UTeM Bus Tracking using Google Map

CHEN JIAN TAT

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: UTeM Bus Tracking using Google Map

SESI PENGAJIAN: 2016/2017

Saya CHEN JIAN TAT

(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

_____	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
_____	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
/	TIDAK TERHAD	

(TANDATANGAN PENULIS)

Alamat tetap: 8-7-18, TAMAN MEHARJA, JALAN LOKE YEW, 55200, KUALA LUMPUR

Tarikh: 22/8/2017

(TANDATANGAN PENYELIA)

ERMAN HAMID

Nama Penyelia

Tarikh: 22/8/2017

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

UTeM Bus Tracking using Google Map

CHEN JIAN TAT

This report is submitted in partial fulfilment of the requirements for the Bachelor of Computer Science (Networking)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DECLARATION

I hereby declare that this project report entitled

UTeM Bus Tracking using Google Map

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : _____ Date : 22/8/2017

(CHEN JIAN TAT)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Networking) With Honours.

SUPERVISOR : _____ Date : 22/8/2017

(Mr. ERMAN HAMID)

DEDICATION

I dedicate this thesis to my family who sacrificial care for me with affections and love and make me possible to complete this project within the time limit.

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to my supervisor, Mr. ERMAN HAMID, who provided me suggestions and assistance along the project development until complete this project successfully.

I also appreciate the support given by my family and friends who always provided me with moral and emotional support when I facing problems.

ABSTRACT

UTeM Bus Tracking System using Google Map is a bus tracking system to track the buses in the Universiti Teknikal Malaysia Melaka (UTeM). This is due to the student at UTeM faces some problem with the bus shuttle service. The problem such as the student does not know the bus location, bus arrival time and no application that can alert the student when the bus is almost reached. Therefore, the purpose of this project is to develop mobile application to allow the student to track, know the bus arrival time and alarm that can alert the student when the bus is almost reached. The local and foreign university bus tracking system is analyzed to find the proposed solution. After the application is developed, testing was conducted with 10 users. The result from the testing is most of the users are satisfy with the system.

ABSTRAK

UTeM Bus Tracking System using Google Map merupakan sistem penjejakan untuk menjejak bus di Universiti Teknikal Malaysia Melaka (UTeM). Hal ini sedemikian kerana, pelajar di UTeM menhadapi beberapa masalah dengan perkhidmatan basnya. Masalahnya seperti, pelajar tidak tahu lokasi bas, masa ketibaan bas dan tiada aplikasi yang boleh menyedarkan pelajar apabila bas hamper sampai. Oleh itu, tujuan projek ini adalah membangunkan aplikasi yang membolehkan pelajar mengesan, mengetahui masa ketibaan bas, dan memberi penyedaran kepada pelajar apabila bas hamper sampai. Sistem penjejakan bas di universiti tempatan dan asing dianalisis untuk mencari penyelesaian yang sesuai. Selepas pembangun aplikasi, ujian telah dijalankan dengan 10 pengguna. Hasilnya, kebanyakan pengguna rasa puas hati dengan sistem ini.

Tables of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction

Universiti Teknikal Malaysia Melaka (UTeM) is the 14th public university in Malaysia. This university consists of three campuses which is main campus, technology campus and city campus. UTeM provides many services for students such as library, health centre, cyber hall and shuttle service. Among these services, students use the most is shuttle service. This service provides free transportation from hostels to campus. Besides, the operation hour is from 6.00 a.m. to 11.00 p.m. on weekdays with service almost every one hour. Although this service is convenient for the students, but students still not satisfy with the services. This is due to some problem such as the students miss the bus, bus driver does not arrive on time, and students do not know the bus estimate arrival time.

Therefore, the purpose of this project is to make an android application helper. This android application should be able to track the location of the bus using Google Map, estimate bus arrival using Google Maps Distance Matrix API, alert the students when the bus is almost reaches the bus stop, and view bus schedule.

1.2 Problem Statement

The main problem with the UTeM shuttle service is the student does not know the location of the bus. Besides, the student does not know the bus arrival time. This make the student always wait for the bus mindlessly. Moreover, there is no application that can alert the student when the bus is almost reached. This can cause the student miss the bus and cannot work as planned.

Table 1.1 Summary of Problem Statement

PS	Problem Statement
PS1	Student does not know the bus location
PS2	Student does not know the bus arrival time
PS3	There is no application to alert the student when the bus is almost reach

1.3 Project Research Question

Project research question is used to identify the reason for the student late to university, reason for the bus late to the bus stops, the existing solution for this problem, and global solution tackle this problem, reason to use the android application and Google Map. Table 1.2 shows the summary of the project research question.

Table 1.2 Summary of Project Research Question

PRQ	Project Research Question
PRQ1	How to get the current location?
PRQ2	How to develop mobile application?
PRQ3	What is the existing system?
PRQ4	What is the solution in the global?
PRQ5	Why use the android application?

1.4 Project Objective

The objectives for this project are shown in below:

1. To develop android application to allow the student to track the bus.
2. To estimate the bus arrival time in the application.
3. To alert the user when the bus is almost reach the bus stop.

1.5 Research Hypothesis

Based on the research, the current android bus app has insufficient features. Some of the hypothesis has been suggested to improve the current bus tracking system. The figure 1.1 shows the problem of the current android bus application and the hypothesis to make an improvement.

