation thus encourage more people taking public bus transportation.

## 1.2 PROBLEM STATEMENT

**Table 1.1 Problem statements in the project**

No	Problem	Description	Effect
1.	Bus schedule provided is not accurate.	Bus company does provide bus schedule for every route, however the actual arrival time might not accurate sometimes.	The bus tends to arrive earlier or delay hence, passenger might have higher chances to miss the bus.
2.	High accident rate.	Bus driver might have to rush for the scheduled timetable could lead to serious accident happen.	Passenger feels unsafe when traveling.
3.	Long waiting time of the buses.	User might have to spend a certain amount of time waiting for the next bus to come.	User unable to predict the arrival time of the buses and have to wait for a certain period of time under the sun.

According to the Table 1.1 above, there are several problems are facing by the public bus passenger everyday life in Malaysia. One of the problems is that the bus

schedule provided is not accurate. Bus company does provide the bus schedule for every route, however the actual arrival time might not accurate sometimes. The bus tends to arrive earlier or delay, hence passenger might have higher chances to miss the bus. Besides that, passengers have to spend a certain amount of time waiting for the bus to come to the bus stop. Passenger unable to predict the arrival time of the buses and have to wait for a certain period of time under the hot weather. According to a study done by the Universiti Teknologi PETRONAS Malaysia collaboration with Gadjah Mada University Indonesia on the passengers' waiting time and punctuality of stage bus operation in Malaysia, the result showing an average punctuality index of 0.29 and expected waiting time of 28 minutes. Based on the punctuality index, by referring to the TCQSM 2003 standard the bus system service reliability is considered to LOS B, meaning that vehicles are slightly off headway (M. Napiah, I. Kamaruddin, & Suwardo, 2011). Lastly, high accident rate also one of the problems facing by the community. Passenger will fell unsafe since driver might have to rush for the scheduled timetable could lead to serious accident happen.

### **1.3 OBJECTIVE**

The goal of this project is to develop a tracking system to convince the public bus user in Malaysia with the following objectives:

- I. To develop an application for smartphone that use the GPS technology to track the actual public bus location in real time.
- II. To develop an application for smartphone to notify the user the actual location of public buses with the estimated arrival time of each bus.
- III. To evaluate the applications fulfil the requirements through User Acceptance Testing (UAT).

## **1.4 SCOPE**

Based on the declared objective, the scope of this project is divided into 2 parts, which are:

- I. Mobile application to provide the public bus location and the estimated arrival time to the user in real time.
- II. Web based application to provide the company a platform to register their route for the buses.

## **1.5 SIGNIFICANCE**

There are few significant in this project and will be explained as below:

- I. Encourage more people taking public bus transportation.
- II. Allow passenger to take their time going to the bus stop based on the estimated arrival time provided and reduce the waiting time of public buses.
- III. Help the bus driver to get notify there are passengers waiting at certain bus stop.

## **1.6 THESIS ORGANIZATION**

This thesis is divided by chapter and it contains six chapters. Chapter 1 give the rough idea of the system and discuss on the introduction which include the background of study, problem statements, objectives, project scope and significant.

Chapter 2 discuss about the literature review which is about the review and compare the similar existing system based on the features, advantage and disadvantage. This chapter also focus on reviewing the latest technology that could be applied in this project.

Chapter 3 discuss the overall approach and framework of the project. In this chapter briefly and justify the method used during the development of the project. Besides, chapter 3 discussed the technique, software and hardware used in this project.

Chapter 4 will discuss the implementation of the project. Describe about the system development and implementation. Tracking system will be deployed, and the result would be tested. It also included the explanation that showed the objectives of the project is fulfilled.

Lastly, chapter 5 would be the conclusion of the project findings and also the discussion about the future enhancement that can be implemented to improve the project requirement.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

This chapter contained the information about the study of the projects in general. It describes briefly about the existing system. This chapter explains in detail technologies which are suitable to be adapted into the project.

#### **2.2 STUDIES ON TECHNOLOGY INVOLVED**

Studies on major technologies involved are necessary for a system development. The Global System for Mobile Communication (GSM), Global Positioning System (GPS) and Android mobile application platform will be discussed detailed as following.

##### **2.2.1 GLOBAL SYSTEM FOR MOBILE COMMUNICATION**

Global System for Mobile Communication, commonly known as GSM is a standard to describe the protocol for second generation (2G) digital cellular network used by mobile devices using a variation of Time Division Multiple Access (TDMA). It is developed by the European Telecommunication Standards Institute (ETSI) and first deploy in Finland in December of 1991. Starting from year 2004, it has become global standard for mobile communication and became the most widely used digital wireless telephony technology nowadays. GSM compresses and digitalizes data and sends it down a channel with 2 other streams of user data, each in its own time slot. It operates at either 900, 1800 or 1900MHz frequency band and using sim card to identify users' account.

GSM specification as a baseline is then evolved to the 3G,4G and LTE technology that we are currently using today. With these mobile communication technologies, GPS



data obtained in the form of latitude and longitude can be send to server and access anywhere, anytime by the users. Mobile application may translate the GPS signal and utilize the GSM capabilities to communicate the data over the service provider’s cellular network. Hence, by applying these communication technology, public bus location can be track in real time. GSM can also be used for the location tracking purpose if GPS service is turned off or there is no GPS signal found in an area such as inside the tunnel or building. GSM will localize the position by using the signal strength from 3 closest GSM base station via the process called triangulation. Localization involve GSM may save the phone battery from draining off however it is unable to provide the precise position as the GPS do.

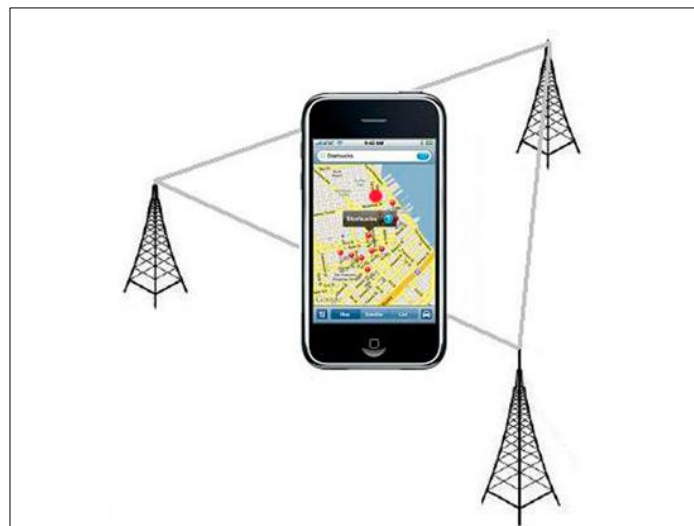


Figure 2.1 GSM location tracking through the process called triangulation

### 2.2.2 GLOBAL POSITIONING SYSTEM

The Global Positioning System (GPS) is a satellite-based navigation system which consist of 24 GPS satellite in the orbit at 12000 miles away from the earth. There are constantly moving, making 2 complete orbits in less than 24 hours. The GPS satellites are powered by solar energy and was given the official name of “NAVSTAR” by the US Department of Defence. For the GPS to work, the receiver must receive and analyse the signal from at least 3 GPS satellites and compute the exact current location in the form of latitude and longitude through the process called trilateration. With the help of forth satellite, the altitude of the object can also be calculated. Commercially, GPS was used for navigation and as position to vehicle, plane and ship.

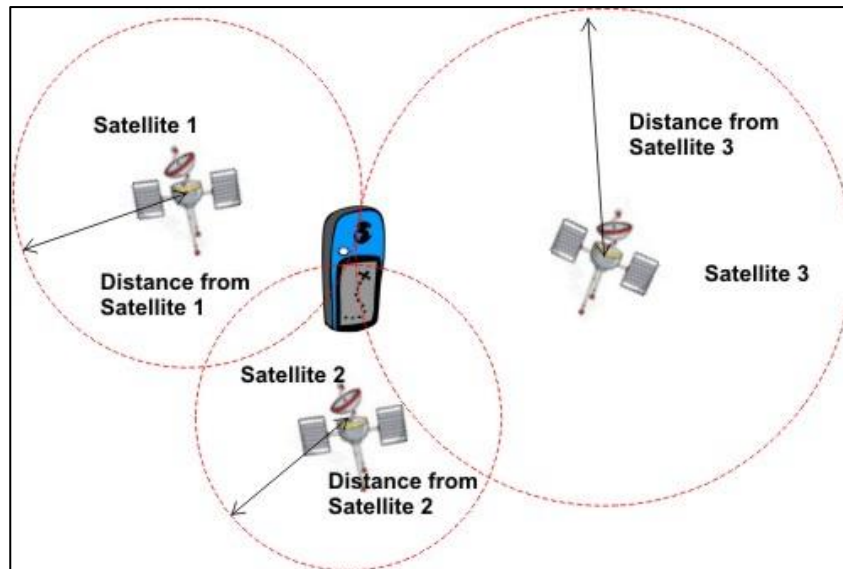


Figure 2.2 GPS localization through trilateration.

Other than traditional GPS, an enhanced version called Assisted GPS also known as A-GPS is introduced in smartphones or mobile devices connected to the cellular network. By default, build-in GPS receiver in smartphone will only receive the position information from the satellite. However, during certain condition or certain places such as area with many buildings around, signal from satellite will be reflected away by the building. Thus, A-GPS is introduced to solve this problem.

A-GPS can help the smartphone to obtain a faster “Time to First Fix” (TIFF) which mean users are able to get faster signal acquisition and position information in their phone. This is because the GPS data does not need to download from the satellite as A-GPS already acquired the location data from satellites and stores it within the cellular network tower. Therefore, user just have to pull the information from the nearest tower when request for the position. A-GPS uses proximity distance between the smartphone and the cellular tower to calculate the position for the user (Faccioni, 2016).

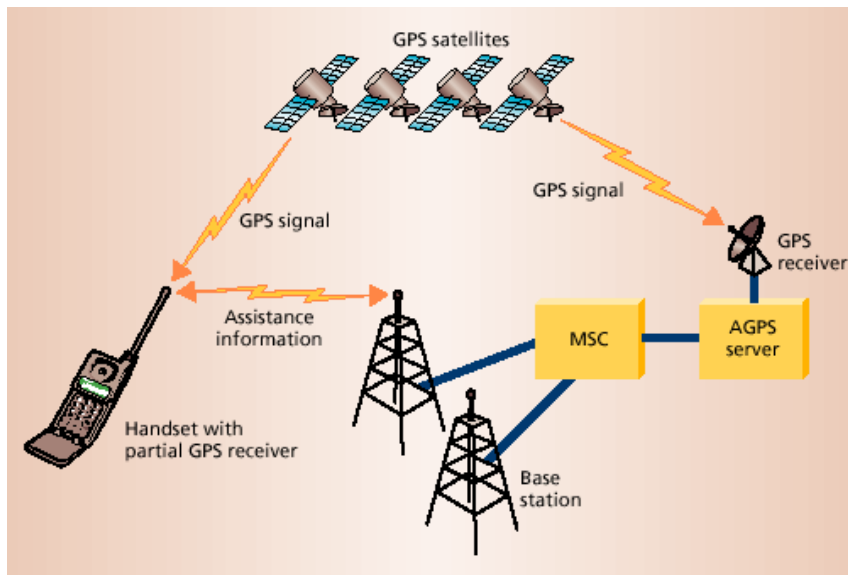


Figure 2.3 AGPS provide assisting position information to the smartphone.

Nowadays, many smartphones in the market are embedded with the GPS module and receiver. Therefore, with the help of Google Map it can be utilized to trace the location of public buses in real time. Public bus users wish to know the current location and estimated arrival time of each buses so that they can go to the bus stop at the correct time instead of waiting there for a long time. GPS acts as the best medium of communication between the passenger and the bus driver. The mobile application utilizes the GPS system to track the position of buses and send it to the database server. The location information is then sent to the user's phone using the cellular network modem. User may also alert the bus driver with the pickup location so that the driver is ready to stop.

### 2.2.3 MOBILE APPLICATION PLATFORM – ANDROID OPERATING SYSTEM

Operating system have developed a lot in last 15 years. Android is a Linux based operating system developed by Google Inc. mainly for mobile devices such as smartphones and tablets. Nowadays, the latest version of android is Android 8.1 (Oreo) which was released on December 5, 2017.

According to the mobile usage statistic and trends provided by RapidValue, smartphone powered by Android has the highest market share in September 2016 with a

percentage of 69.18% followed by iOS at 25.02%. There are some reasons why android is better than iOS and is stated as follow.

1. **Multiple devices and brands** – Android operating system is applicable for variety of devices and smartphone from various company such as Samsung, HTC, Sony, Huawei, etc.

2. **Multiple price level and choices** – Because of many devices are supporting Android application, various of model of smartphones with different design and specification are available in the market from low price to higher prices. So that, everyone is affordable to the smartphone and can purchase according to budget.

3. **Customization** – Android based applications are highly customizable and easier to manage (AA, 2017). Google is more focus on making its user interface customizable to help developers create custom mobile application for business. iOS phone always keeps control user from customization.

4. **Integrated with Google services** – Android phones are always integrated with most commonly use Google application such as Google Map, Gmail, Google Chrome, Google Drive, etc.

Google Map is the most used smartphone application in the world. According to data published by Global Web Index, 54% of smartphone users around the world used Google's navigation application in the month preceding the survey (Richter, 2013). This figure showing that most of the smartphone users are using their smartphone to locate their position and navigate for direction to a destination. People tend to use smartphone for positioning and navigate to the destination instead using a pure GPS device because the smartphone today is embedded with GPS feature (Kendrick, 2011).

In term of develop a GPS application in Android operating system platform, Google does provide variety of convenience tools and APIs for the developers to develop application with GPS features. The GPS features can be easily customized according to the needs. Besides, Google is also provided Google Map API for developers to integrate Google Maps into the project, so that the location can be easily pinpoint on it.

In this bus tracking system, Android mobile applications will be created. This is due to the advantages provided by the Android as listed above. Firstly, Android is applicable for many devices with different prices ranged from low to high for different model and specification. So that, most of the market share is taken by Android and can say that everyone is able to own a smartphone. Furthermore, GPS features can be easily created by the developers because of the useful tools provided by Google Inc. Therefore, Android is selected as the platform for the mobile application development of proposed system.

## **2.3 COMPARISION OF MOBILE APPLICATION FRAMEWORK**

Nowadays, many developers are more prefer working on hybrid mobile application development as they are only required to write once and build the mobile application to run on different platform. The application may run on Android or iOS with a single language and the code can be reused for progressive web application or even desktop. There are many kinds of framework can be used to develop the mobile application such as Xamarin, Ionic, Intel XDK, Kendo UI, Sencha Touch etc. In this project, Xamarin framework will be use because of its ease of integrate with Microsoft Azure cloud computing service.

### **2.3.1 XAMARIN**

Xamarin is a software company founded in the year of 2011 and owned by Microsoft Corporation in the year of 2016. Xamarin provides developers with tools to build the cross-platform mobile application (LavishT, 2017). The mobile applications built will have all the native features by sharing the same code. The only different is the coding for user interface which iOS is using storyboard, Android is using AXML and Windows is using XAML to create the interface. With Microsoft Visual Studio, Xamarin tools can be download and integrated to create Android, iOS and Windows application. The language used is C# which is similar to JAVA, both are object-oriented.

There are few reasons why Xamarin is a great tool for develop native application. Xamarin uses C# complemented with .Net framework to creates apps for any mobile platform. Xamarin also simplified code sharing across multiple mobile operating system such as Android and iOS. Thus, it greatly shortens the time of development cycle and increase the efficiency. Second, Xamarin always believe on sharing and collaboration of

UI for different mobile operating system. Thus, they created the component store which is built right into the Xamarin Studio or Xamarin's Visual Studio extensions. So, developers may choose from a host of free or paid component including UI controls, third party web service and cross-platform libraries to apps with just a few lines of code (Shroff, 2014). Lastly, by using Xamarin together with Microsoft Visual Studio, developers are easier to integrate the project together with Microsoft Azure cloud services such as online database, storage, analytical tools as well as artificial intelligence machine learning etc. The only cons to be consider is that Xamarin is not suitable for heavy graphics application and the community is not as large as those Android and iOS, so finding of good resources will be challenging.

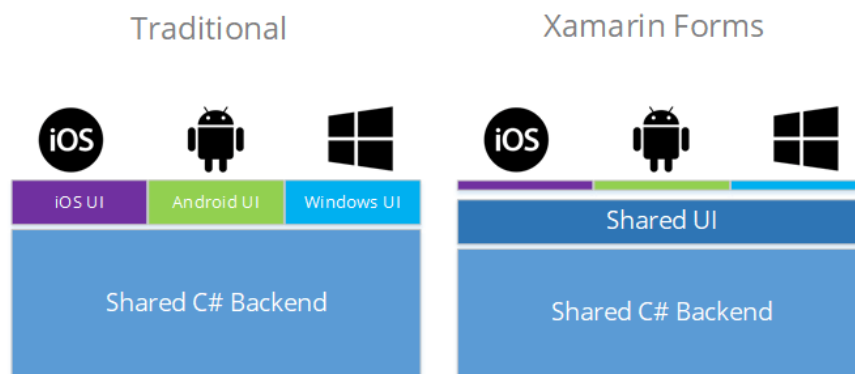


Figure 2.4 Difference between traditional Xamarin with Xamarin Forms

### 2.3.2 IONIC FRAMEWORK

Ionic was founded in the year of 2013 by Drifty Corporation to create a better way for web developer to build application for the app store with their existing skills. Ionic provides tools and services to develop hybrid mobile application using web technology such as HTML5, CSS and JavaScript. Ionic technology is targeting to deliver a solution to create both mobile web application and native apps which can be effectively deployed on Android, iOS and Windows phone.

One of the benefits of using Ionic framework is that they are low cost and having a lot of developers' community support. Besides, this framework come with a lot of CSS component and JavaScript component which allow developers to create thing such as Tabs, Sliding Boxes, Menu, Button, Navigation Pane etc. making the user interface more beautiful and interactive (Tucker, 2015).

### **2.3.3 INTEL XDK**

Intel XDK is an Integrated Development Environment (IDE) created by Intel for developing cross platform mobile applications. Same as Ionic, Intel XDK also code by using HTML5, JavaScript and CSS. Mobile application for Android, iOS, Windows, Tizen, Amazon and Nook can be created with just a single codebase (Prusty, 2015). By using Intel XDK, developer does not need to have extra development skill or even learning a new language because it can be just develop using a web scripting language. By doing so, the cost of development could also be reduced as enterprise does not need to hire another group of developers dedicated for native apps development. Intel XDK integrates testing, debugging and build tools into one integrated solution so that developers does not need to waste the time to the to switch between various tools and at the same time the productivity increased too.

## **2.4 COMPARISON OF MAPS API**

An Application Program Interface (API) is a framework that used to write a program. It consists of a set of classes and functions that prevent developers from writing the program from low level language to achieve specific action. The map API typically consist of classes for maps and layers so that developers can display the interactive map image and draw new layer on the application by simply create a new map object, new layer object or call some method. The map API simplified the complexity of mapping task and allows developers focus on apps development instead of low-level logic. Three most popular Maps' API will be discussed which is Google Maps, Bing Maps and Apples Maps.

### **2.4.1 GOOGLE MAPS API**

Google Maps API is actually a useful tool that can be used to create custom map, searchable map, route planning, check-in function as well as display live location streaming. It allows developers to overlay their own data on top of the map. The overlay data is normally supplied through the KML file and display as an interactive vector graphic drawn on the client side. Developers are allowed to restyle the graphics and make their own custom marker symbol. Popup message can also bound to the map to show additional information. The biggest advantages of Google Maps API is that it can bring

the feel and look of the Google Map to the system. By integrate the Google Map into a system, users will feel comfortable and get use to the system because majority of the internet user will have the experience of using Google Map in their daily life such as searching for location and navigation.

The reason to use Google Maps API is because it is very powerful ecosystem as it mashup of more than one API (Lam, 2015). Google Maps API is so formidable that Bing Maps and Apple Maps are seriously not finding their stand at all. Instead of ordinary Map API, Google provide the API dedicated for certain function. For example, Direction API is used to compute the direction between locations by using a HTTP request. The calculation is can be based on certain mode of transportation such as walking, cycling, driving and transit. Besides that, a Distance Matrix API provided by Google which is very useful in this project. It could be used to provide the calculation of distance and travel time between two location. In addition, the Places API provided by Google is also useful in this project by filtering out all the bus station on the map.

#### **2.4.2 BING MAPS API**

Microsoft's Bing Maps is also another large commercial map provider and the services is offered to web and mobile application as what Google had done. The difference is that Bing Maps does not emphasize on KML usage, since Google popularized the format. Three features of Bing that should be highlighted is Truck Routing API, Isochrone API and the Snap-to-Road API. Truck Routing API calculates routes optimized for trucks and commercial vehicles. Information such as travel distance, travel duration and route instruction are provided by considering the specific requirement such as avoid low bridges, sharp turn and steep road etc. Isochrone API utilize the distance polygon to show the area can be travelled to within specific distance or time. The Snap-to-Road API are able to form a route base on the GPS data point obtained.



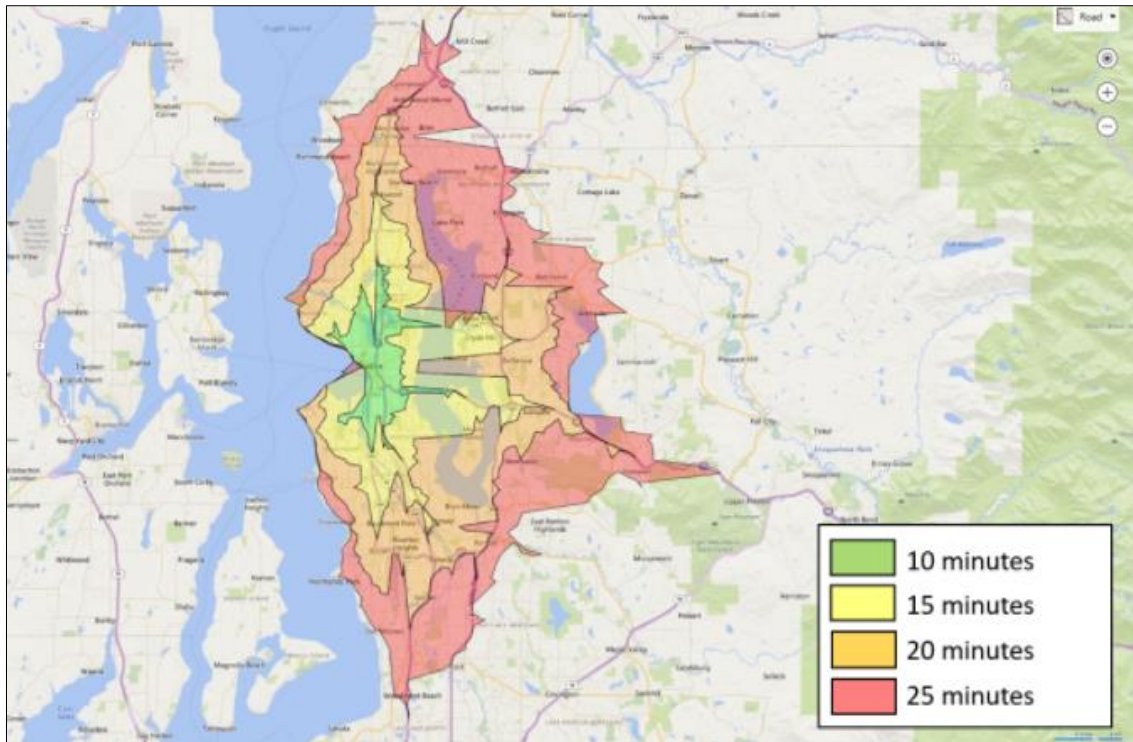


Figure 2.5 Bing's Isochrone API

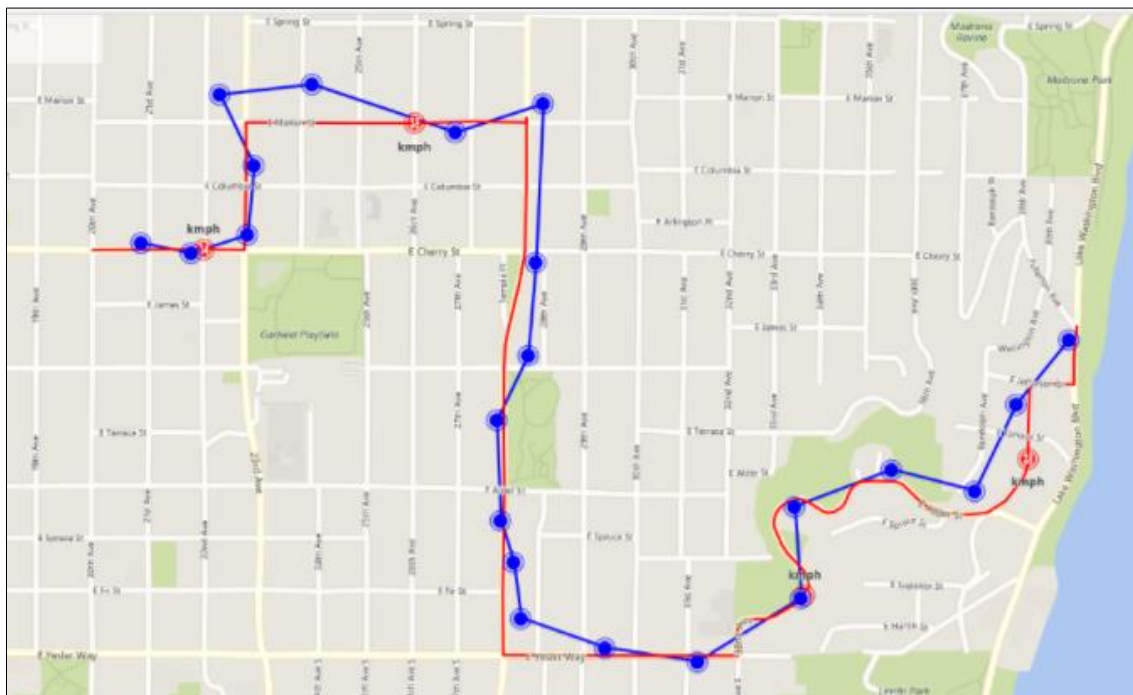


Figure 2.6 Bing's Snap-to-Road API

### 2.4.3 APPLE MAPS API

In the year of 2012, Apple announced their own rival service – Apple Maps which now become the default option in iPhone and iPads instead of using the map provided by Google. However, the Apple’s Mapkit is only available to Apple devices which code by using the Swift and Objective C language. In the post-iOS 6 world, developers now have the choices to embed map tiles and location service by using the Apple’s new MapKit. However, the data quality can be problematic to the extent that it fuels bad review on the Apps Store.