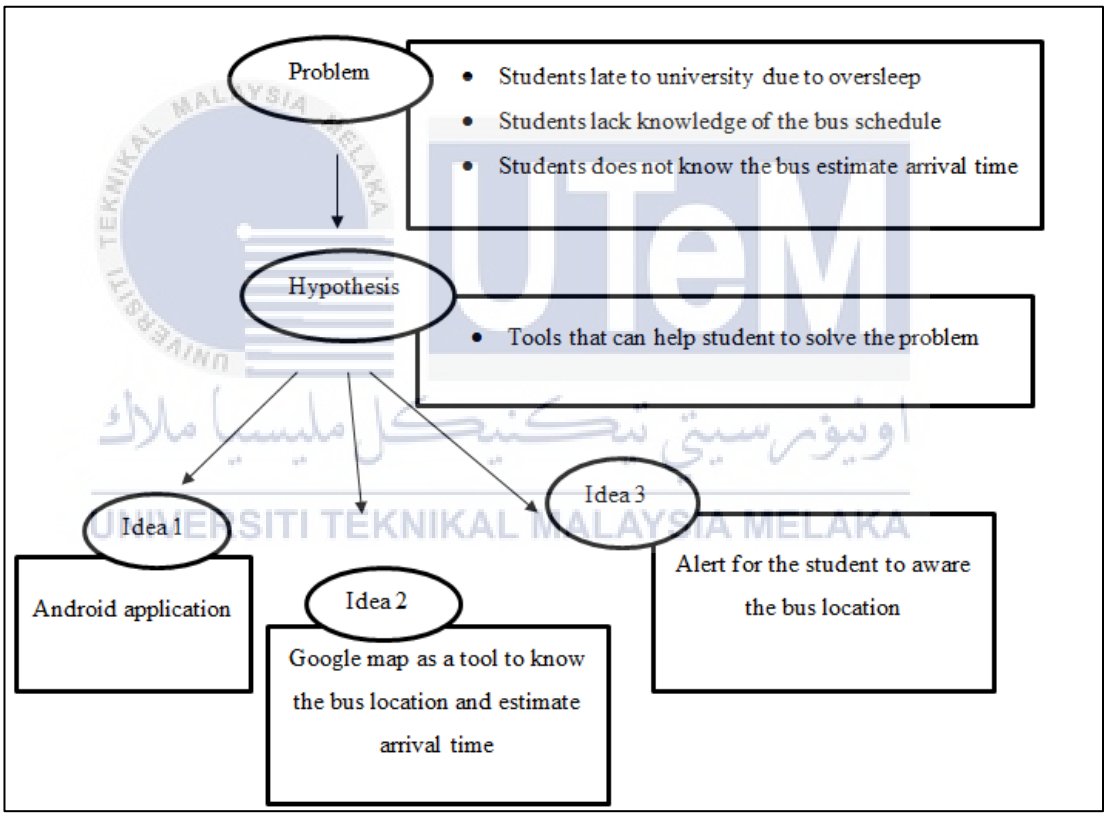


Figure 1.1 Research Hypothesis diagram

1.6 Project Scope

The project scope is focus on how to improve the UTeM shuttle service and give satisfaction to the students when using the shuttle service. The function of this system is to solve some dissatisfaction of the students with the shuttle service. The location is focus on the UTeM.

1.6.1 Modules:

a.) Admin module

Admin can perform administration and manage the shuttle information, such as bus driver information.

b.) Driver module

The bus driver can send bus current location to the database server. The user can know the bus location by retrieving the location of the bus.

c.) ETA module

The application can estimate time of arrival by using Google Maps Distance Matrix API. The application need to retrieve the bus current location and the bus stop location to estimate the arrival time.

d.)Alert module

The application can retrieve the bus estimate arrival time and use it to alert the user.

1.6.2 User

a.) UTeM students

The main user of the system is the students because the purpose of this system is to help the student to solve the shuttle service dissatisfaction.

b.) UTeM shuttle driver

The drivers are also core user because drivers update the bus location continuously and bring the students to their destination.

c.) Admin

Admin is the user that can administrate the system. The Admin can manage the shuttle service information.

1.7 Project Significance

The project can help the students or users know the bus schedule. Besides, the users can get satisfaction from the ability of the system which is Google Map and alert when the bus is almost reached the station.

1.8 Conclusion

In conclusion, this project can help the students to track the UTeM bus, know the bus estimate arrival time, getting alert when the bus is almost reach bus stop, and view the schedule.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This project is to make android application to track the UTeM bus. The study of the problem and solution for the bus tracking system is needed to improve the knowledge in this field and implement the suitable solution in this project. Therefore, literature review is needed.

This chapter discuss the problem and solution for the existing bus tracking system. Besides, this chapter give a better understanding for the technique used in the project.

The domain for this project is Global Positioning System technology (GPS). “The Global Positioning System (GPS) is a global navigation satellite system deployed by the US Department of Defense and maintained by the US Air Force. GPS is a space-based radio navigation system that provides accurate location and timing services to anyone with a GPS receiver. This service, made available to

civilians in 1996 for navigation purposes, is free of charge, can support an unlimited number of users, and functions anywhere in the world."

2.2 Research Problem

A research problem is a definite about an area of concern. This consists of the concept and theory for the area of concern.

2.2.1 Concept

The concept of this project is the driver uses the GPS of his android phone to get the current location the bus and sends to the server. The student can get the hold of the bus location by receiving it from the server. In order to know the estimated time of arrival (ETA) of the bus to bus stop a process is needed. This process is using the Google map distance matrix API. This API need the origin (bus location), and the destination (bus stop) to process it function and produce the ETA. Hence, student can know when the bus will arrive.

2.2.2 Theory

Muthumurugesan, Nalini, and Vinodini (2013) describe "GPS uses 27 satellites (24 currently working, 3 for backup) to enable a user to pin-point his or her current location. The calculation to ascertain the location is based on a arithmetic theory known as trilateration. Since the Earth is a sphere, each satellite generates a

specific part of the sphere it hovers and revolves with. An intersection of three such spheres which is closest to the GPS device's locations done and the location is thus identifiedGPS gather the requesting device's current location and provide accurate response; the GPS receiver requires two vital details, i.e. the location of at least three satellites. Many mobile phones like an android mobile have integrated GPS (Global Positioning System) tracking systems that provide independent mobile tracking through the device alleviating the need for a separate dedicatedGPS unit". This statement proofs that the android mobile can get it accurate location via GPS.

According to Google Developer, "The Google Maps Distance Matrix API is a service that provides travel distance and time for a matrix of origins and destinations." Hence, by having the origin (bus location) get the data from the driver's android mobile GPS and the destination (bus stop) which can be manually inserted, the ETA of the bus can be obtained.

2.3 Research Question

From the research problem, the research question is created.

2.3.1 Background of Android Application

According to Jianye Liu (2011), "Android is a comprehensive operating environment that based on Linux® V2.6 kernel, it is also a layered system, the architecture of Android system"

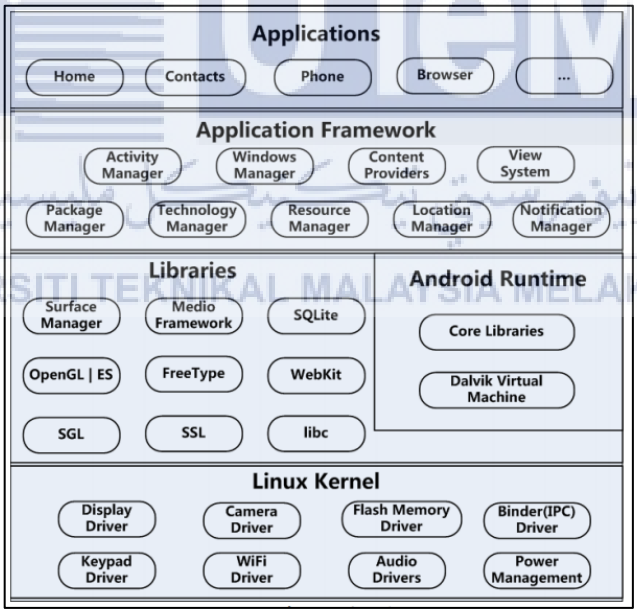


Figure 2.1 Architecture of Android system

Besides, Jianye Liu (2011) also mention there are four types of application components and each of them has their own purpose and lifecycle that define how it is started and end.

The first component is Activity, “activity represents a single screen with a user interface. The activities in an application work together to form a cohesive user experience, but each one is independent of the others. As such, a different application can start any one of these activities. An activity is implemented as a subclass of Activity. The particular form that an activity show users and the amount of activities in an application depend on how the developer design the application. In a multiple activities application, typically, one activity is specified as the "main" activity, which is presented to the user when launching the application for the first time. Each activity can then start another activity in order to perform different actions. Each time a new activity starts, the previous activity is stopped, but the system preserves the activity in a stack (the "back stack")”

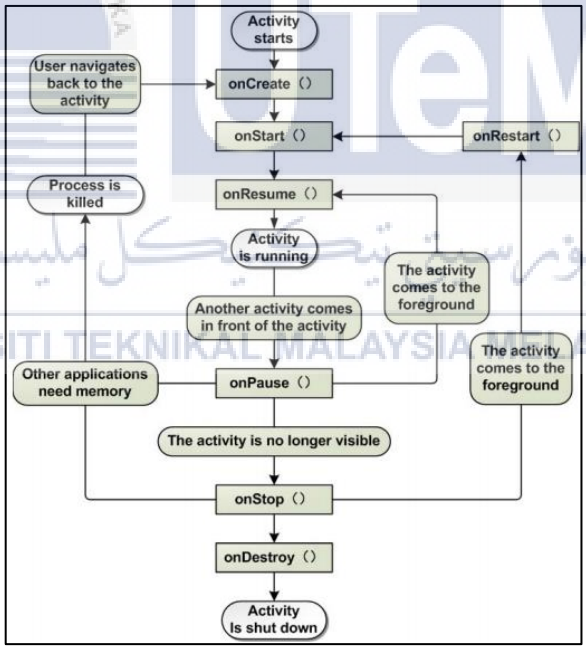


Figure 2.2Lifecycle of an activity

Second component is Service. “Service is an Android component that runs in the background to perform long-running operations or to perform work for remote processes and does not provide a user interface. An activity can connect or bind a service that is running. (if the service is not running, launch it). When connected to a

service, the activity can communicate with the service through the interface that the service exposed. Like other application components, service components always running in the main thread of an application by default. So for the intensive or blocking operating a service performs (may slow down activity performance), it is usually start a new thread inside the service."

Third component is content providers which "provide data share mechanism among applications. The data that be shared could in the file system, a SQLite database, or any other persistent storage location an application can access. A content provider is implemented as a subclass of Content provider, it defines the data format it supported and provides a set of method to enable other applications to query or modify the data. But an application does not call these methods immediately, instead, it call these methods by an object named Content Resolver." "Content Resolver can communicate with every Content Provider. Content Resolver cooperated with Content Provider to manger IPC (inter process communication) while sharing data."

The last component is "Broadcast Receivers is in charge of the reception of system wide broadcast and take response aiming at the information that a broadcast transmitted. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low. Applications can also initiate broadcasts. There could be any number of Broadcast Receivers in an application and each Broadcast Receiver implemented as a sub class of Broadcast Receiver. Although broadcast receivers don't display a user interface, they may create a status bar notification to alert the user when a broadcast event occurs. More commonly, though, a broadcast receiver is just a "gateway" to other components and is intended to do a very minimal amount of work."