**Table 2.1 Comparison of features between maps provided by three company.**

<b>Attributes</b>	<b>Google Maps API</b>	<b>Bing Maps API</b>	<b>Apple Maps API</b>
Backend provider	Google	Microsoft	Apple Inc.
Degree of motion	Vertical, horizontal and depth rotation, 360 panoramic, 3D		
Dynamic Search result based on dragging on map	Yes	Yes	No
Weather	No	No	Yes
Send hyperlink to email	Yes	Yes	No
Business advertising	Yes	Yes	No
Historic traffic	Yes	No	No

## 2.5 REVIEW OF EXISTING SYSTEM

Bus transportation are quite common nowadays, but without a good management system, the bus services might not be fully utilized. In Malaysia, majority of the bus service company does not provide the real time bus tracking for the passenger. Instead, they only provide the scheduled timetable which are not very accurate and punctual. However, there is one private university in Malaysia had a real time bus tracking system implemented.

There are three systems has been studied in the middle of research. The functions, features and technologies used in those systems are explored so that it can be used to enhance the next generation of bus tracking system. The systems involved are Asia Pasic

University (APU) Bus Tracking System, UiTM Campus Bus Tracking System Using Arduino Based and Smartphone Application (UiTM Bus Checker) and IoT Based School Bus Tracking and Arrival Time Prediction System done by Amrita University India.

### **2.5.1 ASIA PASIFIC UNNIVERSITY (APU) BUS TRACKING SYSTEM**

Generally, the bus tracking system is developed by the Asia Pacific University (APU), a Malaysia premier private University. This system is a web-based system which embedded with the Google Map service to allow their students to check for the bus location status with their devices anytime, anywhere. There are bundle of bus icons distributed inside the Google Map which indicate that the location of buses in real time. The bus plate number is linked together with the bus icon to ensure that student get to the correct bus since different buses will have a different route. The system also provides the bus schedule at the right-hand side of the webpage showing the scheduled time of departure from three places which is APU, Faculty of Business & Management and South City Plaza (SCP) to certain popular area such as LRT station, academia, Fortune Park etc. The pages will automatically refresh every 10 seconds to renew the location of buses.

However, this system still having some weaknesses. The timetable provided is the scheduled time of departure from either APU Campus, APIIT@TPM or SCM. First, the system still unable to provide student with the actual arrival time of buses to the destination based on the location of buses in real time. The only advantage of this system is that student still able to know the location of buses that approaching certain bus stop base on the icon in the Google Map. Second, this webpage provides the plate number of buses without giving the information that the bus is going to which destination via which route. Students might confuse that a bus which approaching the pickup bus stop is the correct bus that he or she is looking for. Besides that, without the route information, student will get confuse that which buses to take if they decide to drop off at certain bus stop along the route.

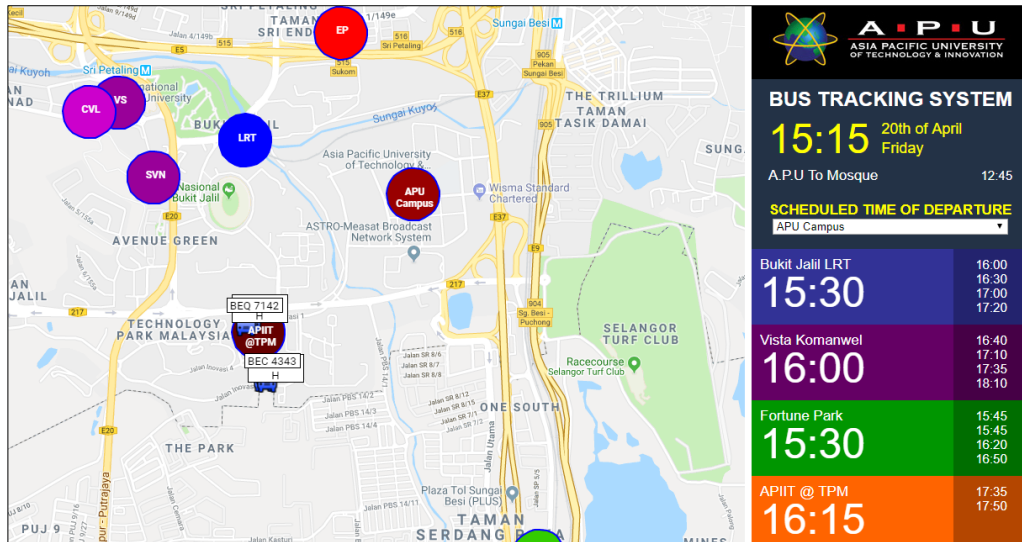


Figure 2.7 Asia Pacific University (APU) Bus Tracking System webpage.

## 2.5.2 UiTM CAMPUS BUS TRACKING SYSTEM USING ARDUINO BASED AND SMARTPHONE APPLICATION (UiTM BUS CHECKER)

Fundamentally, this system is operated by a GPS module together with the Arduino UNO which is attached to a bus. Firstly, the GPS must receive the coordinate data from the satellite in the form of latitude and longitude. The Arduino UNO act as a controller will convert the coordinate into the degree decimal format via coding process. The coordinate data will be sent to the local server and access by the application that has been installed in user smartphone through the IP address provided by the local server. The desired bus location will be displayed in the user smartphone with the help of Google Map service (M.T.Kamisan, 2017).

This system is designed to help people track the location of the bus. By doing so, UiTM students are encouraged to take UiTM buses as their daily life transport. The system will also help students to manage their time instead of rushing for the buses and prevent students from taking wrong bus route since the routing information are provided inside the application. However, 100% Wi-Fi coverage inside the university is required to make the system feasible since there is no cellular technology being used in the system. Extra hardware also needed which will increase the cost of the system. Besides, this is not a real time application because according to TechTarget, a real time is a level of computer responsiveness that user feels as sufficiently immediate or fast enough. Instead, in this mobile application, it only showing a static location of buses at particular time

unless user refresh the content manually. When users tap on the hyperlink of Google Map, the mobile application only showing the Google Map and mark the current location of bus according to the latitude and longitude showing to the user.

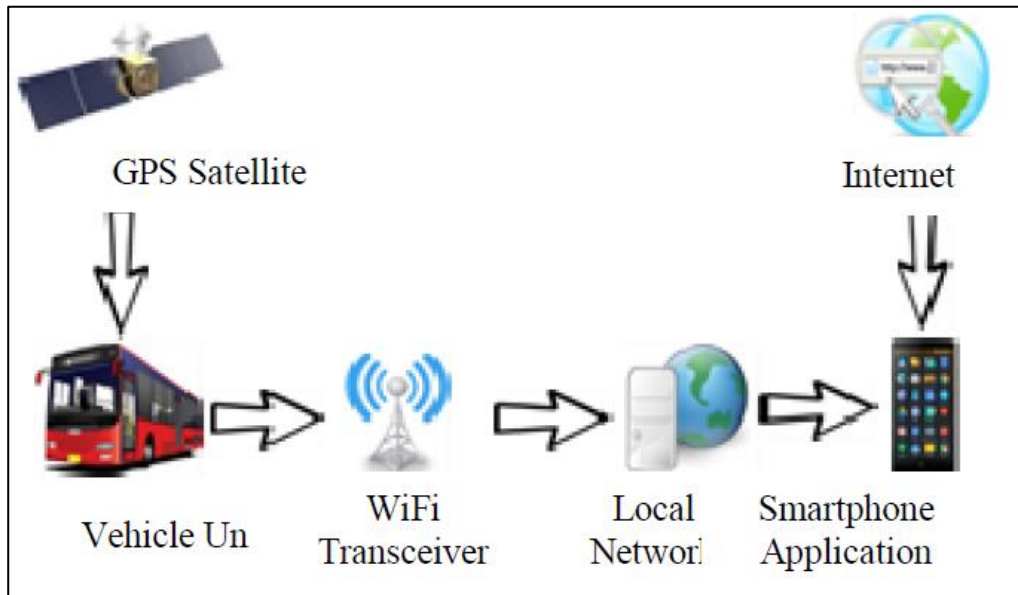


Figure 2.8 Overall system architecture of UiTM Bus Checker.



Figure 2.9 Overall UiTM Bus Tracking System

### **2.5.3 IoT BASED SCHOOL BUS TRACKING AND ARRIVAL TIME PREDICTION SYSTEM.**

There have been numerous of reports regarding kidnaping of school students on their way to school and delaying of school bus due to heavy traffic. Therefore, this is a system that use technology to safeguard the school children as well as provide tracking service for the school bus by using an IoT based monitoring system. By having the in-vehicle device, integrated with the Radio Frequency Identification (RFID) tag, identities of students on the buses can be identified. Besides that, the location of buses can be track by using the GPS module. A GSM module is also deployed to notify the parents about the entry and exit status of student as well as the status of the bus including the travel speed and location. If the predefined route schedule is altered, a short message will be sent to the parents as a notification.

The smartphone application created can provide the real time bus location on the Google Map to the end user by using data collected by the GPS module in the cloud server. All the data exchange is happening in real time through GPRS (Jisha R.C, Aiswarya Jyothindranath, & Sajitha Kumary L, 2017). The mobile application also tends to provide the route information and the expected arrival time of school bus by using a Kalman filtering-based prediction algorithm. Kalman filtering is a powerful method for prediction in real time system, since it estimates the state of system from measured value which include error (Nilah Ravi Nair & P. Sudheesh, , 2016). The research team also compares the expected arrival time as different point with the actual trip data to determine the effectiveness of the arrival time prediction by the application. It was found that the predicted bus arrival time had an error of 11.31%.

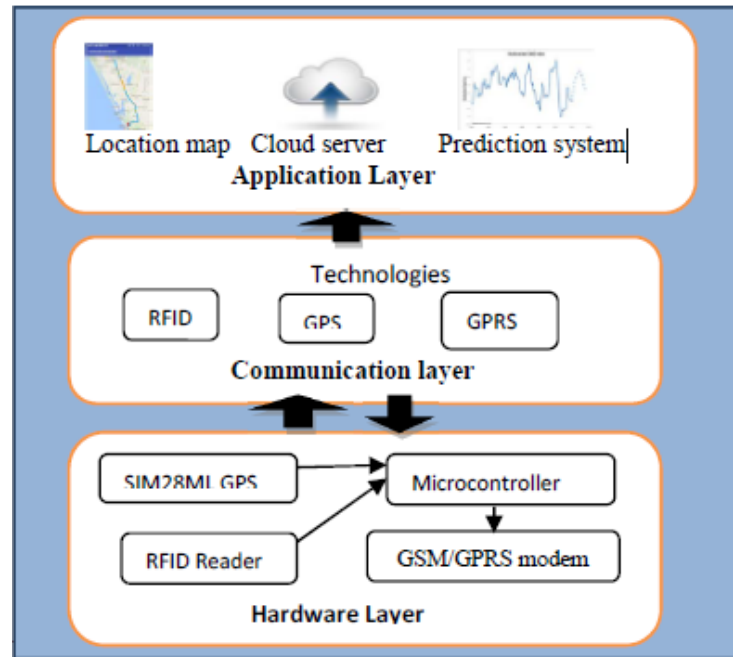


Figure 2.10 Overall system architecture diagram

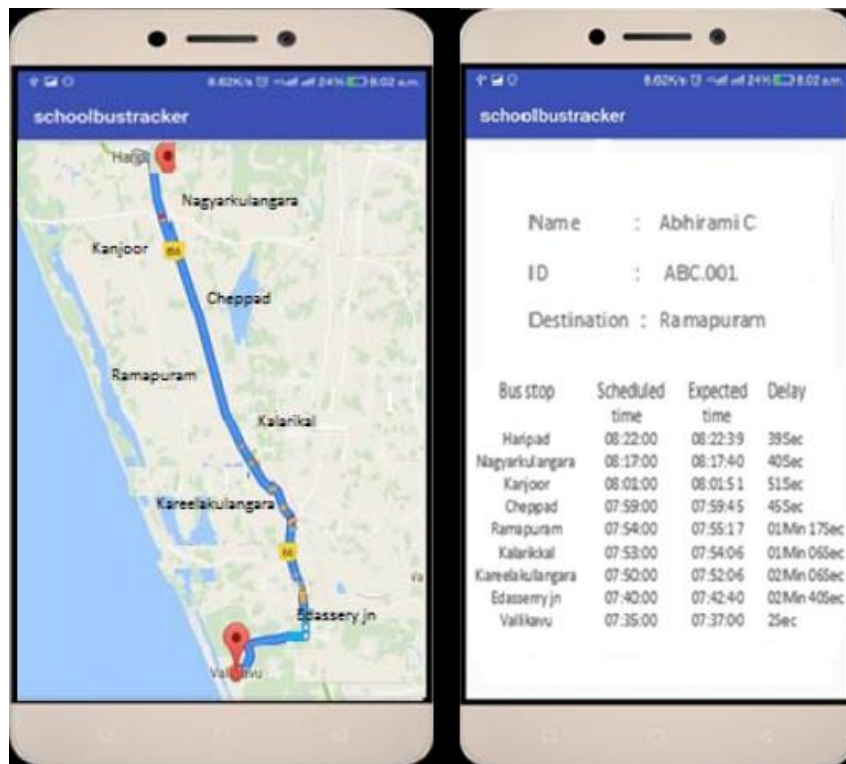


Figure 2.11 Travelling path of school bus, scheduled and predicted arrival time of bus.

## 2.5.4 COMPARISON OF EXISTING SYSTEM

Based on the research being made, there are three similar existing systems that have been developed by others. Table 2.1 shows the comparison of similar existing system in term of their features, advantages and disadvantage.

**Table 2.1 Comparison between existing system.**

<b>Features</b>	<b>Asia Pacific University (APU) Bus Tracking System</b>	<b>UiTM Campus Bus Tracking System Using Arduino Based and Smartphone Application (UiTM Bus Checker).</b>	<b>IoT Based School Bus Tracking and Arrival Time Prediction System</b>
Platform	Website	Mobile	Mobile
Technology involved	GPS & GSM	GPS, Wi-Fi	GPS & GSM, GPRS, RFID
Hardware	-	<ul style="list-style-type: none"> <li>• Arduino Uno</li> <li>• GPS UBlox NEO-7N</li> <li>• Wi-Fi Shield (ESPRESSO Lite V2.0)</li> </ul>	<ul style="list-style-type: none"> <li>• Arduino Mega 2560</li> <li>• EM 18 RFID reader</li> <li>• SIM 28 ML GPS receiver</li> <li>• SIM 900 (GSM/GPRS Modem)</li> </ul>
Advantages	<ul style="list-style-type: none"> <li>• Provide the most up-to-date location of buses with car plate in real time.</li> </ul>	<ul style="list-style-type: none"> <li>• The bus information is provided in detail.</li> </ul>	<ul style="list-style-type: none"> <li>• Using RFID to identify the identity of each passenger.</li> <li>• SMS to parent if the bus exceeds speed limit or the route is altered.</li> <li>• Provide the bus location in real time.</li> </ul>



			<ul style="list-style-type: none"> <li>• Provide the bus routing information as well.</li> <li>• Provide the scheduled and expected arrival time of buses in details for every bus stop.</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>• Not able to give the precise arrival time for each bus</li> <li>• Not able to show the route information.</li> </ul>	<ul style="list-style-type: none"> <li>• System does not provide estimated arrival time of buses.</li> <li>• System only show the location of one bus at one time when user click to open google map instead of real time tracking.</li> <li>• Required 100% Wi-Fi coverage within the campus to make the system feasible.</li> </ul>	<ul style="list-style-type: none"> <li>• Users have to manually enter the bus route number and date.</li> <li>• Cost of project increase due to extra hardware needed.</li> </ul>

## 2.6 Summary

After reviewing on various of technology, framework and existing system, this project will be implemented using Xamarin framework to code for the mobile application and integrated with Google Map Service and GPS technology to show the bus location and estimated arrival time to user. The application details will be highlighted in Chapter 3 and the implementation process will be discussed in Chapter 4.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 INTRODUCTION**

System Development Methodology (SDM) is a term or framework used to structured, plan and control the process of developing an information system. Choosing a suitable methodology is important because it provides step-by-step guide about the process that happens throughout the project development to meet the system requirement and effectively determine the success or failure of the project. In addition, this chapter also discuss the general architecture of the system and software used to make the system.

#### **3.2 RAPID APPLICATION DEVELOPMENT (RAD)**

In this project, a Rapid Application Development (RAD) is chosen to develop the system because RAD emphasizes on rapid prototype development, users' feedback and fast product delivery instead of strict planning and requirement gathering. RAD is actually the first development methodology to challenge traditional waterfall practices (Idesis, 2017). By using RAD, quick delivery of project can be achieved since the functioning part of a system which might be "good enough for now" can be developed as a prototype and showing to the client in a very short time. Client will evaluate and suggest for improvement and changes for either the interface or the functionality after first delivery. Client will actively involve in the project so that the entire development can meet their vision and requirement.

Besides that, RAD greatly reduce the risk and cost of project as compared to waterfall methodology. In waterfall, the time and cost spent on the features development can never be recovered if there are features being gut from final product or that particular

feature is technically not feasible. Therefore, as a student to develop the project in short time with lower budget provided, RAD could be the best development methodology to be choose.

There are four phases involved in the RAD which is requirements planning, user design, construction and cutover phase.

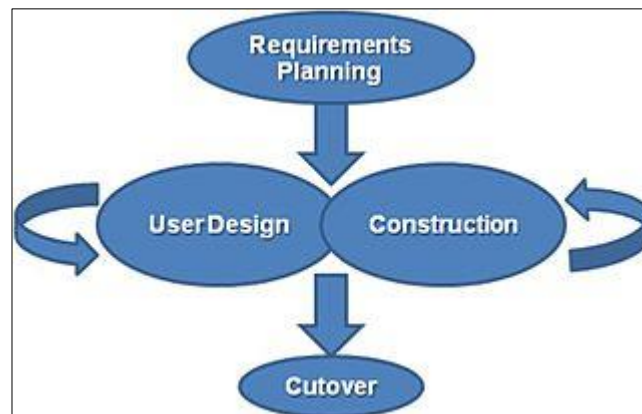


Figure 3.1 RAD phases.

### 3.2.1 REQUIREMENT PLANNING

During this phase, several meeting and discussion have been carried out with the supervisor for the proposed project. After the confirming the project title, all the system requirement is identified including the objective, problem statement as well the project scope. All the system needs are gathered from Automotive Engineering Centre (AEC) of Universiti Malaysia Pahang through the supervisor who as the intermediaries. Information such as how current public bus system work are gathered, project planning, project scope, framework going to use is discussed and the existing systems are reviewed during the meeting. The process ends up with all the requirements is agreed by the reviewers so that the project fulfil the client vision.

### 3.2.2 USER DESIGN

During this stage, the work plan and overall flow of system will be determined and well defining to prepare for the implementation. In Rapid Application Development, all the requirements included are only for the first prototype development in the first loop of RAD phases. After the client evaluated the first prototype and came out with some new requirements, development process will loop back to this stage and proceed with

second prototype design. The detailed system outline, model and implementation plan will be presented through few diagrams as stated below. This could be a documentation for the client as a reference to review and make adjustment for the next system requirement.

### 3.2.2.1 Use Case Diagram

Use case diagram is a Unified Modelling Language (UML) which use a simple chart to show the different ways that a user (actor) might react to the system. It summarizes the whole system in detail by using a set of specialize symbol. An include relationship indicates that the included use case is part of the base use case. A base use case could not be complete without included use case. Extended use case is the one which exists depend on base use case but is usually optional. Figure 3.2 showing how the user interact to the system. Bus company can perform a series of management function such as manage driver, route and bus plate information upon login to the management system. Bus driver will activate the route and provide most up to date location in real time to the users. User will be able to alert bus driver that he or she is waiting at particular bus stop, getting bus and route information as well as estimated arrival time to the bus stop.

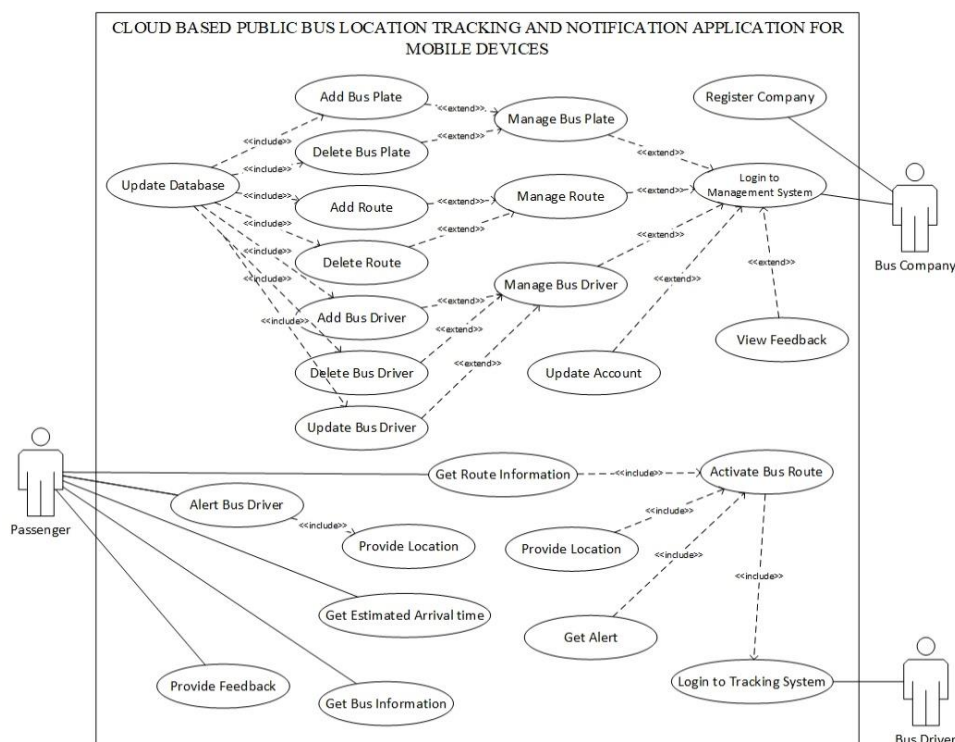


Figure 3.2 Use Case Diagram

### 3.2.2.2 Context Diagram

A context diagram also known as a level 0 Data Flow Diagram (DFD) which is a chart to show how the data is processed by a system in term of inputs and outputs. The diagram only contains one process represent the whole system together with the overall system data flow interaction with the external entities. Figure 3.3 below showing the data flow interaction between three users which is passengers, bus driver and bus company with the system.

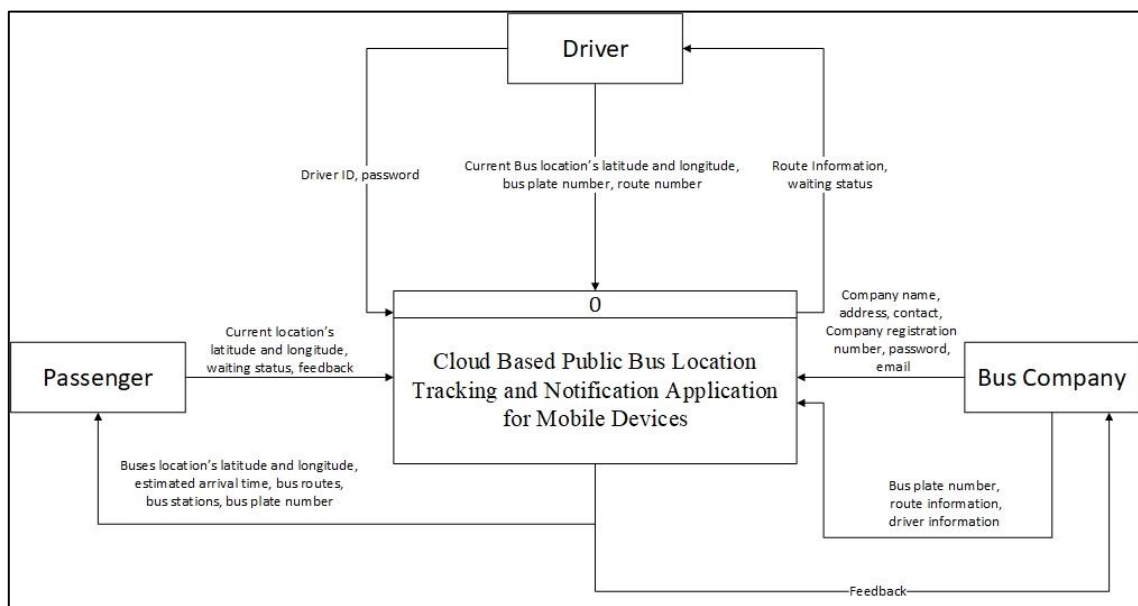


Figure 3.3 Context Diagram

### 3.2.2.3 General Flowchart of The Prototype

A flowchart is a diagram that describe the system algorithm or process by using a series of shapes to define the type of step along with the arrows to show the flow sequences. A rectangular shape represents the process, trapezium represents the data input or output and the diamond shape represents a decision point. Three separate flowcharts will be discussed to show general program flow of the BusSeeker application for user, driver and management system for public bus service provider.

#### 3.2.2.3.1 BusSeeker (Passenger Apps)

The application starts with gather user's location coordinate data from GPS and display the current user location in the Google Map. After getting user's location, system will start to search for the bus stops which located 3km away from user and compared

with the bus stop which already registered to the system. System will list out all the relevant bus stops and select the nearest one by default. However, user is still be able to choose other station if they wish to board on other stop or just check for the bus details. After selecting the bus stop, all the relevant route will be loaded from the database and again user may choose for the desire one. After selecting the bus stop and route, user can see the location of next bus on the map along with the estimated arrival time by tapping on the “Track” button. Location of bus in the form of latitude and longitude is getting from the database every 3.5 seconds interval. After tracking for the bus, user may choose to alert driver. A notification will be shown 3 minutes before the bus reach if user choose to do so. The bus location will continue to refresh, and user will be able to see the dynamic movement of the vehicle. Figure 3.4 show the overall program flow of user’s mobile tracking application.

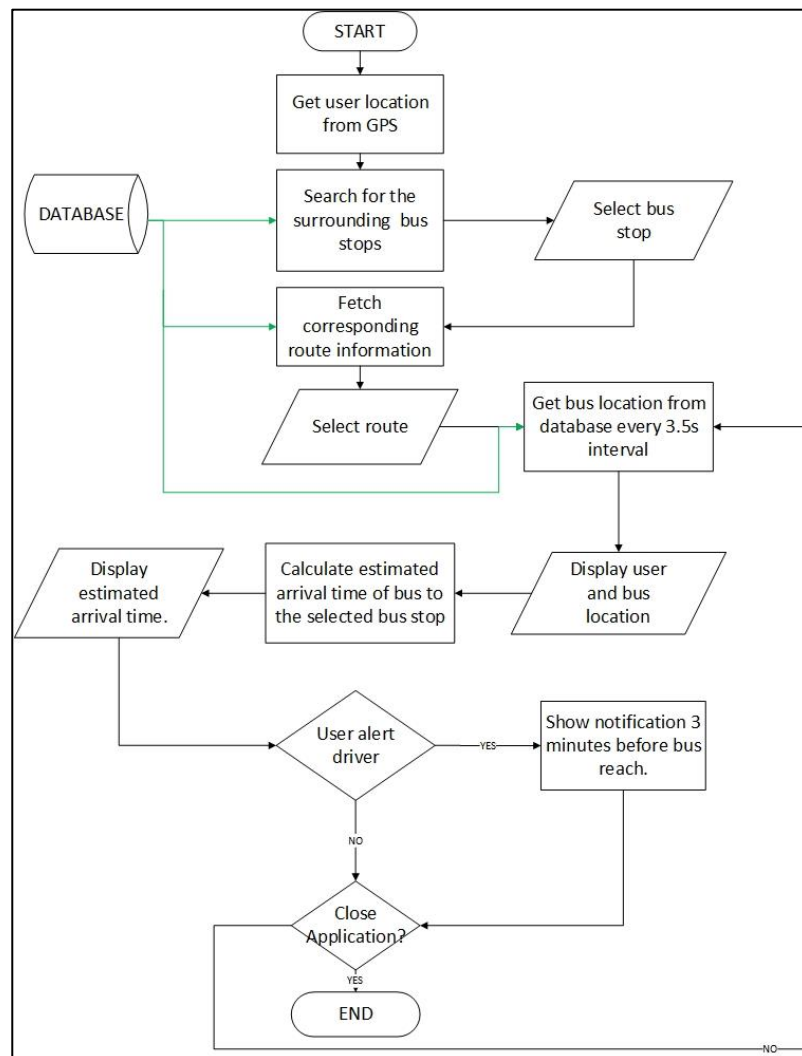


Figure 3.4 Flow Chart of BusSeeker (Passenger Apps)

### 3.2.2.3.2 BusSeeker (Driver Apps)

In order for the bus location to be actively displayed for the users, the bus driver must log into the system to register and activate the bus plate number with the respective route before the journey started. Driver will be able to see the current bus location along the route since the routing information will be downloaded from the database server once driver successfully activate the system. Bus location will also be sent to the server every 2 seconds interval so that user may get the update of bus location in real time. When the route is finish, the session will be ended, and the driver will be signed out from the mobile application.

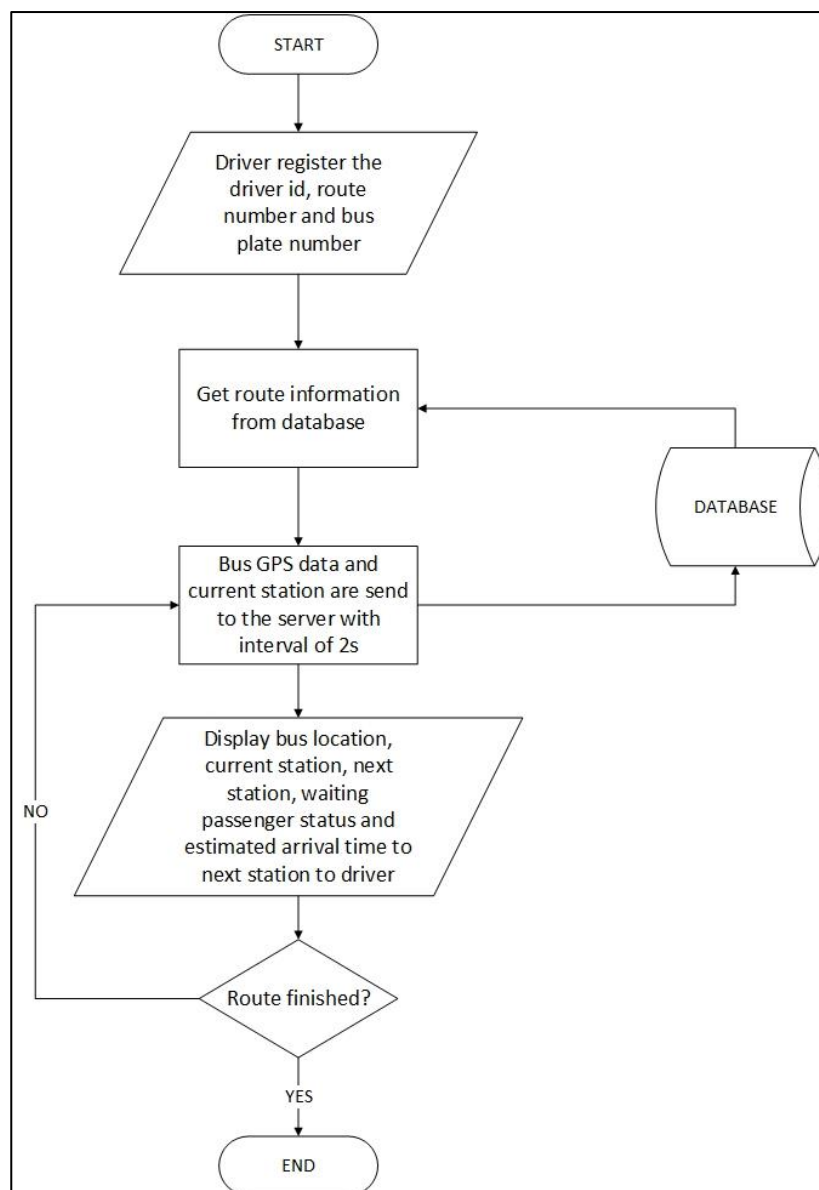


Figure 3.5 Flow Chart of BusSeeker (Driver Apps)

### 3.2.2.3.3 Bus Seeker Management System

The program flow starts with registration of company to the system. Every public bus service provider should provide their company information to the system for registration purpose. When a company successfully registered or log into the web-based application, they will have the authorities to modify their routing information, driver identity as well as bus registration and management. Also, bus company will be able to view all the feedback given by the user and respond to it.

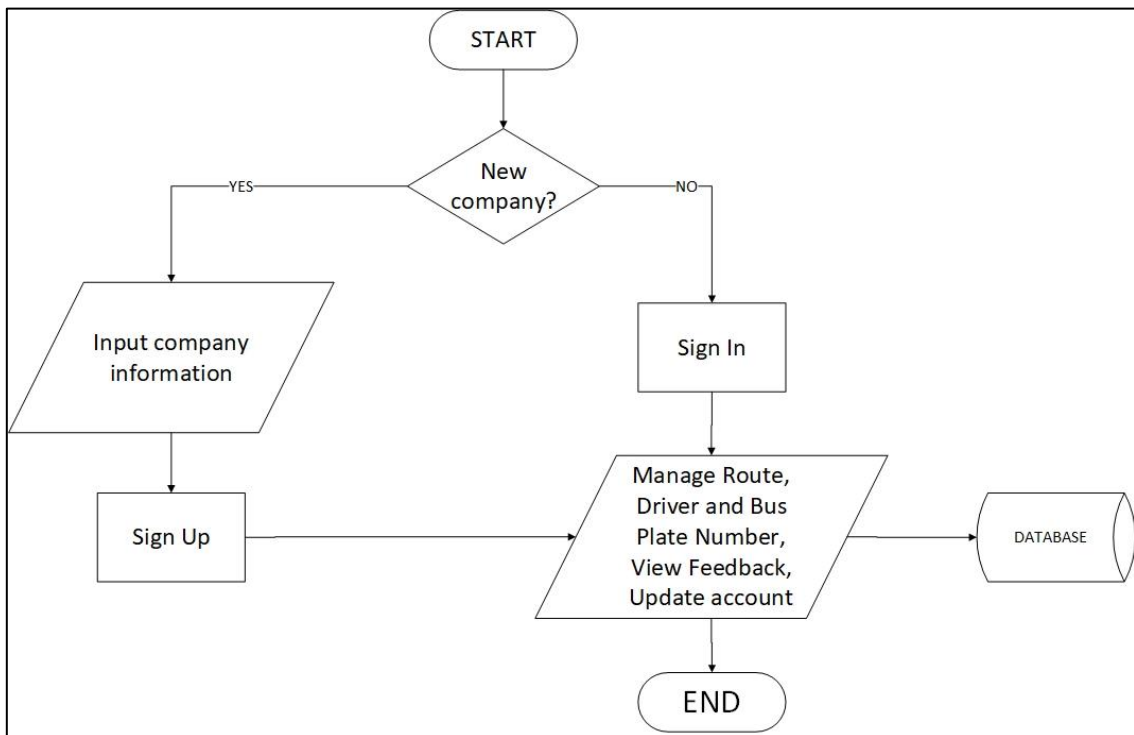


Figure 3.6 Flow Chart of BusSeeker Management System



### 3.2.2.4 USER INTERFACES

Figure 3.7 below shown the proposed user interface of the BusSeeker mobile application for the passenger. By default, when passenger open the application, system will automatically select the nearest bus stop away from user's current location. However, user still be able to choose the desire bus stop and route before search for the bus. After selecting the bus stop and route, user can get the location of next bus by tapping on the "Search Bus" button. The bus icon represents the location of next bus and estimated arrival time together with the bus plate number will be provided to user too. User can choose to notify the driver that he or she is waiting at the bus stop by tapping on the "Alert Driver" button. A notification will be shown 3 minutes before the bus reach. By tapping on the "mail" icon, user can provide any feedback to the selected bus company.

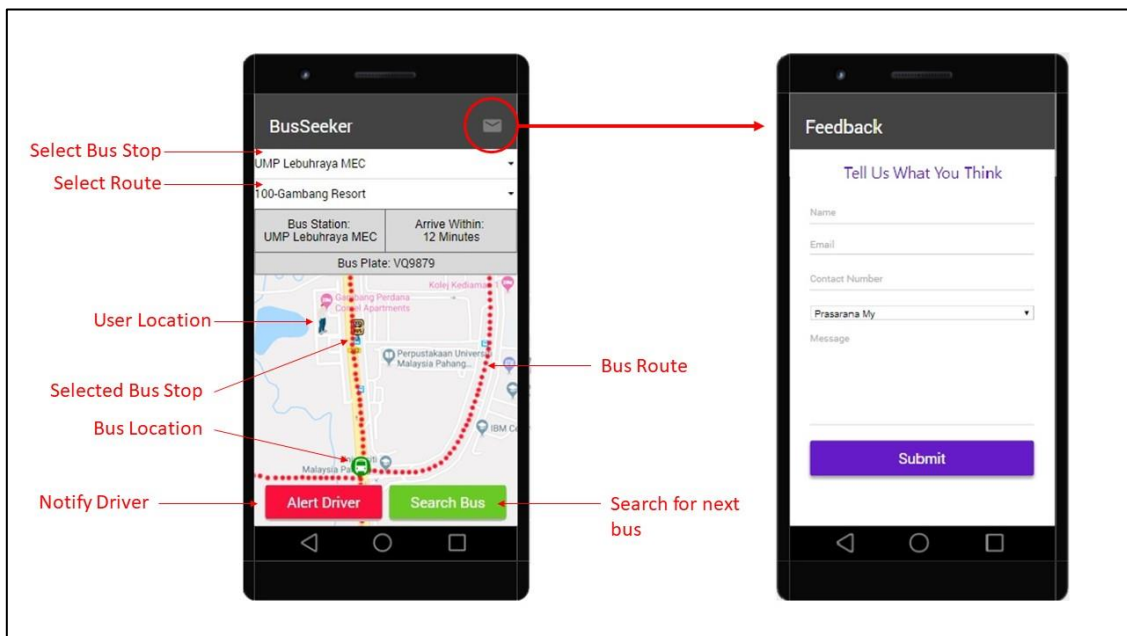


Figure 3.7 GUI of BusSeeker mobile apps for passengers.

A mobile application is created dedicated for driver to stream the bus location in real time. First, driver is required to login to the system by using the driver id and password provided by company. After that, driver is then required to activate the route by simple enter the route number together with the bus plate number. The bus plate number and the route number must belong to the same company as driver in order to activate the route. Upon successful activate the bus route, driver will be directed to the location tracking interface. The interface will show the route number, current station, next station, passengers waiting status as well as the estimated arrival time of the bus to the

next stop. System will automatically sign out once the route is finished or driver may choose to end the route manually by tapping on the top right menu button.

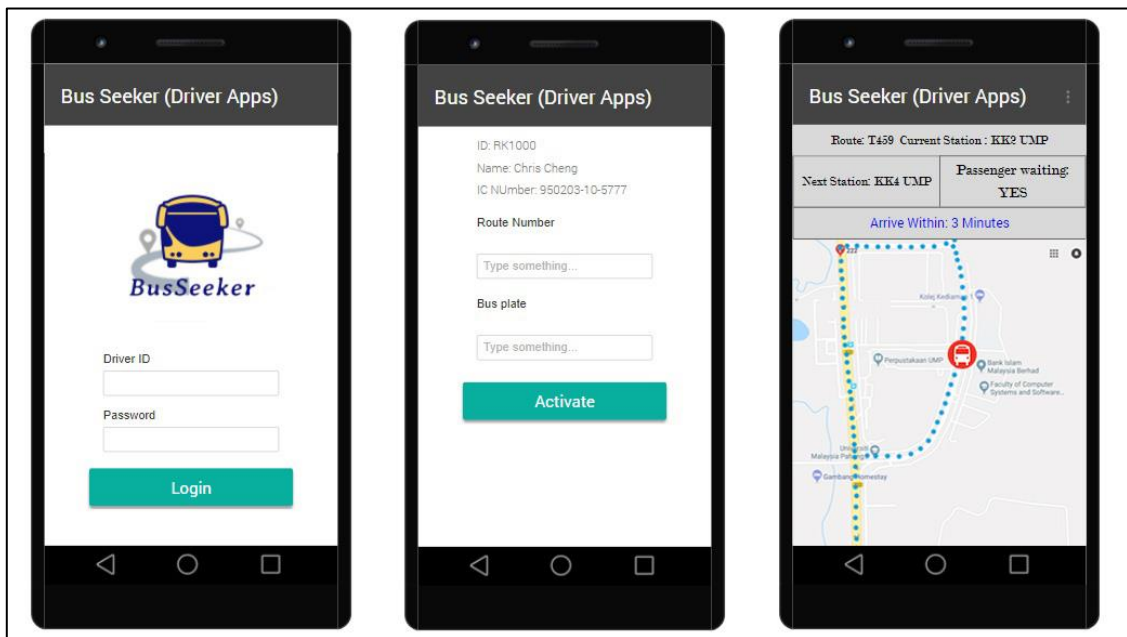



Figure 3.8 GUI of BusSeeker mobile apps for driver.

For the bus company, a web-based application is developed to allow company administrator to perform bus, route as well as driver management. First, the company should sign up to the system by providing all the basic information such as company id for login, company name, company registration number, contact number, email address and password as shown in Figure 3.9. After signed up, the company administrator may use the registered id and password to login to the system.



**Welcome, create your new account...**

Company ID

Company Name

Company Reg no.

Contact no.

Email

Password

Confirm Password

Figure 3.9 Sign up page

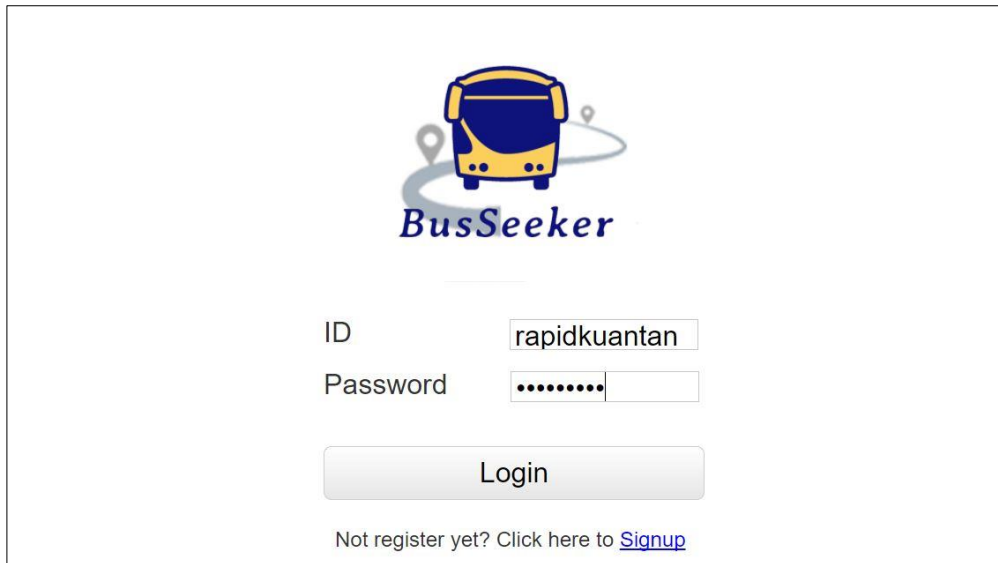


Figure 3.10 Login page.

A bus service provider can register the company buses to the system by simply fill in the bus plate number with the description. Buses that no longer working that can be eliminated from the system by clicking on the delete button.