2.3.2 Integrated Development Environment (IDE)

IDE is needed to develop the android application. There are two very popular IDE which are Android Studio (AS) and Eclipse. Both of them have their own strength and weakness. Table 2.1 shows the comparison between AS and Eclipse.

Table 2.1 AS vs Eclipse

Aspect	Android Studio	Eclipse
User Interface	Simple user interfaces because it is dedicated for android development.	Complex user interface because Eclipse compatible with multiple platforms and need to work on Android Development tools to create android application.
Gradle Structure	Gradle build system is more efficient and organized.	Apache Ant is robust XML based system.
Code Completion	The code completion is more intelligent by using IntelliJ platform.	Might give wrong result.
Google Cloud Platform	Build in support.	Need to add plugin to Google plugin for Eclipse to support it.
App testing and debugging	AS can set up test classes and including them in the run configuration of projects. Hence, program bugs can be detected and ironed out when it still in build stage.	Does not support.

2.4 Research Gap

Following shows research gap for this project.

2.4.1 Definition

A research gap is defined as the project or system which having some missing function or reach the ability limits as a conclusion for a question. It is the research for the missing element for the existing research literature and fills the research approach to improve the existing project. The figure 2.3 shows the idea of the Research Gap.

Figure 2.3 Idea of Research Gap

Figure 2.4 shows the Research Gap for the UTeM Bus Tracking System using Google Map. There are some extensions to focus with to make the system functioning more comprehensive.

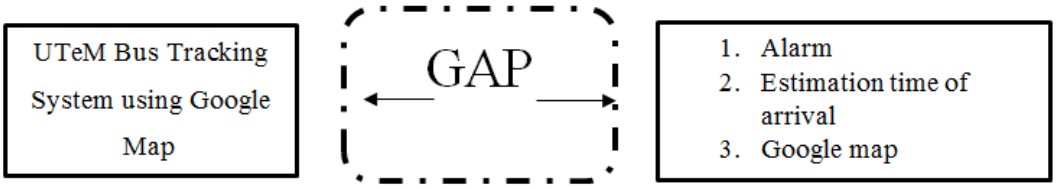


Figure 2.4 Research Gap for Intelligent Automated Gate System

2.4.2 Importance of Research Gap

The Research Gap is very important during working on project. Below shows the importance of Research Gap:

1. Improvement for the future research scope.

2. Identifying the objectives of the future research.

3. Refining the understanding about the research topic.

2.4.3 Comparison of Existing System

The study of the existing system is conducted to understand the current bus tracking system. Study is done in the existing system in Malaysia and foreign country. In Malaysia, there is some university implement bus tracking, for example UTHM public shuttle tracking by katsana and UM Bus Tracking. While foreign country university like Northern Illinois University and Rice University also has their own bus tracking system. Thus, analysis is conducted in these bus tracking systems.

2.4.3.1 UTHM public shuttle tracking

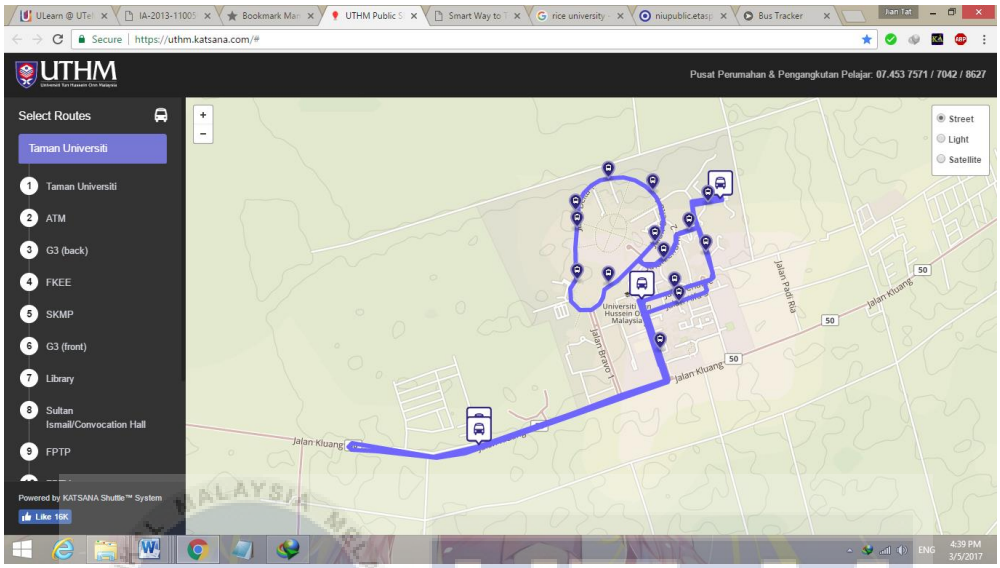


Figure 2.5 UTHM Public Shuttle tracking by katsana

Figure 2.5 show the UTHM Public Shuttle tracking system. This system is developed by KATSANA Advanced Telematics Malaysia. It is a webpage with embed Google Map. This webpage allow the UTHM students to track the shuttle by accessing this web page https://uthm.katsana.com/#

There are bus icons which indicate the location of the bus and the bus stop icon to shows the location of the bus stop. Besides, when tap on the bus stop icon, it will show the bus stop name and when move the cursor on the bus icon, it will show the bus name. On the left site of the webpage, it shows the route name and click it will create the route in the Google Map and show the available bus. Moreover, the option in the top left corner of the webpage allows the user to change the view of the Google Map such as satellite, street and light.

However, this system still lacks some functionality. The system does not display the ETA of the bus. At least, the student can track the bus using this system. Besides, some delay is happen in this system causing the bus icon teleport from one location to another location instead of moving along the route.