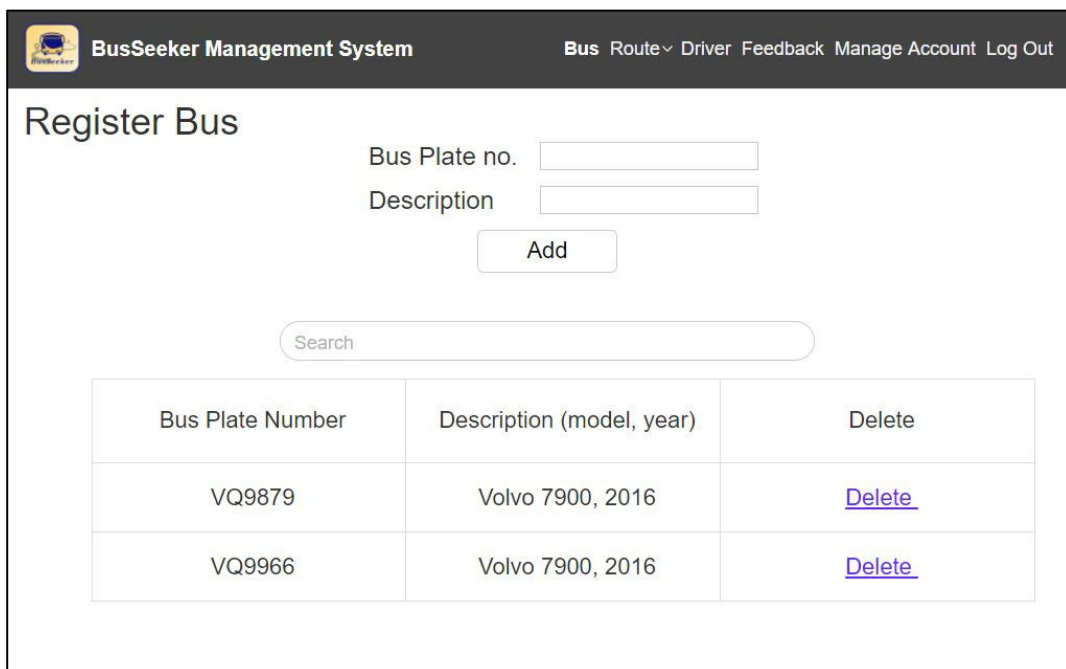
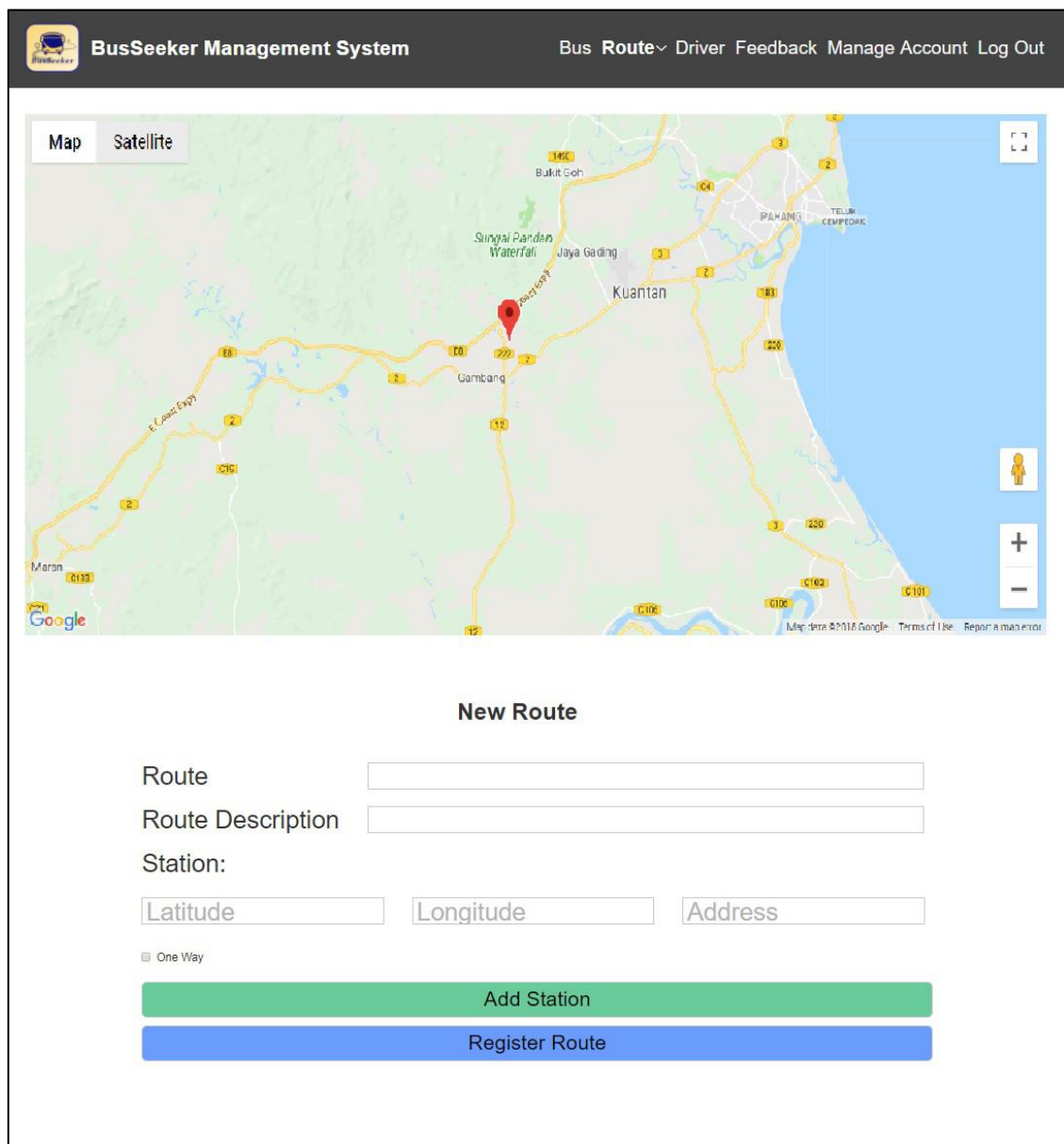


Figure 3.11 Bus adding and deleting page.

Another function of the management system is to bus company to register their route to the system. A bus provider may choose to add the bus stop to a route by clicking the location of bus stop on the map from the first bus stop to the end. A remove button is provided so that wrong input can be removed and once the adding process finished, administrator just click on the register button to finish the process. An administrator may also delete a route if the route is no longer exists as shown in Figure 3.13 or view the route details before delete as shown in Figure 3.14.



The screenshot displays the 'BusSeeker Management System' interface. At the top, there is a navigation bar with the system logo and the text 'BusSeeker Management System'. To the right of the logo, there are links for 'Bus Route', 'Driver', 'Feedback', 'Manage Account', and 'Log Out'. Below the navigation bar is a map of a region in Pahang, Malaysia, showing a yellow route line connecting several bus stops. A red pin is placed on the map, indicating a new station location. Below the map is a 'New Route' form with the following fields and buttons:

- Route:
- Route Description:
- Station:
  - Latitude:
  - Longitude:
  - Address:
- One Way
- 
- 

Figure 3.12 Adding bus route.

BusSeeker Management System Bus **Route** ▾ Driver Feedback Manage Account Log Out

### Route

Search

Route Number	Description	Detail	Delete
100	Rapid Kuantan	<a href="#">Detail</a>	<a href="#">Delete</a>
110	Rapid Kuantan	<a href="#">Detail</a>	<a href="#">Delete</a>

[Back](#)

Figure 3.13 Delete bus route.

BusSeeker Management System Bus **Route** ▾ Driver Feedback Manage Account Log Out

### Route : 122

One Way: Nope

No.	Station ID	Station Address
1	1090	Kolej Kediaman 2 (KK2), UMP
2	1091	Opposite Kompleks UMP Holdings, Kuantan
3	1082	Kolej Kediaman 2 , UMP

[Back](#)

Figure 3.14 Showing detail of each route.

A driver information can be recorded into the system after filled in the information such as name, IC number, contact number and email as shown in the Figure 3.15. The driver is required to report back to the company office and meet with the administrator to register for the account of the BusSeeker driver apps. This is because administrator can verify the information given by the driver so that not everyone can simply register for the account. The secrecy of password is still kept by the driver himself. Also, the information such as contact number and password still can be modified from time to time. Administrator have the right to delete the driver from the system too once the driver is no longer work with the company.

The screenshot displays the 'BusSeeker Management System' interface. At the top, there is a navigation bar with the system logo, the title 'BusSeeker Management System', and a menu with options: 'Bus Route', 'Driver', 'Feedback', 'Manage Account', and 'Log Out'. The main content area is titled 'Driver Registration' and contains a form with the following fields: 'Driver Name', 'Login ID', 'IC', 'Password', 'Contact no.', 'Confirm Password', and 'Email'. A 'Register' button is located below the form. Below the registration form, there is a section titled 'Driver' with a search input field. At the bottom, a table lists the registered drivers with columns for ID, Name, IC, Contact no., Email, and Delete / Update.

ID	Name	IC	Contact no.	Email	Delete / Update
RK1000	Chris Cheng	950828-01-5417	017-8585138	cc9555@gmail.com	<a href="#">Delete / Update</a>
RK1001	Mohd Ali	880606-04-5059	013-3562541	mali88@hotmail.com	<a href="#">Delete / Update</a>

Figure 3.15 Register driver to the system.

**BusSeeker Management System**      Bus Route ▾ Driver Feedback Manage Account Log Out

### Driver Personal Information Update

Driver Name: Alan Tan Li Qin      Login ID: alantan11

IC: 690101-04-5213      Old Password:

Contact no.: 016-6363215      New Password:

Email: alan9911@gmail.com      Confirm Password:

Figure 3.16 Update driver information.

The last function of the management system is to allow bus company administrator to view any feedback given by the user so that further action could be taken to improve the service. For example, user may request or recommend adding bus stop to certain route to the bus company so that bus service provider could respond base on feasibility. Also, if there are any problems related to the system, administrator can contact the developer for further improvement. Figure 3.17 shown how the message from user is conveyed to the bus service provider.

**BusSeeker Management System**      Bus Route ▾ Driver **Feedback** Manage Account Log Out

### User Response

Search:

Date	User	Email	Contact no.	Message	Delete
25/11/2018	Nurul Amalia	alia5050@gmail.com	012-6565123	Nice service, keep it up	<a href="#">Delete</a>
24/11/2018	Lim Ken Tak	eliphas@hotmail.com	018-2480234	Please improve accuracy	<a href="#">Delete</a>

Figure 3.17 User feedback

Lastly, the management system also allows the company to update their information from time to time, changing of password is part of it. A good practice of changing password helps to prevent account being used by hacker or any attacker. Figure 3.18 shown the interface for updating company information.

The screenshot shows the 'Company Details' section of the BusSeeker Management System. The interface includes a navigation bar with links for 'Bus Route', 'Driver', 'Feedback', 'Manage Account', and 'Log Out'. The main content area contains a form with the following fields:

Reg no.	Login ID
467220-U	myrapid
Company Name	Old Password
Prasarana MY	
Contact no.	New Password
09-5663686	
Email	Confirm Password
cnc@myrapid.com	

At the bottom of the form, there are two buttons: 'Update' (highlighted in blue) and 'Back'.

Figure 3.18 Update company information

### 3.2.3 CONSTRUCTION PHASE

This phase focus on system and application development. Tasks at this phase consists of coding, system programming and system testing of prototype. Client should continue participate in this prototype development phase and suggest for improvement. Any changes and requirement enhancement will be carried out by going back to the design phase once the suggestion is given by client.

General task during this phase are:

- Coding: In this system, C# would be the backend language for the BusSeeker mobile application for both passenger and driver. By using the Xamarin ecosystem, the user interface is creating by using the AXML language. The web-based application for the bus company would be code by using the PHP language as the backend language and the interface would be code by using the simple HTML5, CSS and JavaScript. All the interface design would be follow what have been done in the design phase.



- Testing: System testing will take place upon the coding phase is completed. The testing phase will determine whether the prototype created meet the requirement and objective or not. The database and the web-based application will be hosted in the Microsoft Azure cloud service. User acceptance testing will be carried out.

### **3.2.4 CUTOVER PHASE**

Cutover phase resembles the final tasks in the SDLC implementation phase, including data conversion, testing, changeover to new system and user training. After finalizing and completing testing the system, the system will be delivering to the client. Public bus service provider will subscript for the system by accessing to the web-based application and the drivers under the company will utilize the BusSeeker mobile application too. The passenger apps of BusSeeker will be published to the Google Play Store and install by the public bus user.

## **3.3 GENERAL ARCHITECTURE AND PACKAGE MODULES**

### **3.3.1 GENERAL ARCHITECTURE**

In this project, the proposed system consists of both mobile application and web-based application. Target user is the public bus passenger who will be able to track the location of public buses in real time. The system uses mobile cellular network to transmit the location data of buses obtained from the GPS satellite to the server from time to time and get access by the users. The web-based application is used by the bus service provided to register their buses with the respective routes. All the map, routing and estimated arrival time is calculated by using the service provided by the Google Map API.

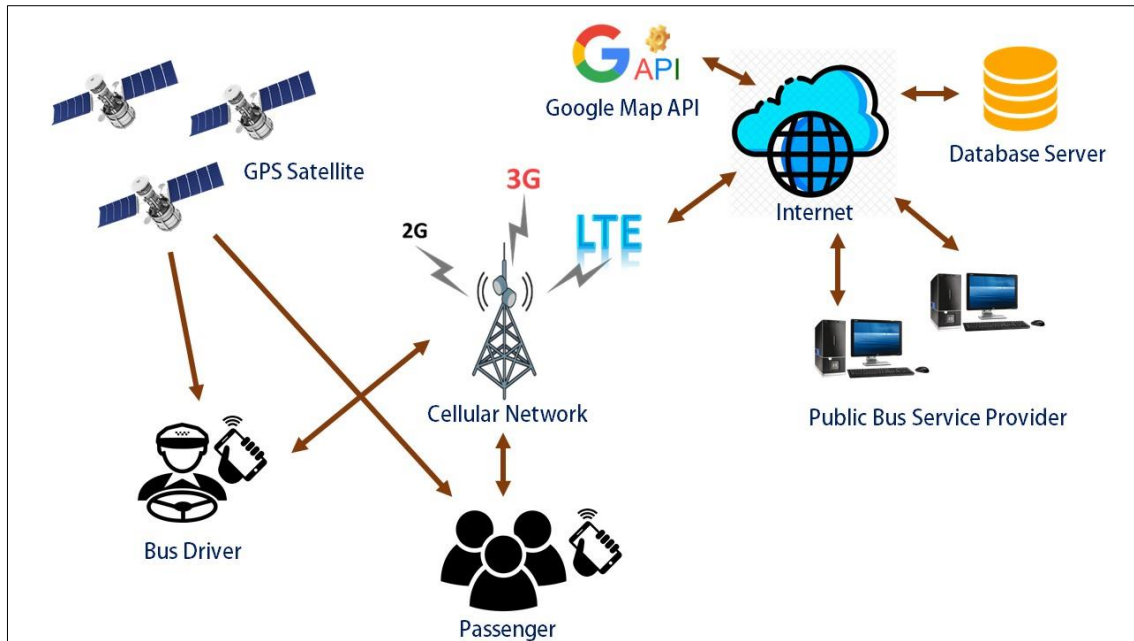


Figure 3.19 General architecture of the system.

### 3.3.2 PACKAGE MODULE

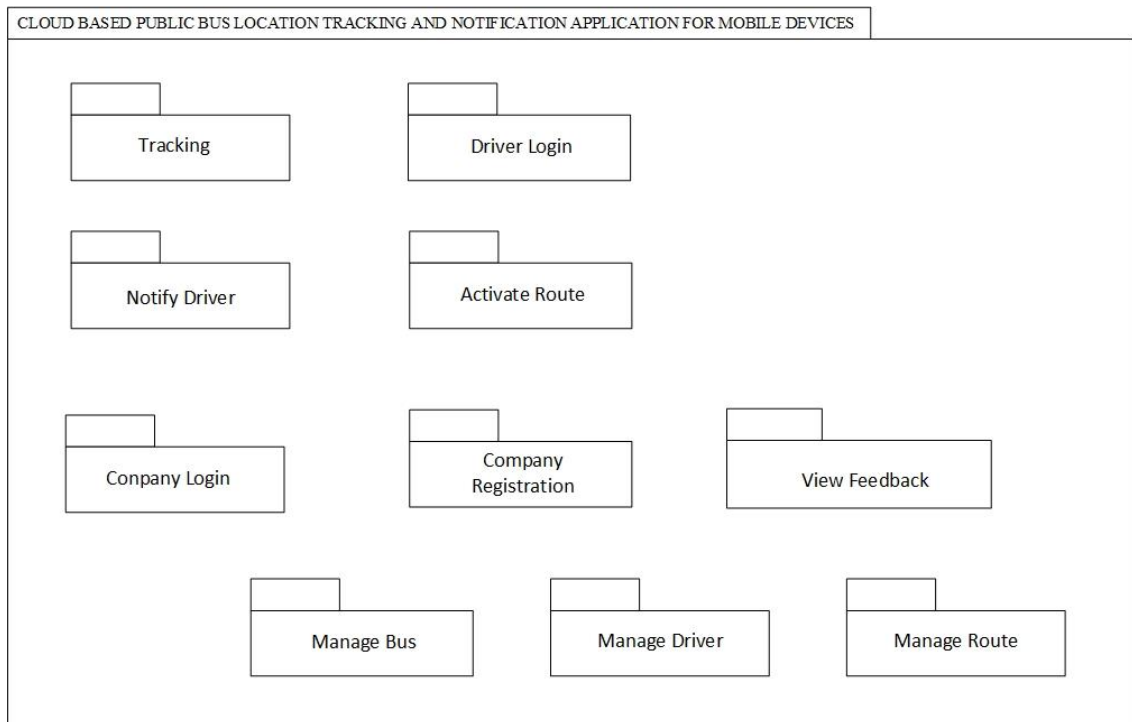


Figure 3.20 System package module.

### **3.3.2.1 Tracking Module**

This package is used to gather the location of public buses and display to the user. The information displayed to user included the bus route, bus plate number, driver information, estimated arrival time and the bus location in real time.

### **3.3.2.2 Notify Driver**

This package allows user to notify driver the location bus stop that he or she is waiting for the bus. Bus driver will get notified of the number of passenger that has been waiting for the bus.

### **3.3.2.3 Driver Login**

This package requests the driver to login to the system before start to use the system. This is to authorize the personnel who is going to stream the location. It could be used to verify the route and bus plate number that soon to be activate by driver. This package is important to give the bus identity to be shown on the map.

### **3.3.2.4 Activate Route**

This package provides the interface for the driver to activate the route before starting the journey. The activated route will navigate driver to the correct bus stop and allow the driver to start streaming the location.

### **3.3.2.5 Company Registration**

This package allows the public bus service provider who which to subscript to the system to register their company by simply enter the basic detail of the company.

### **3.3.2.6 Company Login**

This package allows company to login to the system by using the ID and password registered. Bus company login to the system to do some management stuff.

### 3.3.2.7 Manage Bus

This package allows public bus company to register all the buses under the company by simply providing the bus plate number and detail such as model and year or manufactured. Bus company may delete the bus too once the bus is no longer working.

### 3.3.2.8 Manage Route

This package provides the interface for the bus company to register the route. Administrator just have to click on the respective bus stop on the map to register the bus stop to the route. Bus company may also choose to delete route if the route is no longer exist.

### 3.3.2.9 Manage Driver

This package provides the interface for the company to register their driver to the system so that the driver could use the BusSeeker apps. Bus company may delete the driver information once the driver is no longer working for the company.

### 3.3.2.10 View Feedback

This package provides an interface for the bus company to view and manage any feedback given by the end user so that bus company can take further action to improve the service quality.

## 3.4 HARDWARE AND SOFTWARE

This subchapter will list down all the hardware and software requirement needed to develop the system.

### 3.4.1 HARDWARE REQUIREMENT

**Table 3.1 Hardware requirement**

<b>Hardwa re</b>	<b>Specification</b>	<b>Purpose</b>	<b>Qua ntity</b>
Laptop	<ul style="list-style-type: none"><li>• Windows 10</li><li>• 6<sup>th</sup> Gen Intel Core i5</li><li>• 8GB RAM</li></ul>	<ul style="list-style-type: none"><li>• To run the operating system.</li><li>• System documentation and design</li><li>• System development</li></ul>	1

	<ul style="list-style-type: none"> <li>• 256GB SSD</li> <li>• 1 TB HDD</li> </ul>	<ul style="list-style-type: none"> <li>• Connect to internet for system testing.</li> </ul>	
Smartphone	<ul style="list-style-type: none"> <li>• Android OS</li> <li>• 6GB RAM</li> <li>• HiSilicon Kirin 960 (CPU)</li> </ul>	<ul style="list-style-type: none"> <li>• Running the developed system for testing.</li> <li>• USB debugging</li> <li>• System demonstration</li> </ul>	2
Printer	HP 2520HC	<ul style="list-style-type: none"> <li>• Proposal and document printing.</li> </ul>	1

### 3.4.2 SOFTWARE REQUIREMENT

**Table 3.2 Software requirement**

Software	Purpose
Microsoft Windows 10 Operating System	<ul style="list-style-type: none"> <li>• Platform to develop, run and test the system</li> </ul>
Android Operating System	<ul style="list-style-type: none"> <li>• Platform to run the developed mobile application</li> </ul>
Microsoft Visual Studio 2017	<ul style="list-style-type: none"> <li>• Develop the mobile application</li> </ul>
Cloud Server – Microsoft Azure	<ul style="list-style-type: none"> <li>• To host the database and web-based application</li> </ul>
FileZilla Client	<ul style="list-style-type: none"> <li>• To upload the developed web application to the Microsoft Azure Service</li> </ul>
Notepad ++	<ul style="list-style-type: none"> <li>• To develop web-based application</li> <li>• To design and debug the system</li> </ul>
Microsoft Office 2016 <ul style="list-style-type: none"> <li>• Microsoft word</li> <li>• Microsoft PowerPoint</li> <li>• Microsoft Project</li> <li>• Microsoft Visio</li> </ul>	<ul style="list-style-type: none"> <li>• To prepare documentation</li> <li>• To prepare presentation slide</li> <li>• To create Gantt Chart</li> <li>• To make UML diagram</li> </ul>
JustInMind Prototyper 8.3.1	<ul style="list-style-type: none"> <li>• To design and propose prototype interface</li> </ul>

## **CHAPTER 4**

### **RESULTS AND DISCUSSION**

#### **4.1 INTRODUCTION**

The purpose of this chapter is mainly focus on the explanation about the implementation process of BusSeeker tracking system which include the passenger apps, driver apps and the web-based management system. The function of each interface and how background process works are briefly explained in this chapter. The system is fully tested to ensure that the system free from any error and functioning well.

#### **4.2 IMPLEMENTATION**

##### **4.2.1 TOOLS AND TECHNOLOGY**



Figure 4.1 Visual Studio

In this project, both the passenger apps and driver apps are developed using Visual Studio 2017 as the integrated development environment (IDE). Both passenger and driver apps are native android application which developed using the Xamarin framework with C# as the code base. There are various frameworks for cross platform app development, typically develop a website feel of native android app using HTML and JavaScript.

However, Xamarin is chosen because those kinds of framework normally run slower and does not give the feel of native android application because they are unable to access some of the native UI elements.



Figure 4.2 Microsoft Azure

For the BusSeeker management system, it's developed with traditional web technology with HTML, CSS and JavaScript as the front-end language and PHP as the server-side language. The management system is hosted to Microsoft Azure's App Service to be accessed by the public bus company. Microsoft Azure is a cloud computing service provided by Microsoft for organizations to build, test, deploy and manage their application through a global network.

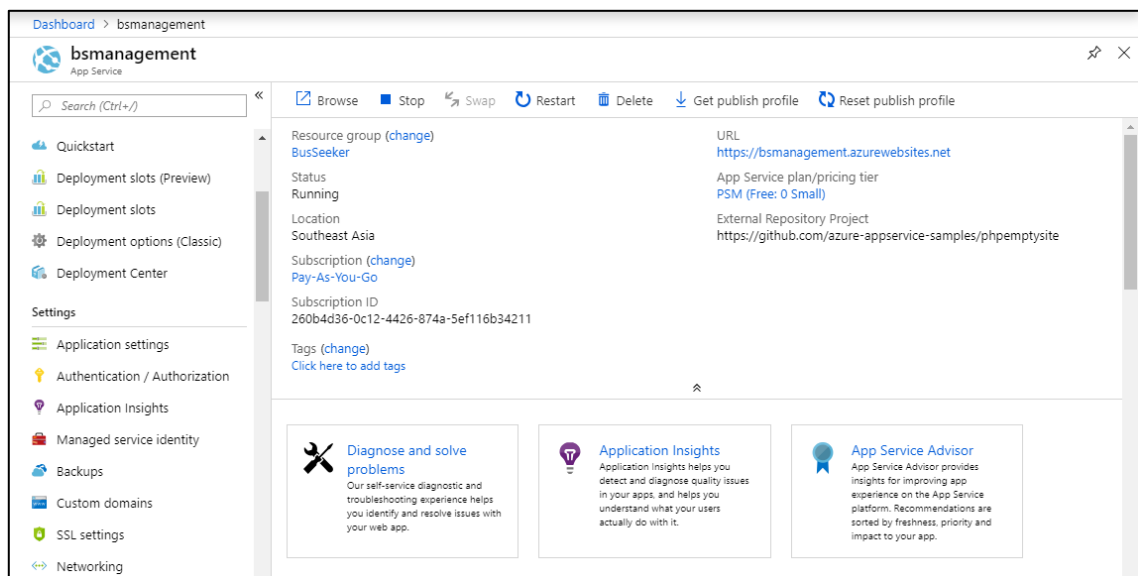


Figure 4.3 Microsoft Azure portal interface for app service.

From figure 4.3, the web application of BusSeeker management system would be hosted in the Microsoft's data centre which is located in the Southeast Asia region (Singapore). The bus company can simply access the website through the url:

<https://bsmanagement.azurewebsites.net>. For access control and security purpose, Azure cloud service allow developer to control the access of application by implementing the IP address restriction rules. This could be useful so that the developer can allow only the company who already pay for the subscription fee to use the system.

#### 4.2.2 DATABASE IMPLEMENTATION

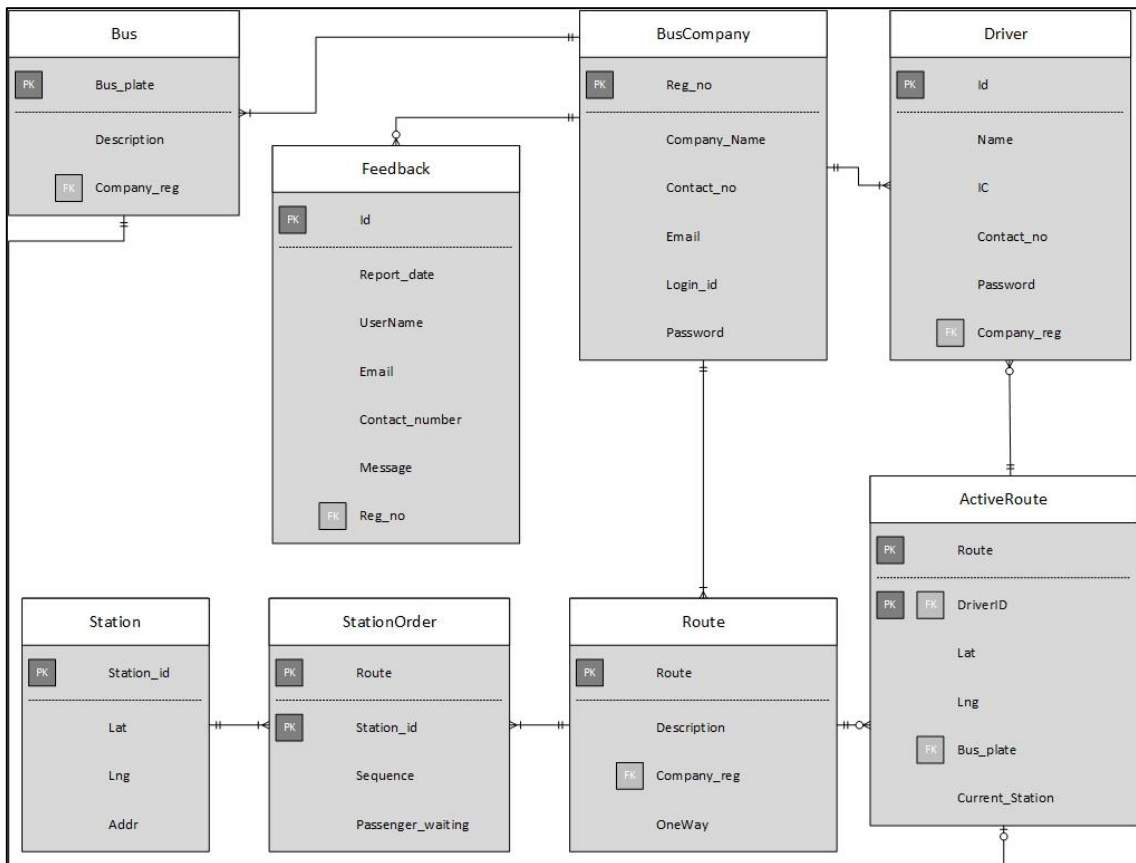


Figure 4.4 Entity Relationship Diagram (ERD)

A good database design is crucial for a high-performance application. In this project, the database plays a very important role to make the system functioning well. Again, the database is deployed to the fully managed and relational database-as-a-service (DBaaS) online SQL database provided by the Microsoft Azure Cloud Service. Figure 4.4 shown the Entity Relationship Diagram (ERD) which showing the attributes of each table and the relationship between all the table through crow's foot notation.



## 4.2.3 INTERFACE DESIGN DESCRIPTION

### 4.2.3.1 BusSeeker Driver Apps



Figure 4.5 Driver login interface

Figure 4.5 shown the first interface that the driver will see once open the application. Before starting a route, the application needs to verify and authenticate that only authorized personnel have the right to use the system. Busy driver id, non-existence driver ID or wrong password are prohibited to access the application.

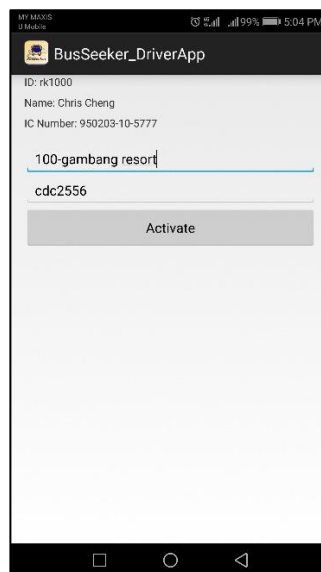


Figure 4.6 Route activation interface

Figure 4.6 shown the driver to activate certain route and provide location update afterward. After successful login to the application, the driver ID, name and Identity Card (IC) number will be shown to driver to indicate that driver successfully login with own ID. To active a route, driver need to manually input the route number with the bus plate number to the system, so that system can know the location of each bus and thus suggest to user. Driver will not be able to activate the route that not belong to the driver's belonging company. Also, driver will not be able to activate route with bus plate number which currently on the road.

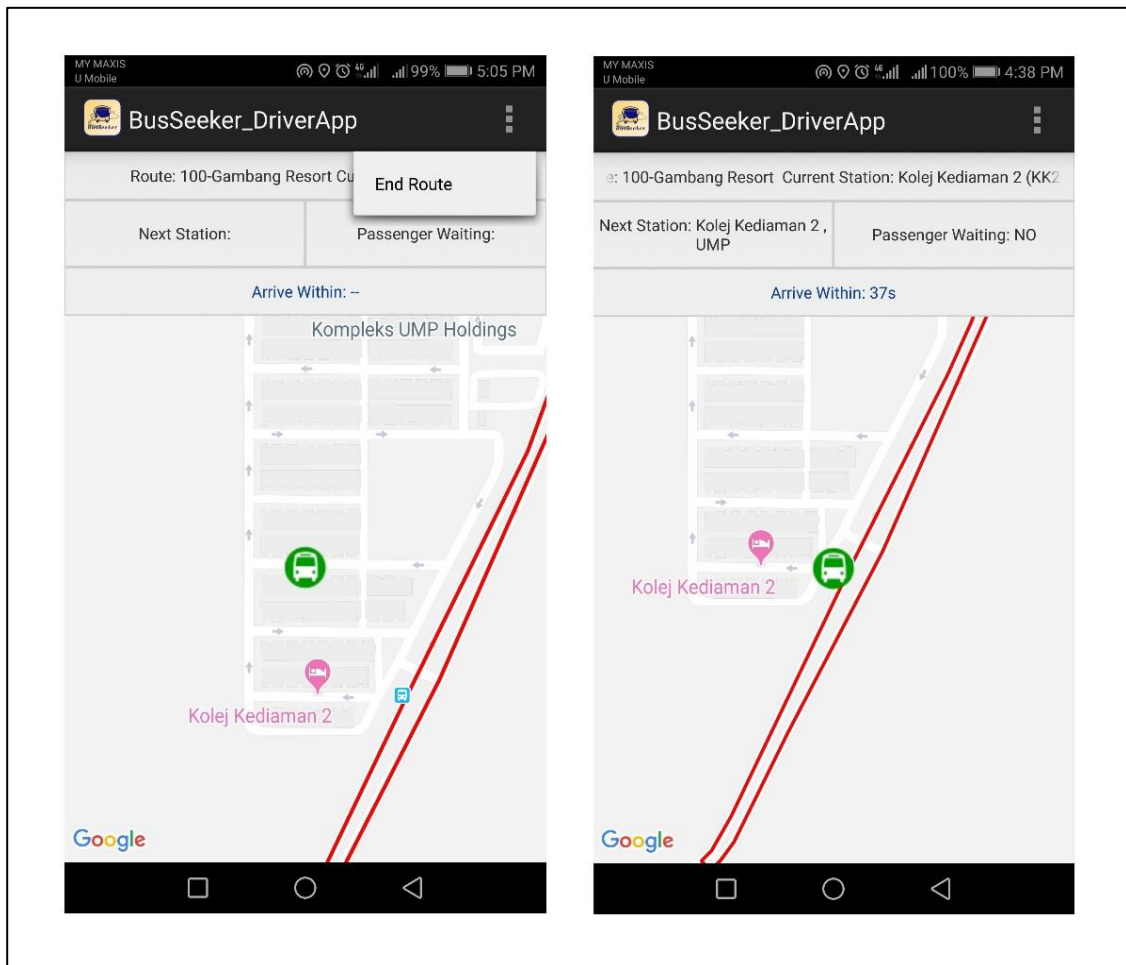


Figure 4.7 Tracking interface

After successfully activated the route, the driver will be navigated to the tracking interface as shown in figure 4.7. The background process started with searching for the starting point of the route based on the current location of the bus. Normally, the driver will start the journey at the first station of the activated route. However, there are some cases that the driver might start the route at intermediate station. Therefore, system will

search for the starting bus station by select the bus stop which located less than 30m away from driver's location with condition that the bus stop must be part of the route. The tracking function will be started after getting the starting point by constantly update the current location coordinate and current station ID to the active route table of the database. From figure 4.7 we can clearly see the different between the condition when the bus is far away from bus stop and the condition when the bus is ready to depart at the bus stop.

The system is set to update the location every second with minimum distance changes of 2 meters. Whenever the location is change, system will perform current and next station update when the bus is approaching next bus stop with the distance less than 30m. The change of current station will also be updated to the active route table in database. Furthermore, the estimate arrival time to next station will also being calculated through Google Distance Matric API and updated whenever the location had change. Lastly, the passenger waiting status will be change to "YES" if there is any passenger alert the driver at the bus stop. Every time the bus approaching a bus stop, system would assume that the passenger who wish to take the bus already on board and reset the waiting passenger status to "NO".

Whenever the bus reaches the last bus stop in non-looping route or first bus stop in looping route, system will prompt a dialog to force driver to end route and sign out from the application. Driver may also end a route manually by tapping on the "End Route" button. This is because the driver will never start another route right after reach the last bus stop. There will always having a gap between each schedule.

#### 4.2.3.2 BusSeeker Passenger Apps

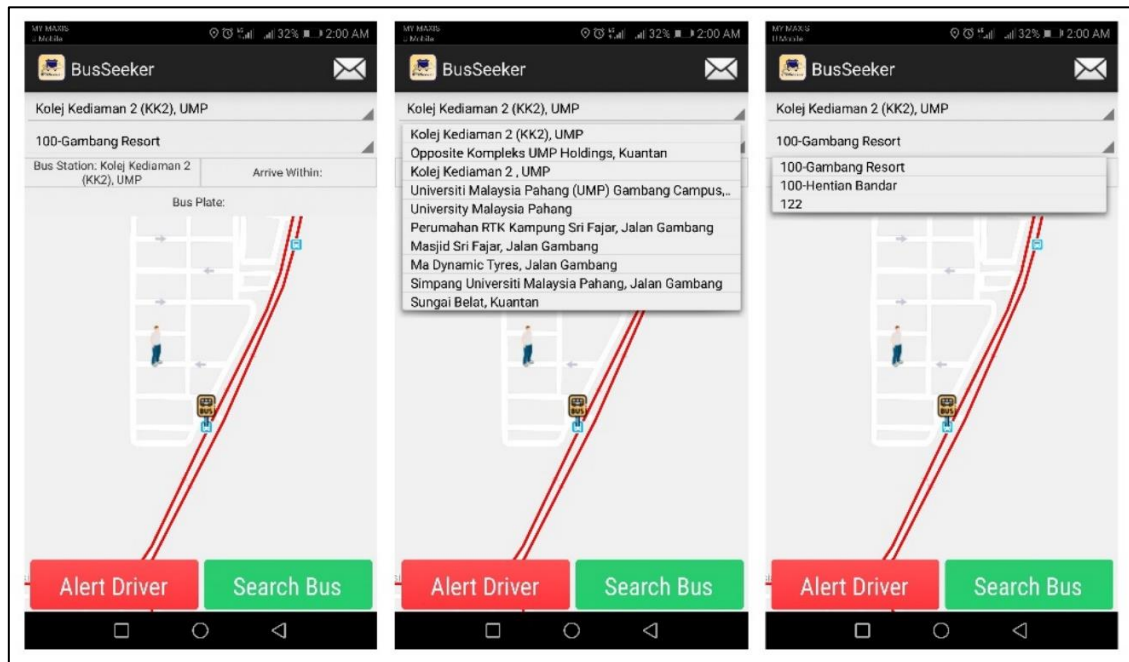


Figure 4.8 User interface of BusSeeker passenger apps.

Figure 4.8 shown the user interface that user will see once open the application. The first selection is the nearby bus stop and second selection is the route number. The application started by gather user's location coordinate and search for the nearby bus stop which located 3km away from user though Google Place API. By compared with the database, only the bus stop registered will be shown and selected by user. After user select the bus stop, relevant route will be shown and selected by user too. Selected bus stop will be marked on map and the route will be drawn on map too.

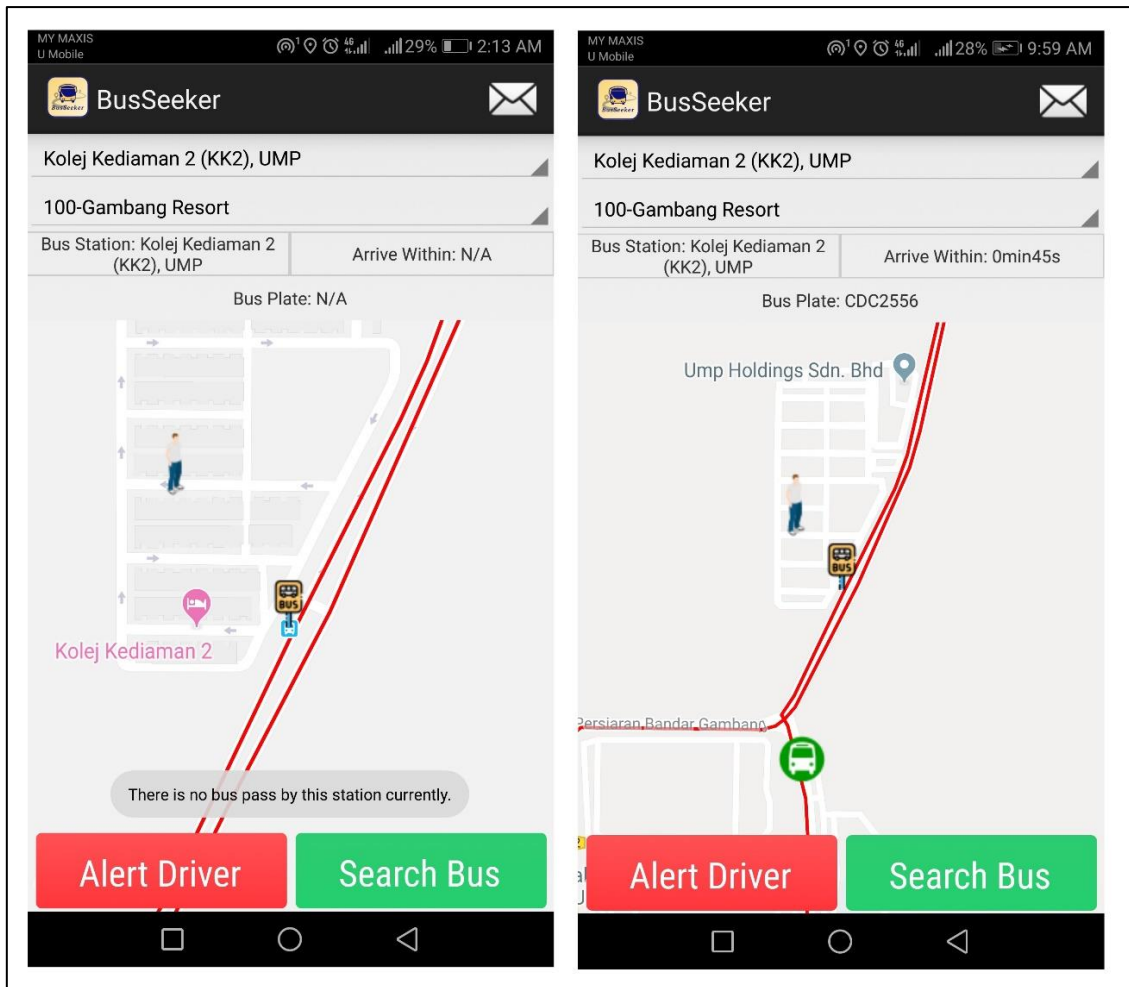


Figure 4.9 Search bus

When user tap on the “Search Bus” button after select the desired route, system will search for the next bus approaching to the bus stop where user currently having. System will inform the user if there is no bus with respective route pass by that bus stop. System will show the location of bus and mark on the map if there is any available bus as shown in figure 4.9. System will update the next bus location of selected route every 3 seconds of interval and suggest the estimated arrival time as well. Bus plate number will be provided to user as a reference so that user can catch the correct bus. At this moment, user will be able to click on the “Alert Driver” button and the passenger waiting status at the bus stop of selected route will become “YES”.

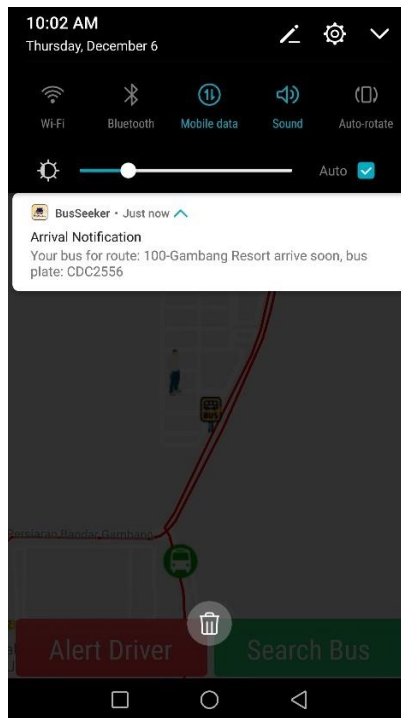


Figure 4.10 Notification

A notification will appear to inform user when the bus is nearly reaching the bus stop after user tap on the “Alert Driver” button as shown in figure 4.10.

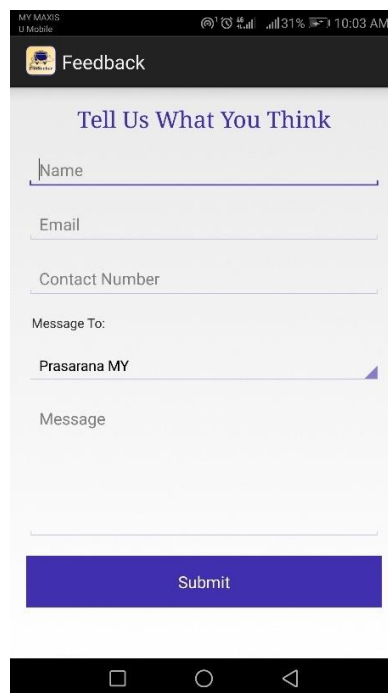


Figure 4.11 Feedback

Figure 4.11 shown the interface provided for user to give their comment to the service provided by public bus company so that the company can enhance their service. If there is any problem or error with the system, user might also provide their feedback to the bus company so that the bus company could reflect it to the developer.

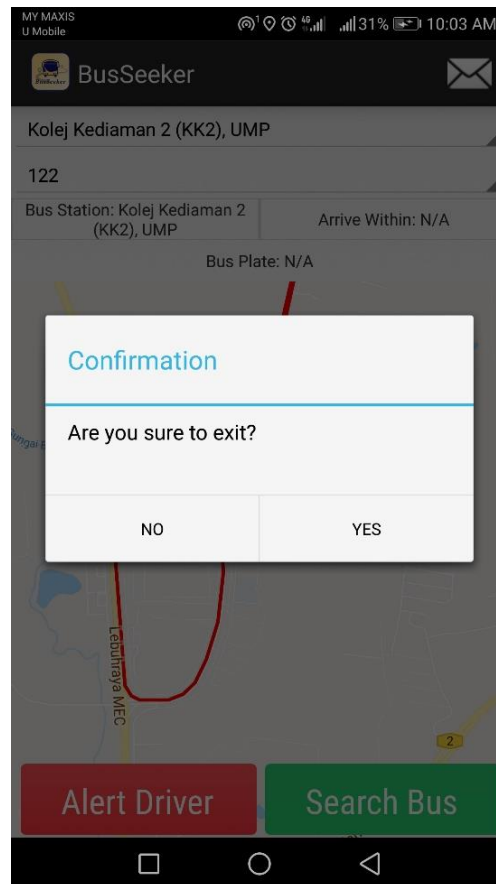


Figure 4.12 Exit confirmation dialog

A confirmation dialog will be prompted to user when user tap on the back button as shown in figure 4.12 to avoid unintentional action and causing the application to close.

### 4.2.3.3 BusSeeker Management System

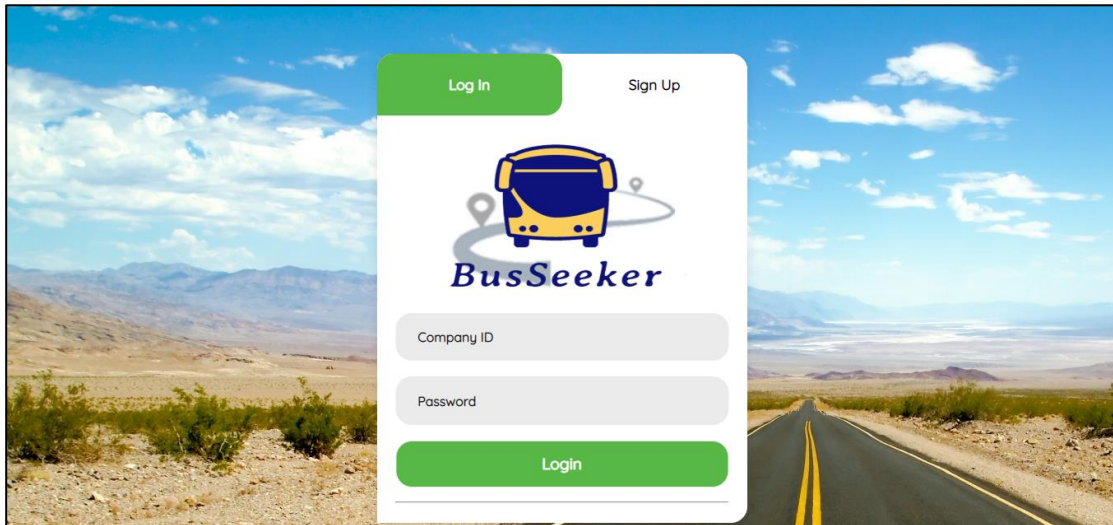


Figure 4.13 Management system login page

Figure 4.13 shown the interface design of BusSeeker Management System. This is the first page that will be loaded before using the system. Bus company are required to provide the ID and password to login to the system.

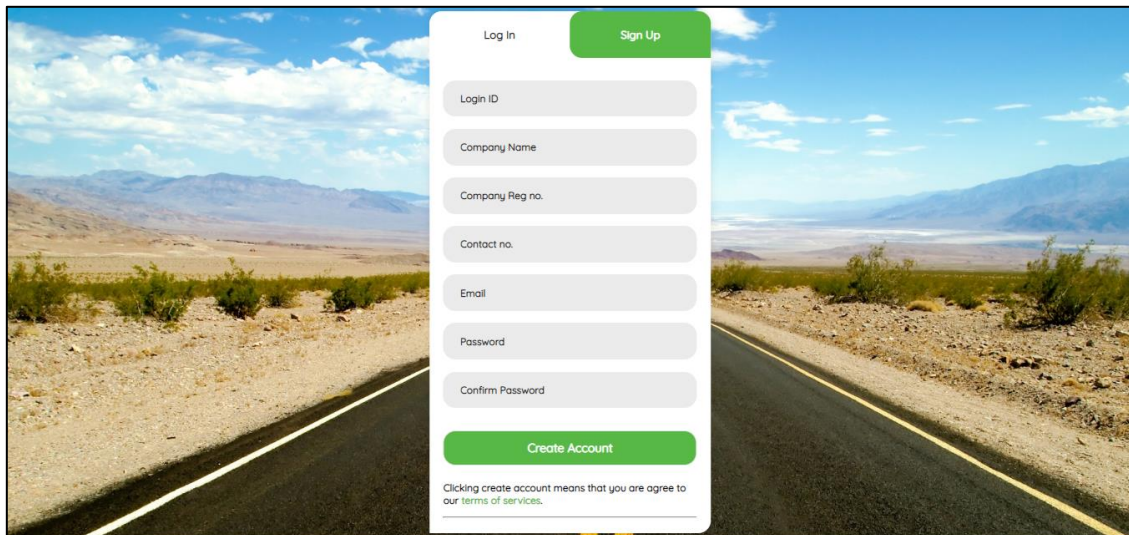


Figure 4.14 Management system signup page

For first time user, details such as login ID, company name, company registration number, contact number, email address as well as password need to provide to the system for sign up purpose as shown in figure 4.14. Any empty field or incorrect format of email address are prohibited, and the login ID must be unique too.