2.4.3.2 UM Bus Tracking

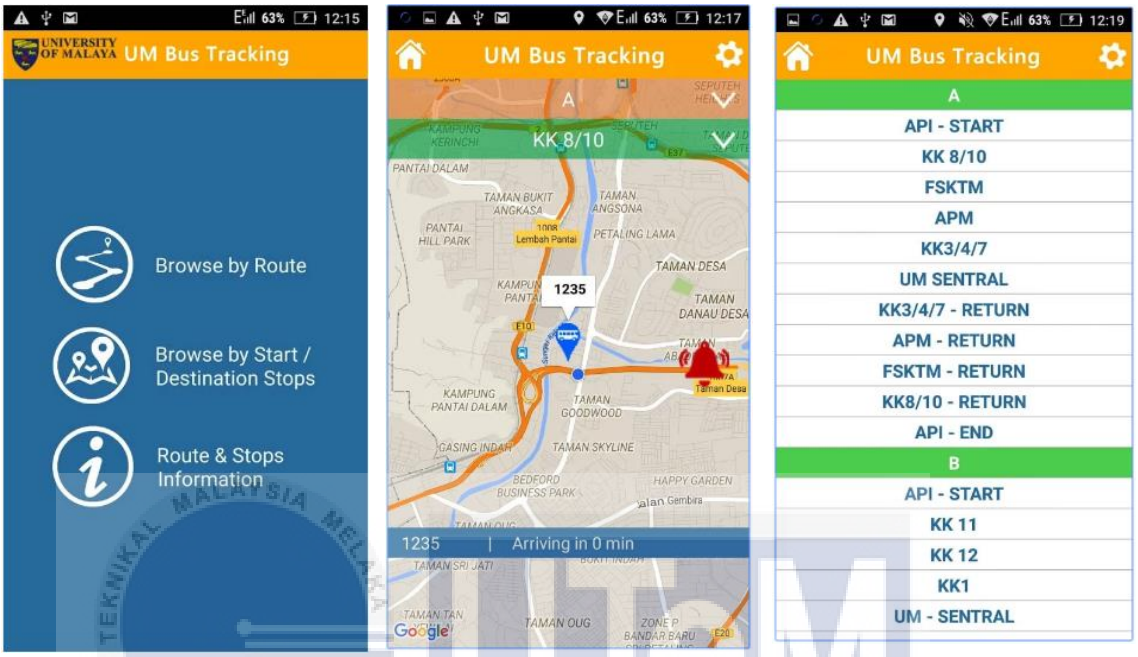


Figure 2.6 Interfaces of UM Bus Tracking application

The bus tracking application shows in Figure 2.6 is used in University of Malaya (UM), Malaysia. This application is Google Map embedded. This application allow the students of UM to track the UM bus, and estimate the ETA of the bus, and set the alarm before the bus reach the bus stop as the students using this application online.

There are bus icon shows in the application and the student can know the ETA of the bus to the bus stop by just select the Route ID and choose the bus stop. Besides, the student can see the route and bus stop along the selected route and know what bus stop is it by tap on the bus stop icon. Lastly, the application allows the student to set the alarm by click the setting icon on the top right corner of the interface. In the setting, the students can set the frequency up to 24 hours.

Although this application is awesome, but there are still some flaws. First, the alarm frequency can set too long which is 24 hours. Second, the bus drivers need to use their devices to send their GPS location to the server. Causing the bus drivers need to buy mobile data plan to use this application and consume a lot of data because it need to send its location in a short interval. Besides, nowadays the student complaints about the application is useless because they cannot track the bus simply due to the bus driver does not sent their location to the server.

This UM Bus Tracking application should be the fundamental design for the propose solution. These is due to the application is operating in the same environment (University Bus Tracking using android application) and the idea is quite similar.