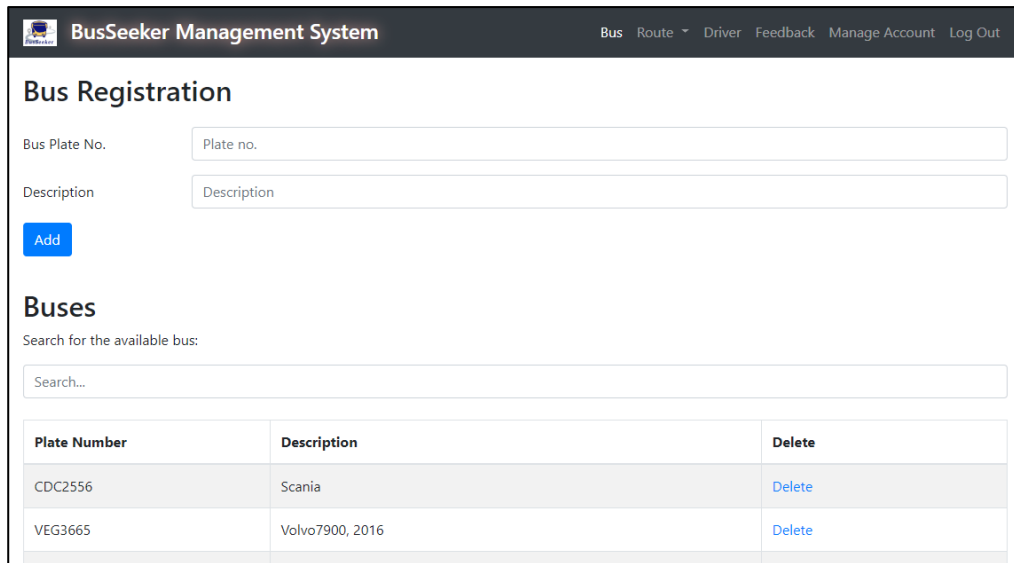


Figure 4.15 Bus management page

Figure 4.15 show the page where bus company can add their bus to the system by simply provide the bus plate number with the description. Registered bus will be shown at the lower part of the page and admin can search through it by typing any related keyword on the search bar. Unwanted bus can be removed from the system by clicking on the delete button.

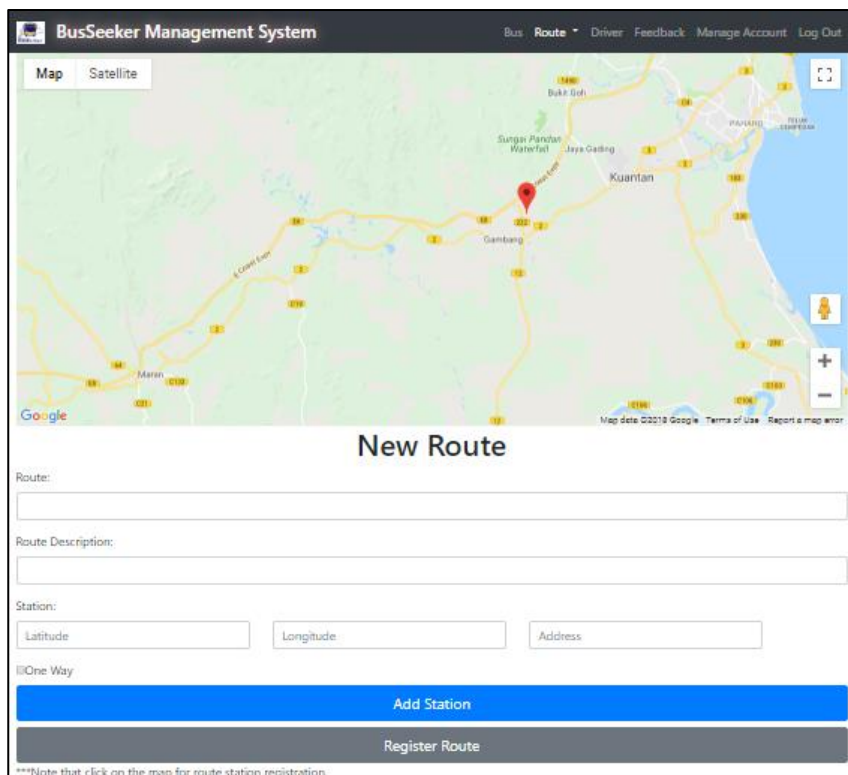


Figure 4.16 Route management page (add new route)

Figure 4.16 show the page for bus company admin to register their route to the system. Fill in the required field and click on the map to add station information to the latitude, longitude and address field. Admin can only add new station upon filling previous station information.

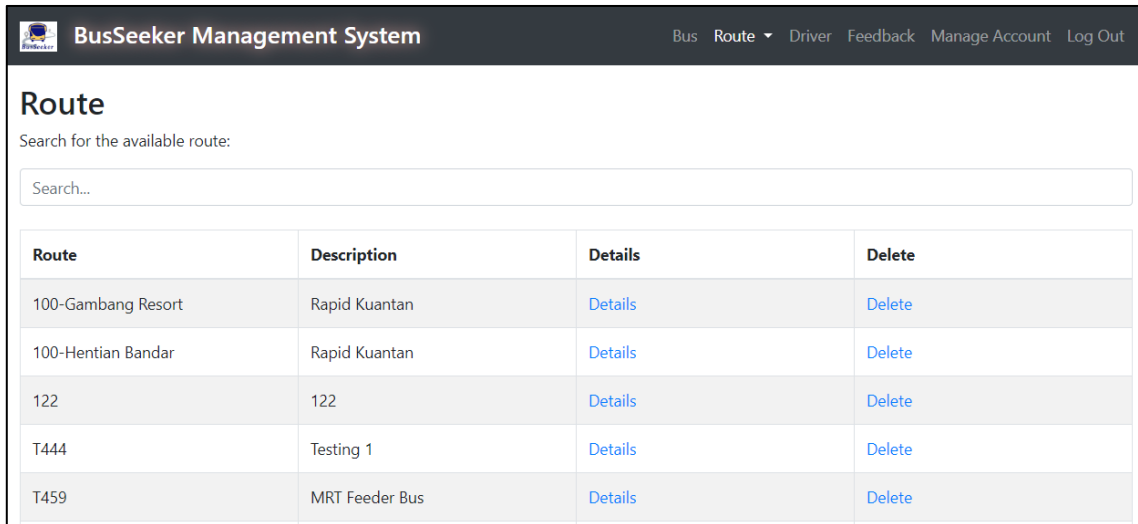


Figure 4.17 Search route

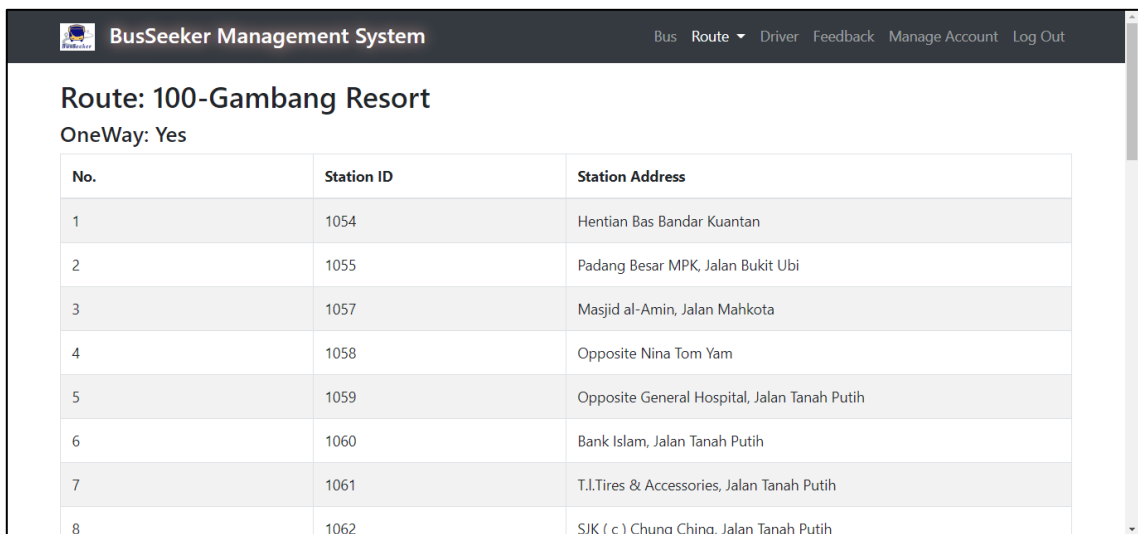


Figure 4.18 Route details

Figure 4.17 show the page for admin to manage the company route. Admin can search for the route by typing any keyword on the search bar and look at the detail of each route by clicking on the “Details” button. The details of the route with station information will be displayed to the admin as in figure 4.18. In addition, admin can delete unwanted route by clicking on the “Delete” button.

**Driver Registration**

Driver Name:  Login ID:

IC:  Password:

Contact no.:  Confirm Password:

Email:

[Register](#)

**Driver**

Search for the existing driver:

ID	Name	IC	Contact no.	Email	Delete/Update
arccus95	LIM GUO HONG	951104-04-5817	011-6868158	arccuslim@hotmail.com	<a href="#">Delete / Update</a>
M1000	CHEE XIAN YANG	950828-01-5413	017-6388158	xian-yang@hotmail.com	<a href="#">Delete / Update</a>
M1001	NORMAN YEW	690101-01-1119	016-1314520	norman69@yahoo.com	<a href="#">Delete / Update</a>

Figure 4.19 Driver management page.

**Driver Personal Information Update**

Driver Name:  Login ID:

IC:  Old Password:

Contact no.:  New Password:

Email:  Confirm Password:

[Update](#) [Back](#)

Figure 4.20 Update driver information

As what proposed, figure 4.19 is the page for admin to register the company driver by filling up the personal information. Admin might search for the driver to delete and update personal information. All the attribute can be changed except the driver name and login ID and IC number as shown in in figure 4.20.

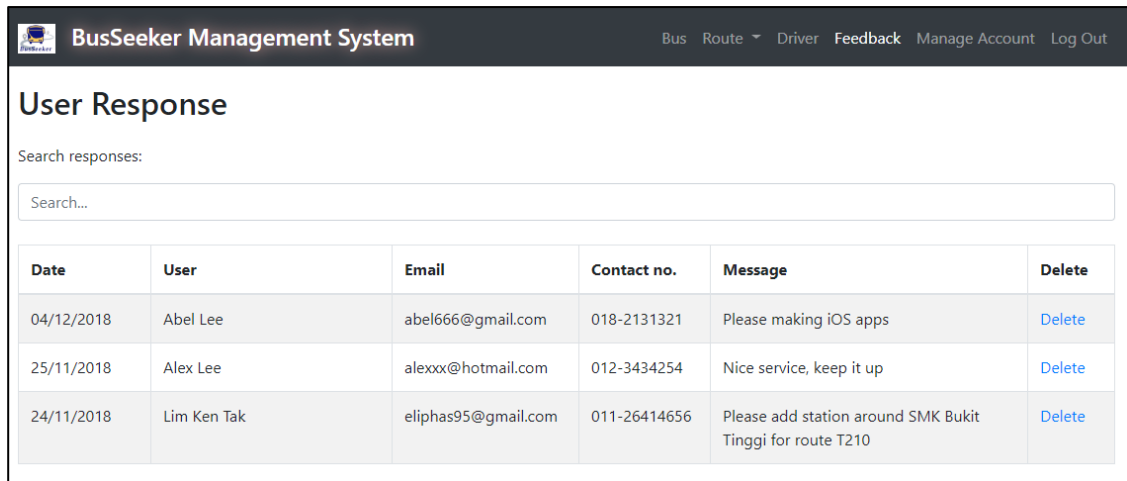


Figure 4.21 Feedback management page

Figure 4.21 is the interface designed for bus company admin to view the feedback from the passenger. Admin have the right to delete any irrelevant feedback from the system.

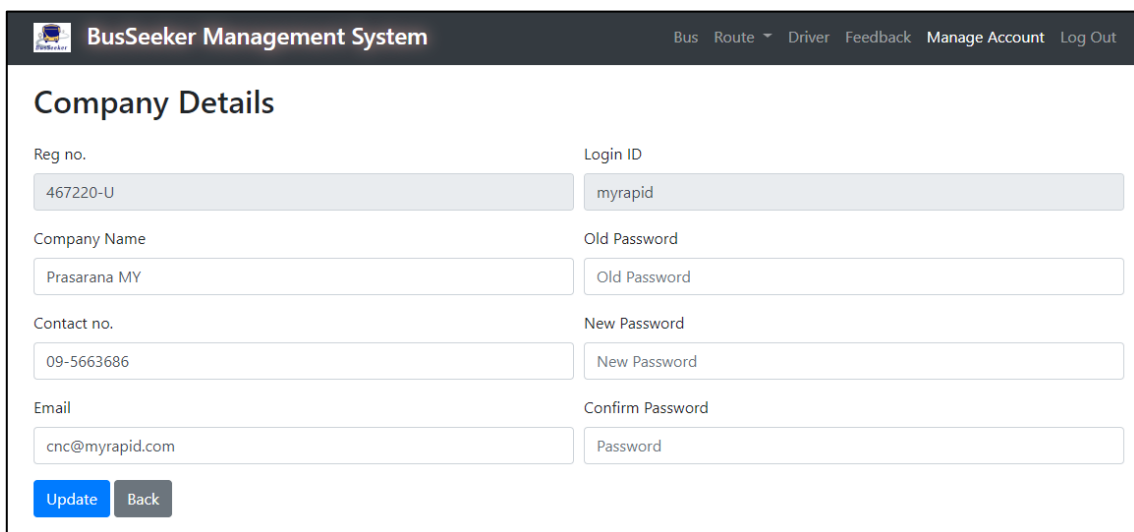


Figure 4.22 Update company information

Figure 4.22 is the interface designed for bus company admin to update the company information such as contact number or email address as well as update password for security purpose.

### 4.3 TESTING AND RESULT DISCUSSION

The project is undergoing a series of functional testing to ensure that the system developed is free from error, working well and able to handle task in real world scenarios.

The testing will use the outline of User Acceptance Test (UAT) to evaluate by the end user. UAT is usually the last phase of a software testing process before release the software or system. Refer Appendix B for User Acceptance Test (UAT).

#### **4.4 USER MANUAL**

Refer Appendix C for user manual.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 INTRODUCTION**

This project was created and aimed to encourage more people to take public bus transportation on daily basis to reduce the congestion as well as create a healthier and greener environment by increase the fuel efficiency and reduce carbon monoxide emission. In Malaysia, the public bus service is not fully utilized by the citizens due to several reasons such as unpunctuality, long waiting time and no direct access. Therefore, this project achieves its objectives by producing 2 mobile applications which connect passenger and driver for bus tracking purpose. By doing so, user can know the location of next bus and thus plan their journey earlier by referring the estimated arrival time instead of spending time waiting at the bus stop. A web-based management system also created for company to integrate their service to the system.

As what proposed in Chapter 3, the prototypes of the system are successfully implemented and delivered through Rapid Application Development (RAD). The prototypes are developed within the time stated in the Gantt chart and follow the system requirements, flow and overall architecture proposed in Chapter 3. After implementation, the prototypes are fully tested by the users through User Acceptance Test (UAT) to prevent any error happening.

## 5.2 PROJECT CONSTRAINTS

There are several constraints while implementing the project due to lack of resources, time and experience. The constraints are as follow:

- I. This system required bus stop data to functioning well. All the bus stop including the one registered to database and the suggested surrounding bus stops are based on the data returned by Google Map API. Therefore, this system only available for certain state in Malaysia such as Penang, Perak, Selangor and Pahang.
- II. Due to the cost issue, the Google API key used in this project are based on standard usage limit which will have certain request limit implied to the API usage. Also, the estimated time returned by the Google Distance Matrix API and Google Direction API are based on the average historical data instead of real-time traffic data. However, the estimated time is already accurate enough even it's not the real-time traffic data and shouldn't be a serious problem because the app will constantly refresh the estimated arrival time.
- III. Another constraint is regarding the network issue. This system communicates with the SQL database provided by Microsoft Azure Cloud Service through the port 1433 which is the port officially used by Microsoft SQL Server with TCP/UDP as the protocol. Some organization network or firewall might block the port to prevent spread of trojan. For example, UMP network is unable to access the database of the system, instead personal 4G cellular data is needed in order for the system to functioning well.
- IV. This system has own management system which required public bus service provider to register their route to the system. However, the bus company might also have their own database storing route data. Therefore, if possible external script has to be written for data migration especially for large company like Rapid Bus to increase the efficiency use of the system instead of insert data manually.

### **5.3 FUTURE WORK**

The BusSeeker mobile application is currently developed for mobile phone with Android operating system only. It might not benefit to the iPhone user which running on iOS. Therefore, a future work is needed to for making BusSeeker application available on iOS devices to benefit the Apple user. Besides that, a push notification should be done in future to notify the user even the app is killed instead of using a location notification service which only show notification while the app is running on background. For driver apps, the driver should allow to choose route during the route activation process instead of entering the route name manually. This can be useful to prevent any typo error by the driver. Driver should allow to choose the route base on certain area. Last but not least, the snap-to-the-route API should be implemented to driver apps to increase the accuracy of GPS location. In addition, further consideration and enhancement is still needed to improve the whole system in term of efficiency, security, scalability and cost. The system is needed to optimise so that the battery does not drain off due to excessive request of location service.

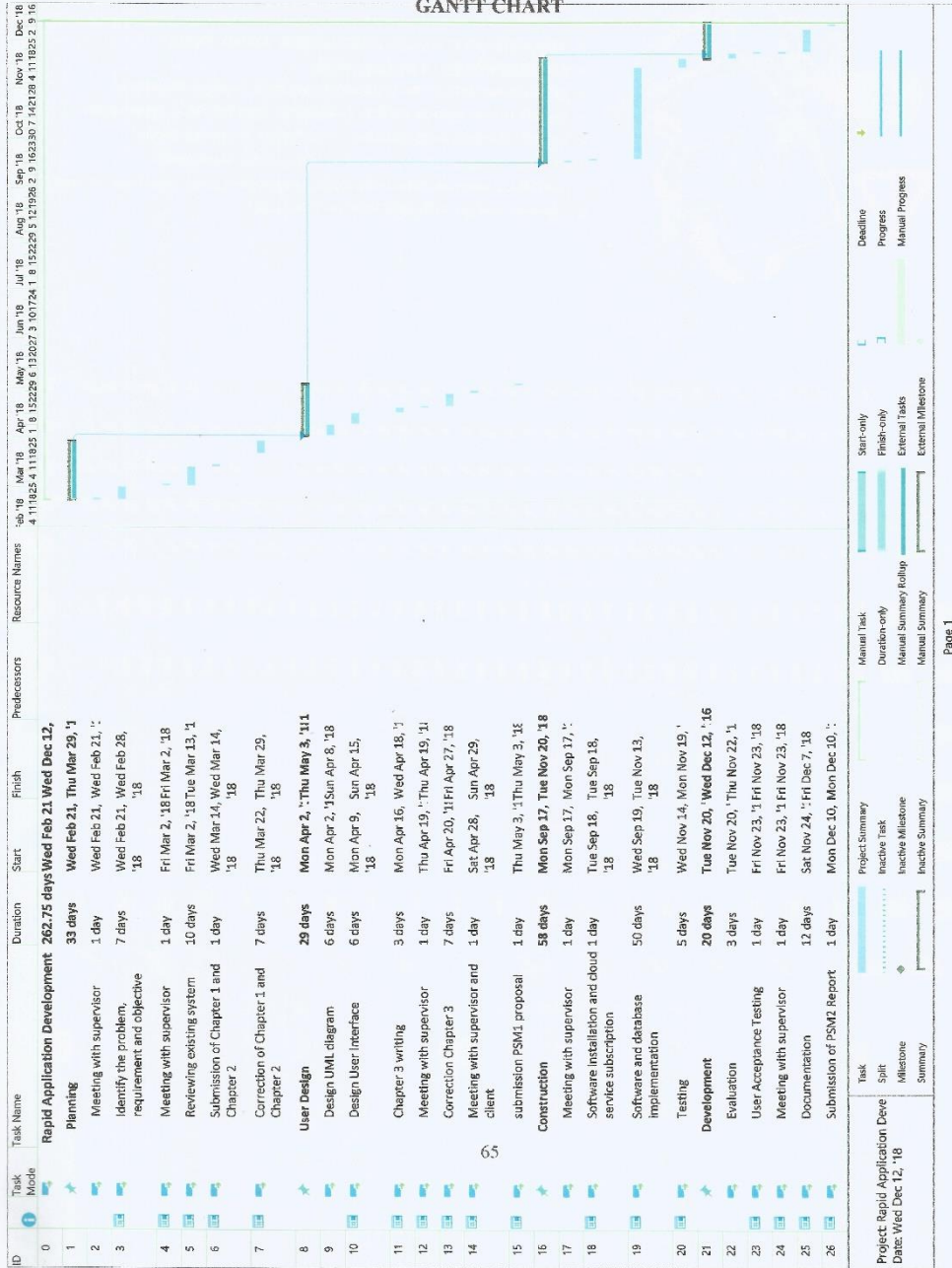


## REFERENCES

- (27 March, 2008). Retrieved from Centers for Medicare and Medicaid Services:  
<https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
- (2016). Retrieved from rapidvaluesolutions: <https://www.rapidvaluesolutions.com/wp-content/uploads/2016/10/Mobile-Usage-Statistics-and-Trends-2016.pdf>
- AA, T. (12 November, 2017). *10 reasons why Android is still better than iOS*. Retrieved from Android Authority: <https://www.androidauthority.com/10-reasons-why-android-is-still-better-than-ios-145370/>
- Faccioni, A. (November, 2016). Retrieved from GPSports Systems:  
[http://www.gpsports.com/gpsports\\_website/articles/GPS%20-%20What%20is%20it.pdf](http://www.gpsports.com/gpsports_website/articles/GPS%20-%20What%20is%20it.pdf)
- Idesis, S. (16 August, 2017). *What Is Rapid Application Development?* Retrieved from outsystems: <https://www.outsystems.com/blog/rapid-application-development.html>
- (2016). *INTERNET USERS SURVEY 2016: STATISTICAL BRIEF NUMBER TWENTY*. Cyberjaya, Selangor, Malaysia: Malaysian Communications and Multimedia Commission.
- Jisha R.C, Aiswarya Jyothindranath, & Sajitha Kumary L. (2017). IoT BASED SCHOOL BUS TRACKING. *Advances in Computing, Communications and Informatics (ICACCI)*, 509-514.
- Kamarudin Ambak, & Rozmi Ismail. (2012). Passengers Preference and Satisfaction of Public. *Australian Journal of Basic and Applied Sciences* , 410-416.
- Kendrick, J. (28 June, 2011). *Smartphone for GPS Navigation is better than a dedicated device*. Retrieved from ZDNet: <https://www.zdnet.com/article/smartphone-for-gps-navigation-is-better-than-a-dedicated-device/>
- Lam, H. (21 January, 2015). *Why, when, and how to use the Google Map API*. Retrieved from Medium: <https://medium.com/@helenflam/why-when-and-how-to-use-the-google-map-api-f5dfa35986dc>

- LavishT. (5 May, 2017). *What is Xamarin? How does it help in cross-platform mobile app development?* Retrieved from The Windows Club:  
<http://www.thewindowsclub.com/what-is-xamarin-and-cross-platform-mobile-development>
- M. Napiah, I. Kamaruddin, & Suwardo. (2011). Punctuality index and expected average waiting. *WIT Transactions on The Built Environment, Vol 116*, 215-226.
- M.T.Kamisan. (2017). UiTM Campus Bus Tracking System Using. *IEEE 15th Student Conference on Research and Development (SCORED)* , 137-141.
- Nilah Ravi Nair, & P. Sudheesh, . (2016). 2-D Airborne Vehicle Tracking Using Kalman Filter . *016 International Conference on Circuit, Power and Computing Technologies [ICCPCT]*.
- Press Room. (17 April, 2014). Retrieved from Nielsen: [www.nielsen.com/my/en/press-room/2014/rising-middle-class-will-drive-global-automotive-demand.html](http://www.nielsen.com/my/en/press-room/2014/rising-middle-class-will-drive-global-automotive-demand.html)
- Prusty, N. (8 March, 2015). *Overview of Intel XDK*. Retrieved from CodeProject:  
<https://www.codeproject.com/Tips/883716/Overview-of-Intel-XDK>
- Quinn, S. (n.d.). *Open Web Mapping*. Retrieved from Penn State university e-education:  
<https://www.e-education.psu.edu/geog585/node/763>
- Richter, F. (8 August, 2013). *Google Maps is the Most-Used Smartphone App in the World* . Retrieved from statista: <https://www.statista.com/chart/1345/top-10-smartphone-apps-in-q2-2013/>
- Shroff, R. (4 November, 2014). *5 reasons why Xamarin is the king of cross platform development*. Retrieved from Softweb Solution:  
<https://www.softwebsolutions.com/resources/5-reasons-why-xamarin-is-cross-platform-development-king.html>
- Tucker, J. (13 November, 2015). *onic Framework: Features and Benefits in Mobile App Development*. Retrieved from BDaily News:  
<https://bdaily.co.uk/articles/2015/11/23/ionic-framework-features-and-benefits-in-mobile-app-development>
- Vermaat, M. E. (2014). *Discovering Computers: Technology in a World is Computers. Mobile Devices and Internet*. Boston, USA: Cengage Learning.