2.4.3.3 Rice University, Texas, USA

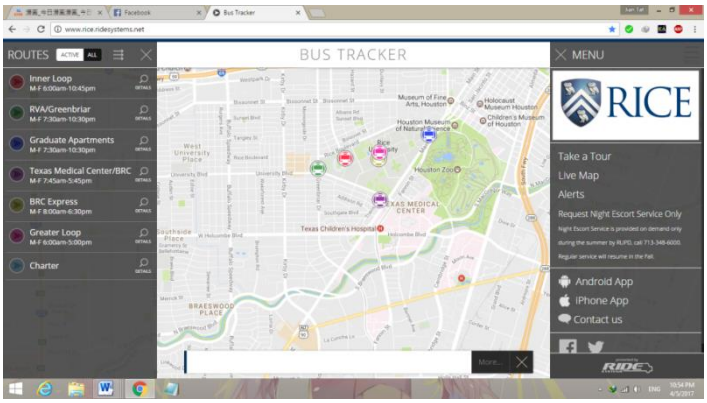


Figure 2.7 Rice University Bus tracking System- main page

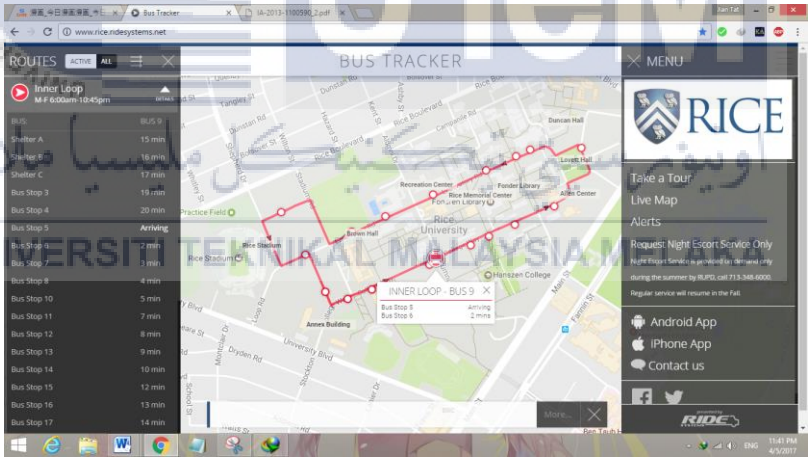


Figure 2.8 Rice University Bus tracking System – Route Inner Loop

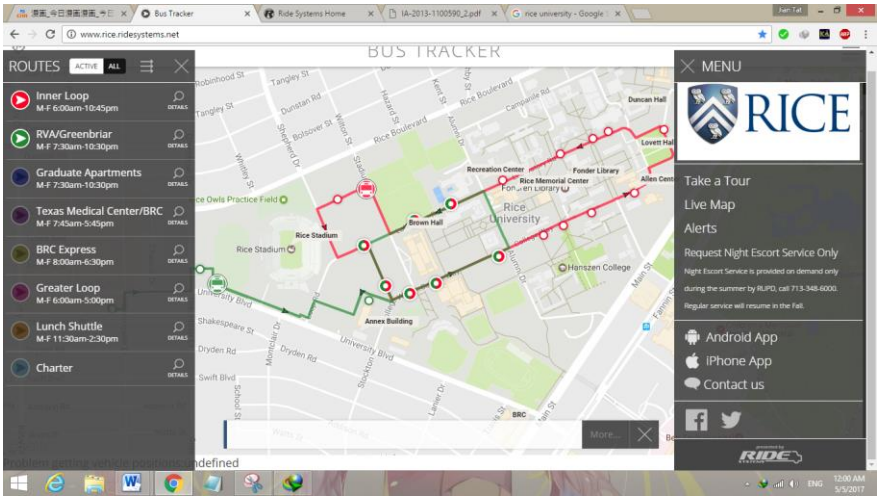


Figure 2.9 Rice University Bus tracking System – Routes Inner Loop and RVA/Greenbriar

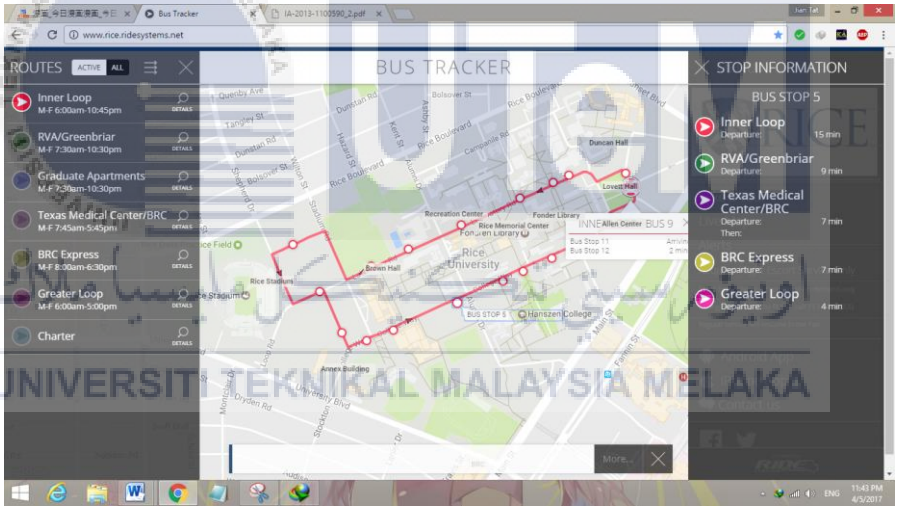


Figure 2.10 Rice University Bus tracking System –Bus stop 5

The Figure 2.7, Figure 2.8, Figure 2.9, and Figure 2.10 show the bus tracking system used in Rice University, Texas, United States. This system is developed by Ride Systems. This tracking system uses webpage which is embedded with Google Map. In this webpage, the students can track the buses easily and know the ETA of the bus. The student can easily access the webpage via URL: http://www.rice.ridesystems.net/ .

This system has many advantages. First, the every color indicate difference route and bus service in the campus .Moreover, the bus location is update very frequently so the students the current location of the bus accurately. Next, Figure 2.8 shows Inner Loop route, the ETA of the bus can easily know by click on the bus icon, detail option next to the “Inner Loop”, and the bus stop icon. Besides, this webpage allow the user to choose more than 1 route, and the route information is very clear. These can be shown in Figure 2.9, the bus stop icon is merge with the other color which show the bus stop allow 2 buses to stop.

Although this webpage is very useful for the student to track the bus, but the android app does not tracking the bus at the Rice University. The android app tracks the bus at Alexandria, US which is totally useless for the Rice University Student.

The webpage of the Rice University bus tracking system is very useful and efficient for the Rice University student, but the android application which tracks the bus outside the campus is totally useless for the student.

2.4.3.4 Northern Illinois University (NIU)

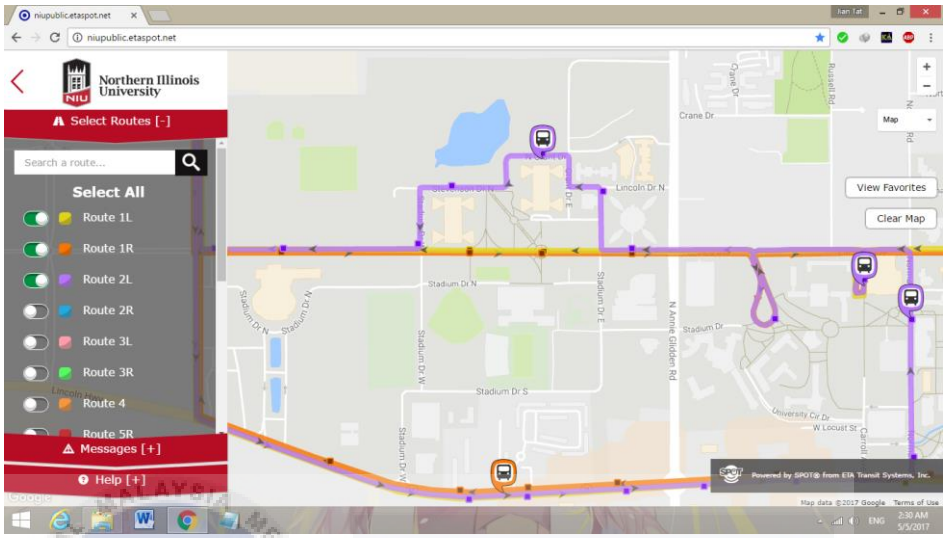


Figure 2.11 NIU Bus tracking System

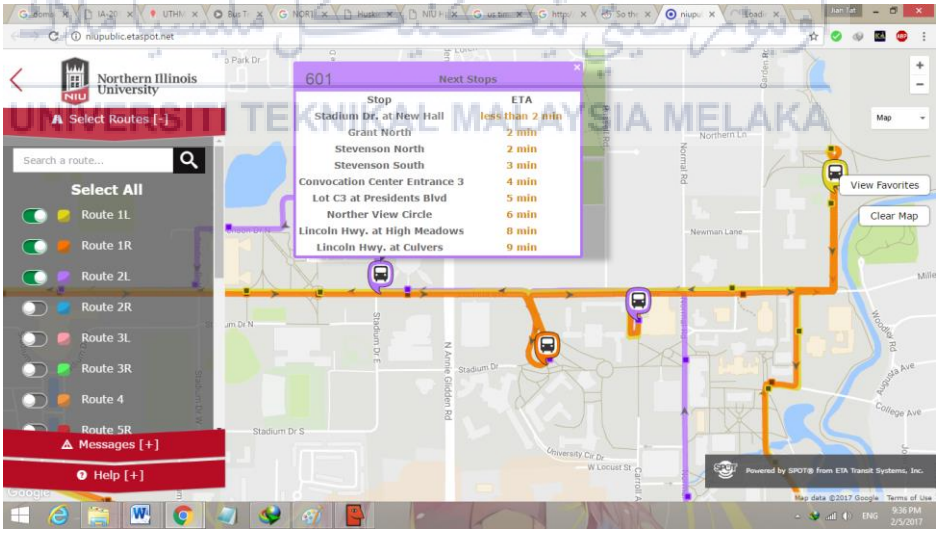


Figure 2.12 NIU Bus Tracking System ETA

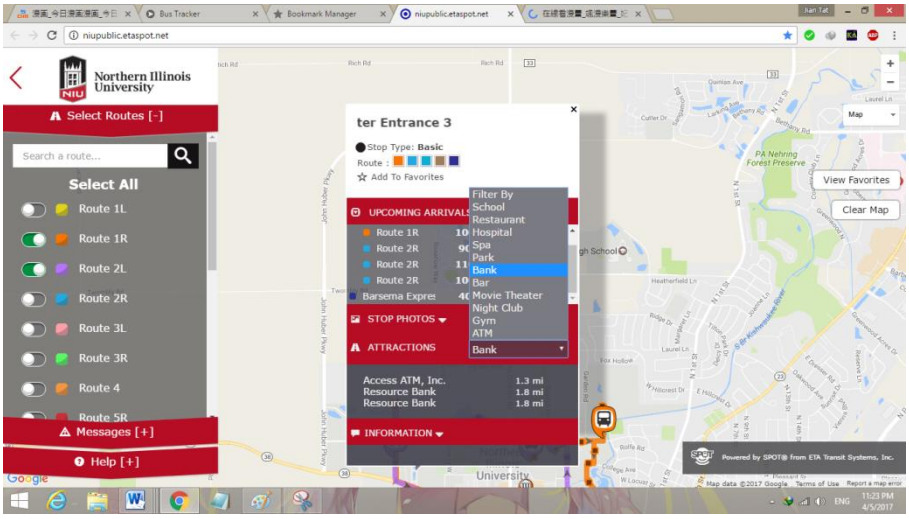


Figure 2.13 NIU Bus Tracking System – Attraction near bus stop

The Figure 2.11, Figure 2.12, and Figure 2.13 show the bus tracking system used by NIU. This system is developed by SPOT from ETA Transit Systems, Inc. This bus tracking system uses webpage which is embedded with Google Map. In this webpage, the students can track the buses easily and know the ETA of the bus. The student can easily access the webpage via URL: http://niupublic.etaspot.net/ .

This system has many advantages. First, the every color indicates difference route and bus service in the campus like NIU bus tracking system. Moreover, the bus location is update very frequently so the students the current location of the bus accurately. Next, Figure 2.13 shows option to check the attraction near the bus stop by simply click on the bus stop icon. Besides, this webpage allow the user to choose more than 1 route, and the route information is very clear. These can be shown in Figure 2.11.

Although this webpage is very useful for the student to track the bus, but when many bus route is chosen, more new bus stop icon is created even the bus stop is overlapped.

2.4.4 Critical review of current problem and justification

By studying the related work, the idea of the proposed solution can be obtained. Table 2.2 shows the summary of the comparison between the existing system.

There are two bus tracking system study in Malaysia which are UTHM and UM. Besides, study also conducted on the bus tracking system in foreign university such as Rice University and Northern Illinois University (NIU) .