## APPENDIX A GANTT CHART



**APPENDIX B**  
**USER ACCEPTANCE TEST**

**VERSION**  
**1**

**CLOUD BASED PUBLIC BUS LOCATION TRACKING AND NOTIFICATION  
APPLICATION FOR MOBILE DEVICES**

---

**FACULTY OF COMPUTER SYSTEM & SOFTWARE ENGINEERING**

**User Acceptance Test**  
**(UAT)**

## 1.0 Testing Report

The form below shows the User Acceptance Test (UAT) of the project: "Cloud Based Public Bus Tracking and Notification Application for Mobile Devices". Two android application and 1 web-based application will be tested by respective user to ensure that the system achieved the objective, functionality and usability. Driver Apps will be tested by the faculty driver, Passenger Apps will be tested by the student and the BusSeeker Management System is aimed to test by the Department of Property Management and Development (JPPH) of Universiti Malaysia Pahang (UMP).

### 1.1 BusSeeker Driver Apps Testing

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Driver login with account which is currently active	Tested with driver ID which already activated certain route.	Login failed. Prompted: "The ID is currently busy"	As expected	Pass	
Driver login with incorrect password	ID: RK1000 Password: 123456	Login failed. Prompted: "Incorrect ID or password"	As expected	Pass	
Driver login with account which does not exist	ID: RK100 Password: 123456	Login failed. Prompted: "Driver identity not found"	As expected	Pass	
Driver login with correct ID and password	ID: RK1000 Password: kuantandriver	Login successfully Proceed to route activation interface	As expected	Pass	
Driver activate route with non-existing route	Route Number: 123456 Bus Plate Number: CDC2556	Route activation failed. Prompted: "Route not exists, please check again"	As expected	Pass	
Driver activate route with non-existing bus plate	Route Number: T459 Bus Plate Number: 123456	Route activation failed. Prompted: "The bus plate is not registered to the company"	As expected	Pass	

Driver activate route with both non-existing route and bus plate	Route Number: 123456 Bus Plate Number: 123456	Route activation failed.  Prompted: "Route and bus is not registered to the company"	As expected	Pass	
Driver activate route with active bus plate	Route Number: T459 Bus Plate Number: VEG3665 (active bus plate number)	Route activation failed.  Prompted: "Bus plate is currently used by others"	As expected	Pass	
Driver activate route with correct route number and bus plate number	Route Number: T459 Bus Plate Number: CDC2556	Route activated successfully, proceed to tracking interface.	As expected	Pass	
Getting coordinate of current bus location and display on Google Map	Obtain location coordinate from device's GPS service	Obtain accurate location coordinate from device	As expected	Pass	
View the estimated arrival time to next bus stop	Obtain estimated arrival time through Google API	The time displayed should be as accurate as what suggested by Google Map and approximately same as the vehicle travelling time during on-the-road testing.	As expected	Pass	
View passenger waiting status	Notify driver by using passenger apps	Driver should notice that passenger waiting status changed to "Yes"	As expected	Pass	
Driver end route	Click on the end route button	Delete active route in database and back to login page	As expected	Pass	

## 1.2 BusSeeker Passenger Apps Testing

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
-------	-----------	-----------------	---------------	-------------	---------

Getting the coordinate of user location and display on Google Map	Obtain location coordinate from device's GPS service	Obtain accurate location coordinate from device	As Expected	Pass	
Select nearby bus stop	Getting nearby bus stop information through Google API and display the relevant one by compare with database.	User able to choose the boarding bus stop and application will mark for the selected bus stop on Google Map	As Expected	Pass	
Select desire route	Using route information from database	User able to choose the desire route and application will draw the route on Google Map	As Expected	Pass	
Bus tracking	User click on the "Search Bus" button	User will be able to see the location of next bus, estimated arrival time and the bus plate number	As Expected	Pass	
Notify Driver	User click on the "Alert Driver" button	Changing of waiting status on driver apps.	As Expected	Pass	
Notification	After user click on the "Alert Driver" button	A notification will be shown to user 3 minutes before the bus reach	As Expected	Pass	
User submit empty feedback	Empty field	Submit not successful  Prompted: "Please complete the details"	As Expected	Pass	
User submit filled feedback	Filled data field	Submit successfully  Prompted: "Thanks for your feedback"	As Expected	Pass	

### 1.3 BusSeeker Management System

Event	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
-------	-----------	-----------------	---------------	-------------	---------

Login with invalid company ID	Company ID: id123 Password: 123	Login failed Prompted: "Wrong username or password"	As expected	Pass	
Login with correct Company ID wrong password	Company ID: myrapid Password: 123	Login failed Prompted: "Wrong username or password"	As Expected	Pass	
Login with empty field	Empty field	Could not proceed to login	Expected	Pass	
Login with correct company ID and password	Company ID: myrapid Password: admin123	Login successfully	Expected	Pass	
Sign up with empty field	Empty field	Could not proceed to sign up	Expected	Pass	
Sign up with invalid email address	Email: sss@	Could not proceed to sign up	Expected	Pass	
Sign up with unmatched password	Password: 123 Confirm password: 55555	Could not proceed to sign up	Expected	Pass	
Sign up with correctly filled data field	Sign up form is fully filled with valid data	Signup successfully	Expected	Pass	
Register new bus	Empty field	Could not proceed to register bus	Expected	Pass	
	Bus Plate No: CCC5059 Description: Volvo 7900, 2016	Bus registered Prompted: "Bus added"	Expected	Pass	
Delete bus (click on delete button)	Click "OK" when prompted: "Are you sure to delete this bus?"	Bus deleted	Expected	Pass	
	Click "Cancel" when prompted: "Are you sure to delete this bus?"	Bus not delete	Expected	Pass	
Register new route	Register route with empty data field	Could not proceed with register route	Expected	Pass	

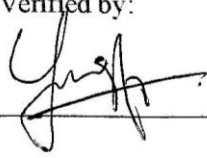



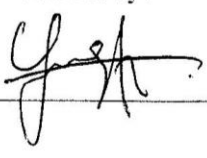
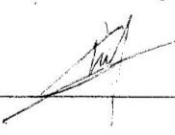
		Prompted: "Please complete the station detail!"			
	Add station with empty previous station	Could not proceed with register route  Prompted: "Please complete the station detail!"	Expected	Pass	
	Click on the map	Station latitude, longitude and station name will be automatically filled in with the bus stop nearest to the clicked location	Expected	Pass	
	Register route with complete data field	Route registered  Prompted: "Route registered successfully"	Expected	Pass	
Route details	Click on the details of each route	Route information will be shown	Expected	Pass	
Delete route (click on delete button)	Click "OK" when prompted: "Are you sure to delete this route?"	Route deleted	Expected	Pass	
	Click "Cancel" when prompted: "Are you sure to delete this route?"	Route not delete	Expected	Pass	
Register driver	Empty field	Could not proceed with registration  Prompted: "Please complete the driver detail!"	Expected	Pass	
	Unmatched password  Password: 122  Confirmed password: 1222	Could not proceed with registration  Prompted: "Password is not matching!"	Expected	Pass	

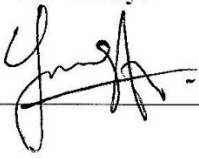
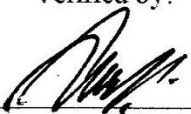
	Registered driver id	Register failed Prompted: "Driver already exists!"			
	Completely filled data field with matching password and unregistered driver ID	Driver registered	Expected	Pass	
Delete route (click on delete button)	Click "OK" when prompted: "Are you sure to remove this driver?"	Driver removed from the system	Expected	Pass	
	Click "Cancel" when prompted: "Are you sure to remove this driver?"	Driver not removed from the system	Expected	Pass	
Update driver details	Empty field	Could not proceed with update Prompted: "Please complete the driver detail!"	Expected	Pass	
	Unmatched password New Password: 122 Confirmed password: 1222	Could not proceed with update Prompted: "Confirm password is not matching!"	Expected	Pass	
	Wrong old password	Update failed Prompted: "Wrong Password!"	Expected	Pass	
	Completely filled information with correct old password with matching new password and confirm password	Update successfully Prompted: "Driver details updated."	Expected	Pass	

Delete feedback (click on delete button)	Click "OK" when prompted: "Are you sure to remove this user response?"	Feedback deleted	Expected	Pass	
	Click "Cancel" when prompted: "Are you sure to remove this user response?"	Feedback not delete	Expected	Pass	
Update company details	Empty field	Could not proceed with update  Prompted: "Please complete the detail!"	Expected	Pass	
	Unmatched password  New Password: 1010  Confirmed password: 1000	Could not proceed with update  Prompted: "Confirm password is not matching!"	Expected	Pass	
	Wrong old password	Update failed  Prompted: "Wrong Password!"	Expected	Pass	
	Completely filled information with correct old password with matching new password and confirm password	Update successfully  Prompted: "Updated"	Expected	Pass	
Log out	Click on the log out button	Log out from the management system	Expected	Pass	

## 2.0 System Testing Approval

	Name	Date
Verified by:  <hr/> Developer	CHEE XIAN YANG	12/12/18
Verified by:  <hr/> Driver	MAHMUD BIN ABDUL SAMAD General Office Assistant Faculty of Computer Systems & Software Engineering Universiti Malaysia Pahang Lebuhraya Tun Razak, 26300 Gambang, Kuantan, Pahang Tel: 09-549 1146 Fax: 09-549 2144	12/12/18

	Name	Date
Verified by:  <hr/> Developer	CHEE XIAN YANG	12/12/18
Verified by:  <hr/> Passenger	CHENG WEI QUAN	12/12/2018

	Name	Date
<p>Verified by:</p>  <hr/> <p>Developer</p>	CHEE XIAN YANG	12/12/18
<p>Verified by:</p>  <hr/> <p>Bus Company</p>	<p>MOHD AZRUL NAIM BIN AZME  PEMBANTU TADBIR  JABATAN PEMBANGUNAN &amp; PENGURUSAN HARTA  UNIVERSITI MALAYSIA PAHANG  LEBUHRAYA TUN RAZAK, GAMBANG  26300 KUANTAN  PAHANG DARUL MAKMUR  TEL: 09-549 2500 FAKS: 09-549 2499</p>	12/12/18

## **APPENDIX C**

### **USER MANUAL**

#### **1.0 GENERAL INFORMATION**

##### **1.1 System Overview**

“Cloud Based Public Bus Location Tracking and Notification Application for Mobile Devices” is a system developed by integrate with Google Map service to provide the location as well as estimated arrival time of public buses to user in real time. The system consists of two Android based mobile application which is the driver apps and passenger apps for location tracking purpose and one web-based management system for bus company to manage their route information. The system needs database server to functioning well therefore, internet connection is a must for the system.

##### **1.2 Organization of the Manual**

- Chapter 1: General information that provide an overview of “Cloud Based Public Bus Location Tracking and Notification System for Mobile Devices”.
- Chapter 2: System summary that provide the system configuration details, user access levels and contingencies and alternate mode of operation.
- Chapter 3: Get started the enclose with guideline on how the use the passenger apps, driver apps and the management system

## **2.0 SYSTEM SUMMARY**

### **2.1 System Configuration**

The BusSeeker (Driver Apps) and BusSeeker (Passenger Apps) is compatible to run on all the Android phone with minimum Android version 5.0 (Lollipop) and above. Internet connection and location service is a must for the system so that location of the driver location can be obtained and provided to user via the database server. For the web-based management system, internet connection is also needed for bus company admin to perform a series of management action towards the route, bus and driver.

### **2.2 User Access Level**

User of “Cloud Based Public Bus Location Tracking and Notification Application for Mobile Devices” is targeted to have at least education level of secondary school with basic map reading skill.

### **2.3 Contingencies and Alternate Mode of Operation**

- In case of no internet access, driver apps will not be able to login to the application and provide any location update to the server, passenger will not be able to retrieve the bus stop, route and bus information. For the management system, bus company admin will not be able to access the system and perform and kind of management function.
- In case of no location access, driver apps will not be able to retrieve the bus location, passenger apps will not be able to start by retrieving user’s location and suggest the surrounding bus stop.

### 3.0 GET STARTED

#### 3.1 BusSeeker (Driver Apps)

##### 3.1.1 Driver Login

After installation of BusSeeker (Driver Apps) to the Android device, the apps would be ready to use by the driver without any configuration. The login page will be prompted out for driver to login to the system. Driver required to provide driver ID and password and tap on the “Login” button to login to the system. Figure 3.1 shown the login interface of BusSeeker (Driver Apps).



Figure 3.1 Driver apps login interface

##### 3.1.2 Activate Route

After successfully login, driver is required to activate the route before proceeding to real time location update for tracking purpose. To activate a route, driver must enter the route number and the bus plate number which the driver currently driving. Press “Activate” button to activate the route.



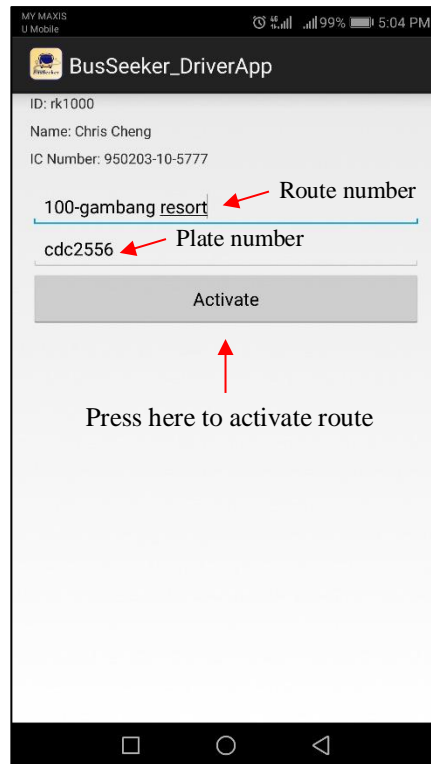


Figure 3.2 Route activation interface

### 3.1.3 Location Tracking

After driver successfully activated the route, the tracking interface will be shown to the driver. Driver are advised to start the application at the bus stop so that the application can start to initiate the tracking process. Passenger waiting status will become “YES” when there is passenger waiting at next bus stop. Driver can press on the “End Route” button to stop the tracking process. In figure 3.3, picture A show the situation when driver activated the route far away from a bus stop. Therefore, application cannot initiate the location tracking process. If driver moved to the bus stop or activated a route on the bus stop, tracking process will automatically be initiated as shown in picture B.

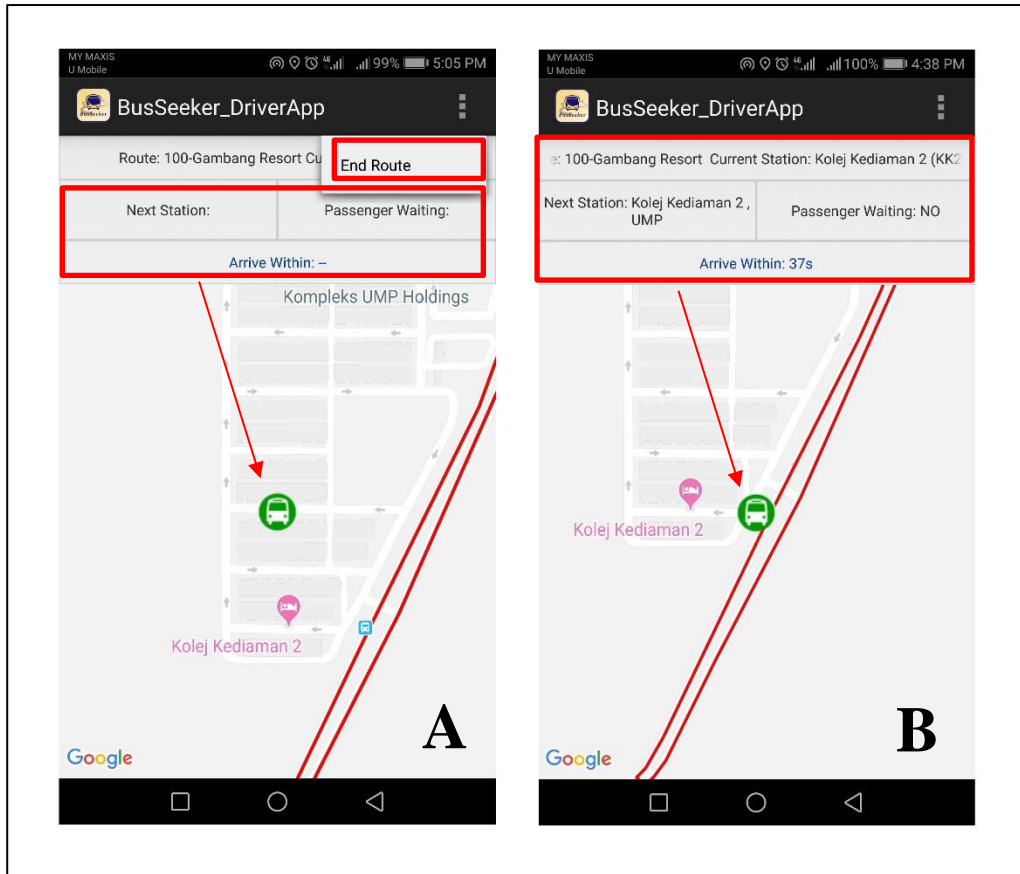


Figure 3.3 Location tracking interface

## 3.2 BusSeeker (Passenger Apps)

### 3.2.1 Bus Stop and Route Selection

After installation of BusSeeker (Passenger Apps) to the Android device, the apps would be ready to use by the passenger without any further configuration. User will be able to see his or her location on the map. Before getting the location of any buses, user is required to select the boarding bus stop by press on the first selection bar. Nearby bus stop name will be suggested, and the location of bus stop will be marked on the map based on user's selection. The next step is to choose the desired route so that system can search for the next bus coming to the bus stop. The route will be drawn on the map too based on user's selection of route. Previous tracking process will be stopped whenever user reselect the bus stop and route. Figure 3.4 shown the interface on how user going to select the bus stop and route.

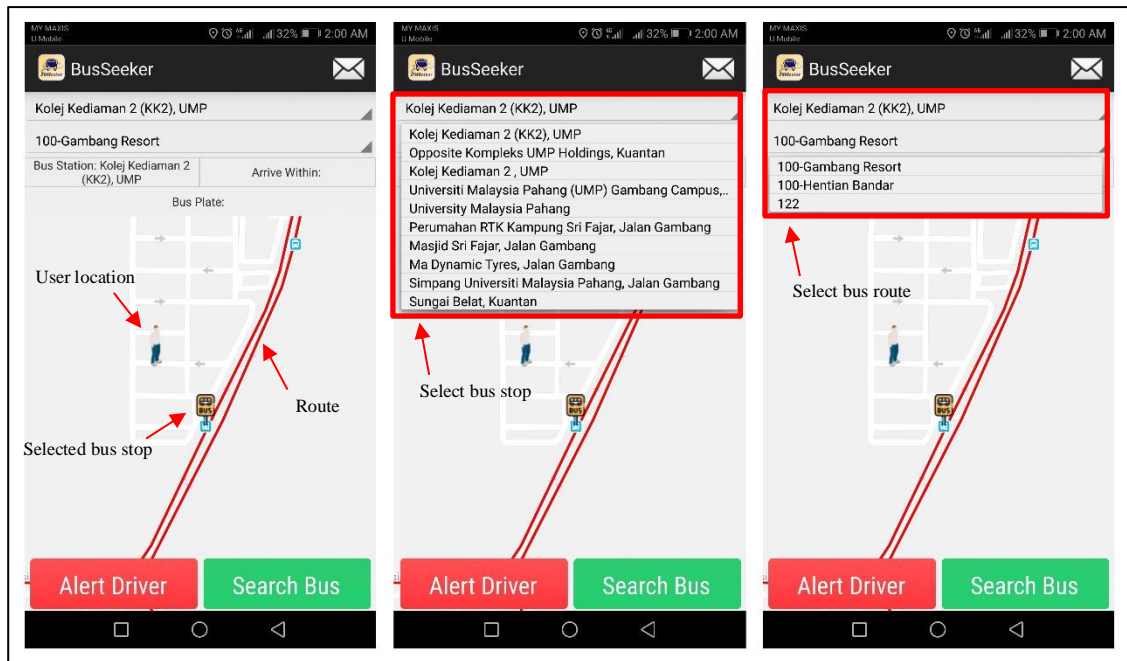


Figure 3.4 Bus stop and route selection.

### 3.2.2 Search Bus

After selecting the desired bus stop and route, user can start to search for bus by press on the “Search Bus” button. System will suggest and display the location of next bus pass by the bus stop to the user. User can obtain the estimated arrival time of bus as well as the bus plate number. If user decided to get on the bus, user can alert the driver by pressing on the “Alert Driver” button. Figure 3.5 shown the situation when there is no bus pass by the station and the situation when bus is available. After user alert the driver, a notification will be displayed to user before the bus reach the bus stop. In figure 3.6, a notification is prompted to the user telling that “Your bus for route: 100-Gambang Resort arrive soon, bus plate number: CDC2556”

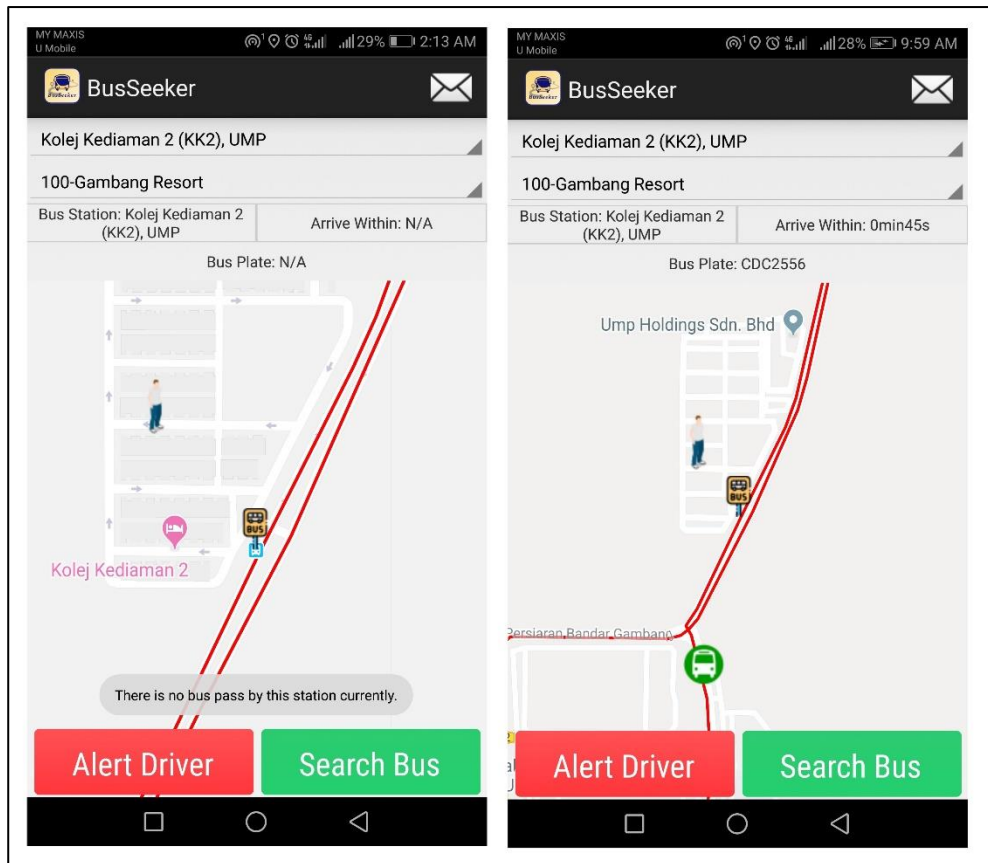


Figure 3.5 UI for bus tracking.

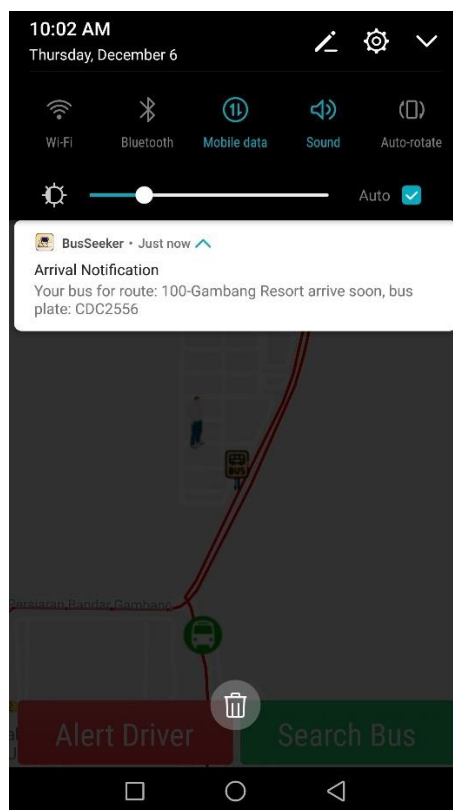


Figure 3.6 Notification

### 3.2.3 Provide Feedback

User can give responses to the bus company by press on the “mail” icon on top right of the user interface. User can fill in all the details with message and press “Submit” button to send the message to bus company admin. User need to choose which company to give response to.

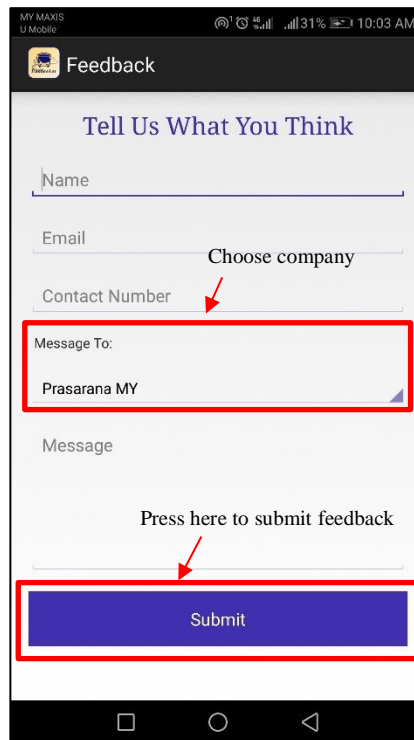
The image shows a mobile application interface for providing feedback. At the top, the status bar shows 'MY MAXIS U Mobile' and the time '10:03 AM'. Below the status bar is a header with a logo and the word 'Feedback'. The main content area is titled 'Tell Us What You Think' and contains several input fields: 'Name', 'Email', 'Contact Number', and a dropdown menu labeled 'Choose company' which currently displays 'Prasarana MY'. Below these fields is a large text area for the message, with the text 'Message' and 'Press here to submit feedback' above it. At the bottom of the form is a prominent blue button labeled 'Submit'. Red annotations are used to highlight specific elements: a red box around the 'Choose company' dropdown, a red arrow pointing to it, another red box around the 'Submit' button, and a red arrow pointing to the text 'Press here to submit feedback'.

Figure 3.7 Feedback form

## 3.3 BusSeeker Management System

### 3.3.1 Management System Login Page

Bus company admin is needed to provide the login ID and password to login to the system before performing any management system. Click on the “Login” button to login to the system. Figure 3.8 shown the login interface of BusSeeker Management System. To sign up to the system, click on the sign-up button to proceed to show the sign up form.

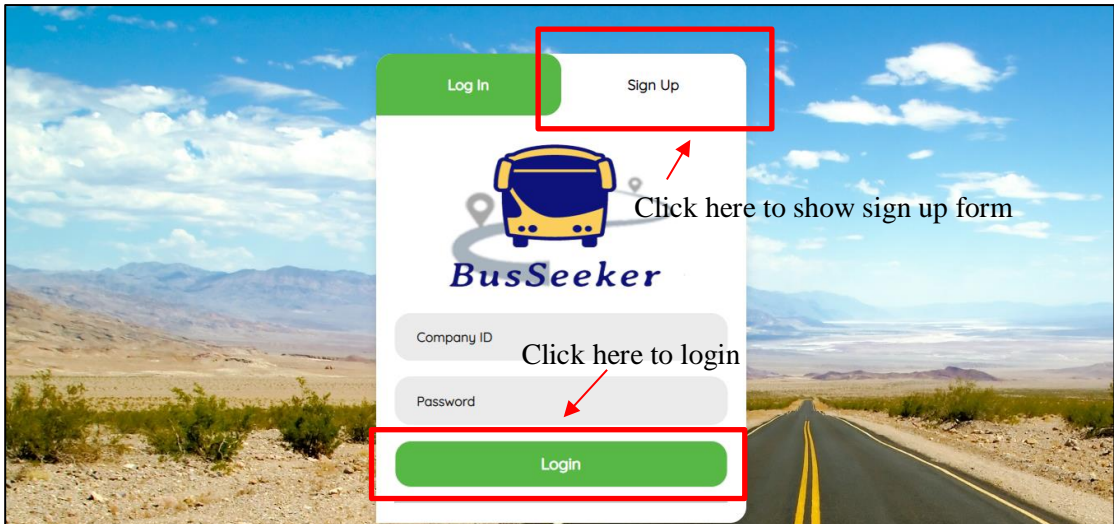


Figure 3.8 BusSeeker Management System login interface.

### 3.3.2 Management System Sign Up Page

Bus company admin need to fill in company details, company ID as the login ID as well as password to sign up to the system. Click on the “Create Account” button as the final step to sign up to the system. Figure 3.9 shown the sign-up form to be filled in by the bus company admin.

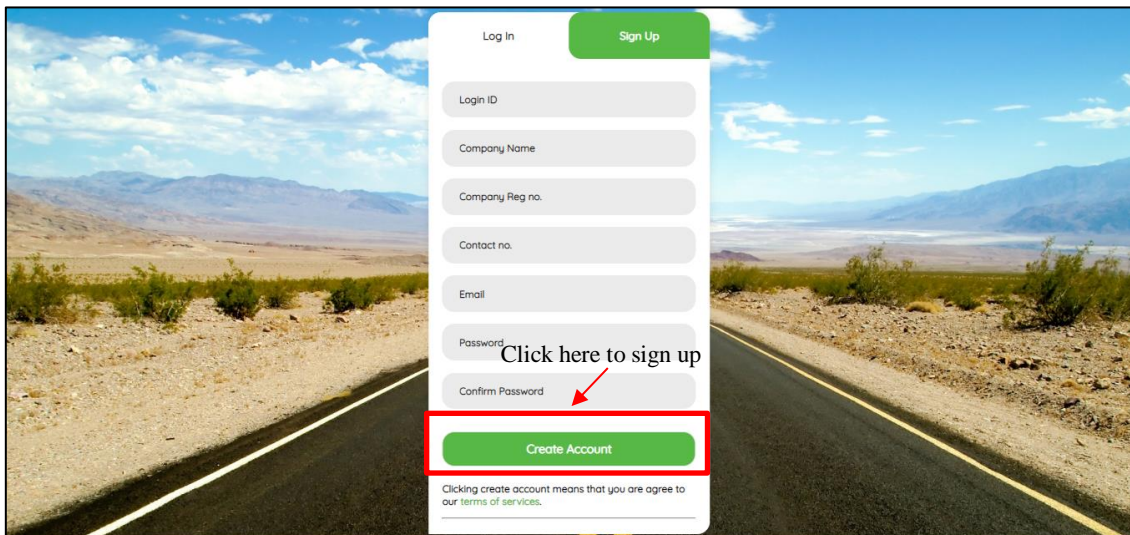


Figure 3.9 BusSeeker Management System sign up form

### 3.3.3 Manage Bus

Bus company admin will need to fill in the bus plate number, description and click on the “Add” button to register the bus to the system. Search any bus information by enter the bus plate number of description keyword in the searching bar. Click on the “Delete” button to delete the bus from the system. Figure 3.10 shown the bus management page for the bus company admin.

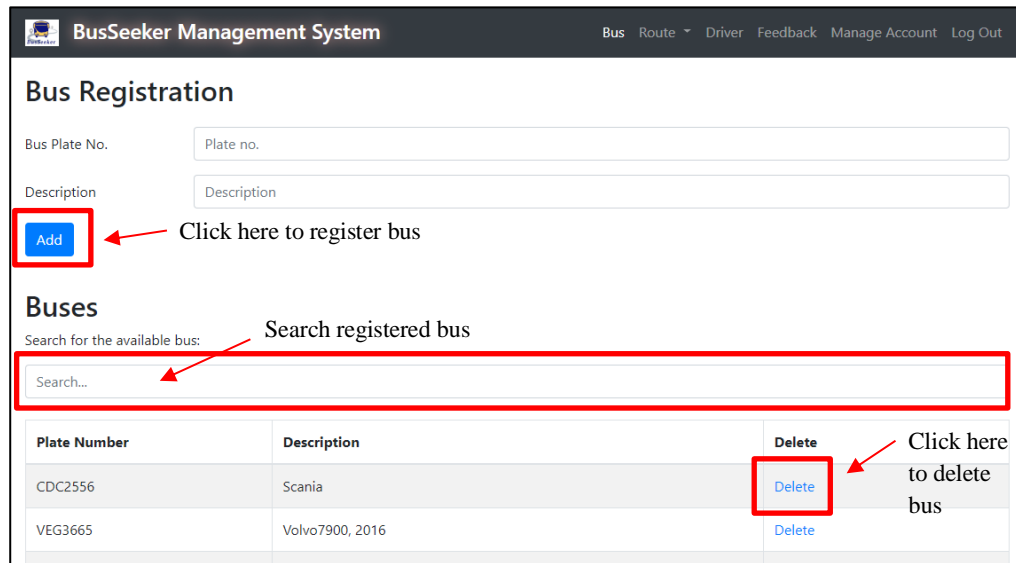


Figure 3.10 Bus management page

### 3.3.4 Manage Route

To add new route to the system, bus company need to provide route information as shown in figure 3.11. System will automatically complete the bus stop information including the latitude, longitude and bus stop name whenever user had clicked on the bus stop on the map provided. Bus company admin is also advised to click on the road area beside the bus stop so that the route can be formed more precise in the BusSeeker Android application. Click on the “Add Station” button to add the next bus stop. Repeat the steps until finish registered all the bus stop to the system in ascending order. Click on the “Remove” button to remove the unwanted bus stop. Select the “One Way” checkbox if the route is not a loop. Click on the “Register Route” button to finish the route registration process.

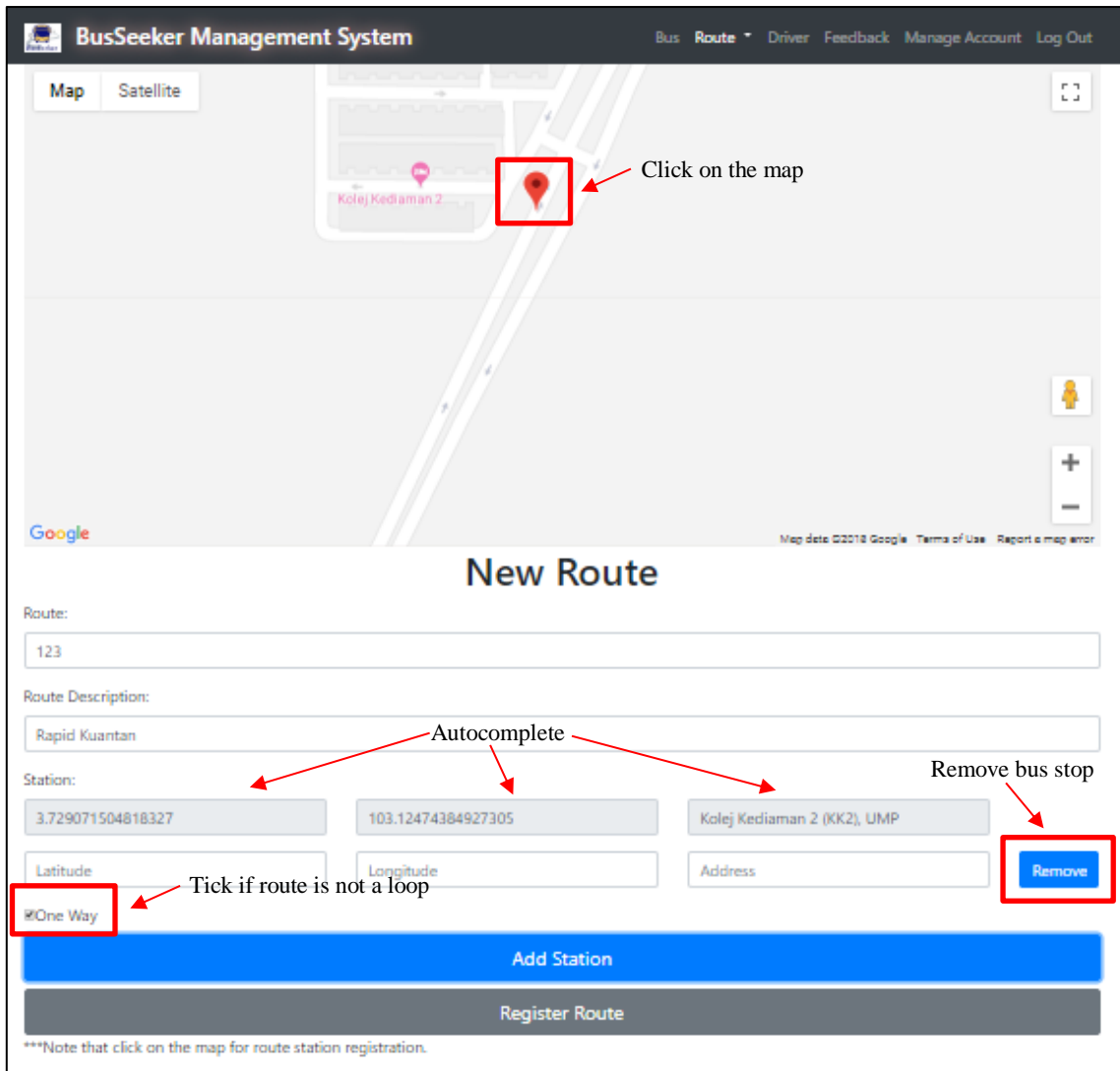


Figure 3.11 Route registration page

Figure 3.12 shown the route management page where bus company admin can see the bus stop details of each registered route by clicking on the “Details” button. Search function is also provided to ease of route filtering. Bus company admin can also click on the “Delete” button to delete the old or outdated route. Click on the “OK” button to continue with route delete process.



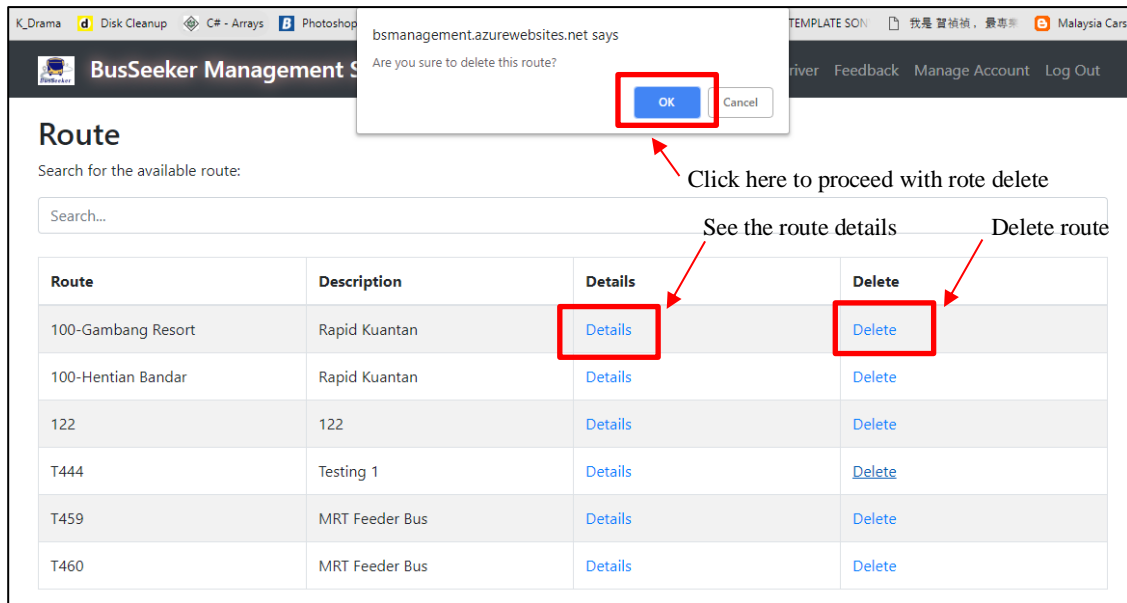


Figure 3.12 Route management page

### 3.3.5 Manage Driver

Bus company admin can add their driver to the system by providing the driver's personal information such as name, IC number, contact number, email address, login ID and password to the system. Normally, the driver has to report duty to the company office and fill up the password field themselves. Bus company admin can search through the driver list with any keywords. To remove the bus driver from the company, admin just have to click on the "Delete" button and click "OK" while confirmation dialog prompted out. Click on the "Update" button to update the driver personal information.

**BusSeeker Management System** Bus Route **Driver** Feedback Manage Account Log Out

### Driver Registration

Driver Name:  Login ID:

IC:  Password:

Contact no.:  Confirm Password:

Email:

**Register** ← Click here to register driver

### Driver

Search for the existing driver:  [Delete driver](#)

ID	Name	IC	Contact no.	Email	Delete/Update
arccus95	LIM GUO HONG	951104-04-5817	011-6868158	arccuslim@hotmail.com	<b>Delete</b> <b>Update</b>
M1000	CHEE XIAN YANG	950828-01-5413	017-6388158	xian-yang@hotmail.com	Delete / Update
M1001	NORMAN YEW	690101-01-1119	016-1314520	norman69@yahoo.com	Delete / Update

Update driver details

Figure 3.13 Driver management page

Figure 3.14 shown the page for bus company admin to update the driver personal information. Only contact number, email address and password are changeable. Click on the “Update” button to update the driver details.

**BusSeeker Management System** Bus Route **Driver** Feedback Manage Account Log Out

### Driver Personal Information Update

Driver Name: LIM GUO HONG Login ID: arccus95

IC: 951104-04-5817 Old Password:

Contact no.: 011-6868158 New Password:

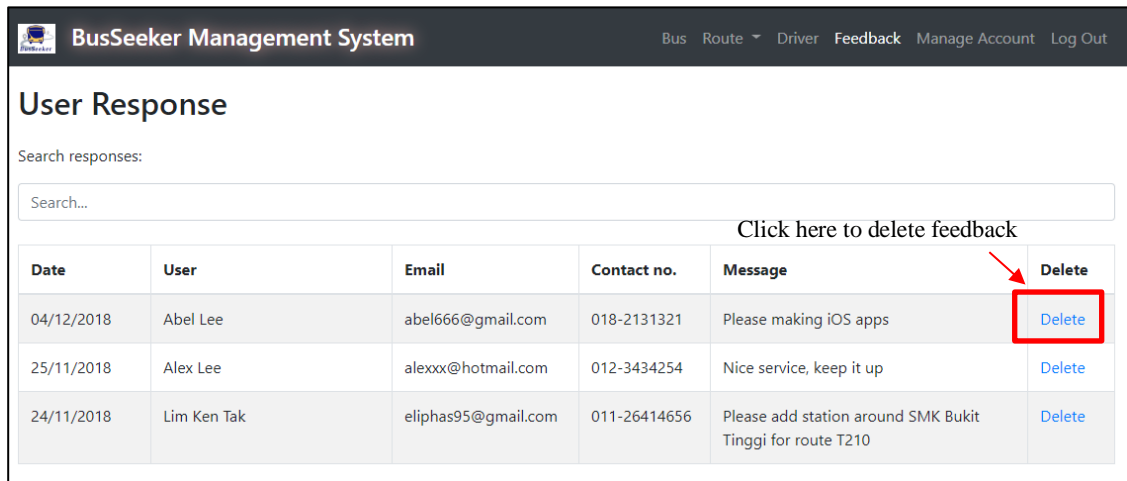
Email: arccuslim@hotmail.com Confirm Password:

**Update**  ← Click here to update

Figure 3.14 Driver information update field

### 3.3.6 Manage Feedback

Bus company admin can view the responses given by the passenger and make improvement to the service provided. Admin can also delete irrelevant feedback by clicking on the “Delete” button. Bus company admin may also search for the responses by date or any keyword in the message column. Figure 3.15 shown the feedback management page.



BusSeeker Management System

Bus Route Driver Feedback Manage Account Log Out

### User Response

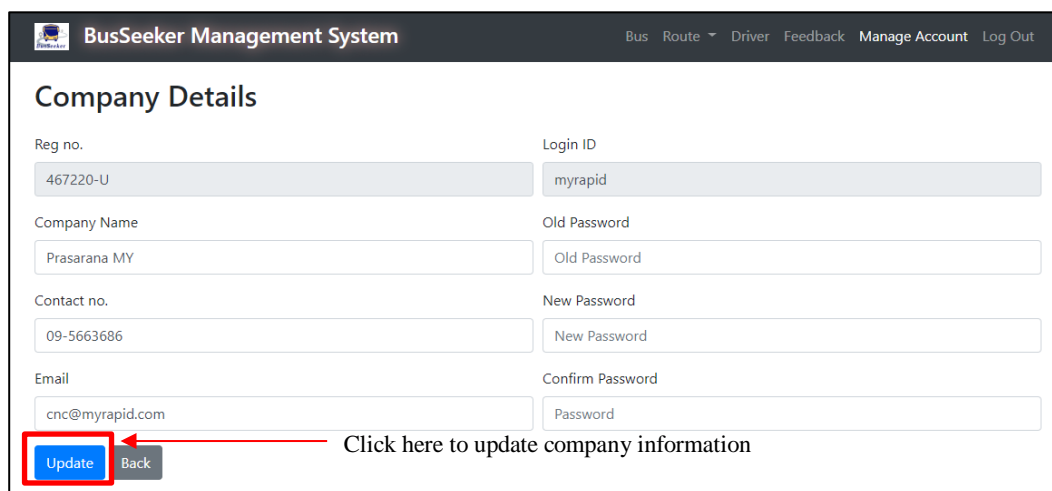
Search responses:

Click here to delete feedback

Date	User	Email	Contact no.	Message	Delete
04/12/2018	Abel Lee	abel666@gmail.com	018-2131321	Please making iOS apps	<a href="#">Delete</a>
25/11/2018	Alex Lee	alexxx@hotmail.com	012-3434254	Nice service, keep it up	<a href="#">Delete</a>
24/11/2018	Lim Ken Tak	eliphas95@gmail.com	011-26414656	Please add station around SMK Bukit Tinggi for route T210	<a href="#">Delete</a>

Figure 3.15 Feedback management page

The last function of BusSeeker Management System is to allow bus company admin to change their company details as well as change the password for security purpose. Only company name, contact number, email address and password are changeable. Click on the “Update” button to proceed with the information update as shown in figure 3.16. Make sure that the new password is same as the confirm password.



BusSeeker Management System

Bus Route Driver Feedback Manage Account Log Out

### Company Details

Reg no. 467220-U Login ID myrapid

Company Name Prasarana MY Old Password Old Password

Contact no. 09-5663686 New Password New Password

Email cnc@myrapid.com Confirm Password Password

[Update](#) [Back](#)

Click here to update company information

Figure 3.16 Bus company details update field