Each of them have their own feature. For example the UTHM public shuttle tracking is very easy to use as the user just need to choose the bus route. Besides, the map clearly show the bus location, and the bus stop. UM bus tracking has the feature to set the alarm frequency to alert the student. Rice University bus tracking shows the bus stops very clearly by using different color for different route and mix the color when the bus stop has multiple route. Lastly NIU bus tracking has the feature to find the attraction spot near the bus stop.

Table 2.2 Comparion between the existing system

Aspect	UTHM	UM Bus Tracking	Rice University	NIU
Type of bus tracking system	Website	Android application	Website, android, iOS Application	Website
Show bus icons	Yes	Yes	Yes	Yes
Show bus stops and bus route	Yes	Yes	Yes	Yes

on the map				
Ability of choose multiple map together	No	No	Yes	Yes
Show ETA	No	Yes	Yes	Yes
Alarm	No	Yes	No	No
Bus Information like schedule	No	No	Yes	No

2.5 Proposed Solution

By study the existing systems, the proposed solution is found. The UTeM Bus Tracking System should be has the function to track the bus location near real time, calculate the bus ETA, alarm system to alert the student, and contain the bus information.

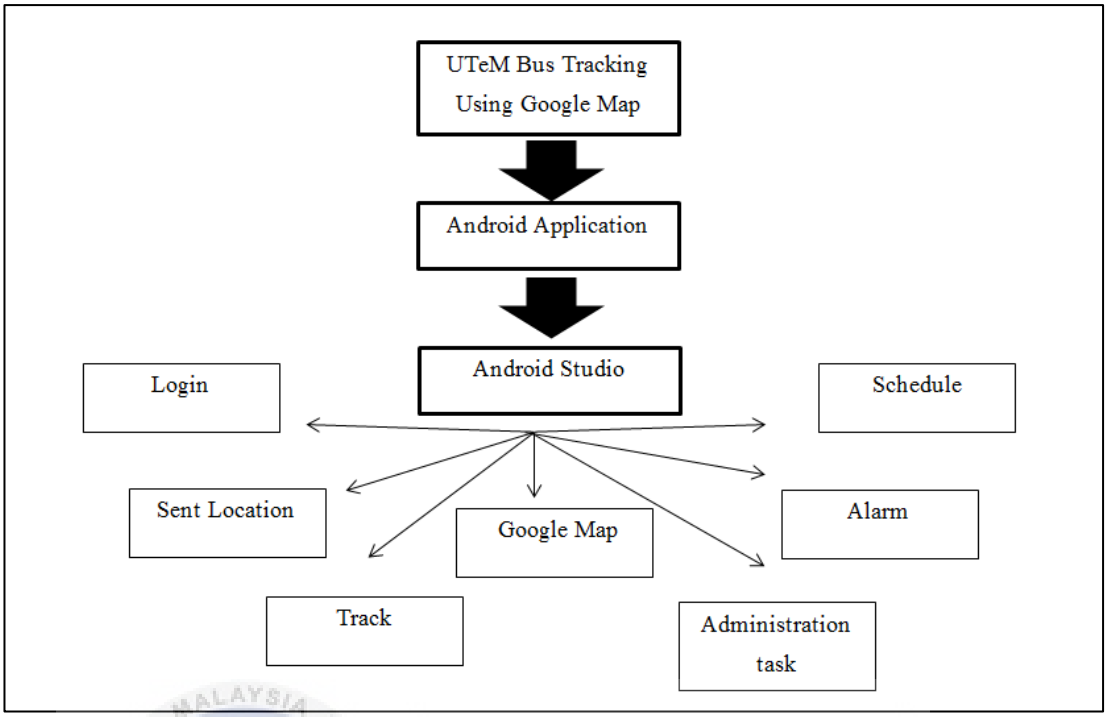


Figure 2.14 Functionality of proposed solution

2.6 Conclusion

In conclusion, literature review is a necessary chapter and it is very important part to build the project concept, it helps to understand the existing features of the system and to get clear picture to implement the system. The research and study will make the progression on doing this project smoothly and more understanding.

Chapter 3

METHODOLOGY

3.1 Introduction

This chapter discuss about the methodology used in this project. Software development methodology is used in this project because the objective of this project is to develop an android application. There are many software development methodologies such as waterfall development, prototyping, rapid application development (RAD), and systems development life cycle (SDLC). Each of the methodology has their own pros and cons. Hence, suitable methodology had to be chosen carefully. In this project, the chosen methodology is waterfall. This is because waterfall gives a review at the end of each phase and keep track on the project progress.

3.2 Research Process

The Research Process is referring to the step-by-step development of the product. There are several stages included which are Data Collection, Analysis, Design, Implementation and Testing process. Every stage is important for carry out the project.

3.2.1 Data Collection

Based on the previous research, Android app is a software application running on the Android platform. Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS. There are four application components that make up the android application. The components are Activity, Content provider, Broadcast Receiver and Service.

Besides, there are two IDE to develop the android applications which are android studio and Eclipse. The chosen IDE is android studio. This is due to android studio support the Google Cloud Platform which is very important to use the Google Map API, it also supports the app testing and debugging.

3.2.2 Analysis

Analysis is done on the bus tracking system in Malaysia and foreign country. In Malaysia, there are UTHM public shuttle tracking and UM Bus tracking. UTHM public shuttle tracking is a website based bus tracking system. In the web interface, the user can know the bus location and track the bus in real time. While, UM Bus tracking is android application based bus tracking system. The feature of this

application is it does not just track the bus location, it also allows the user to set the alarm frequency.

In the foreign country, the analysis is done on Rice University and Northern Illinois University (NIU). Rice University bus tracking system is available in website, android, and iOS application. The website is very intuitive for the user as the user can select the bus routes and track the bus they desire. It also has the bus estimated time of arrival shown. However, the android and iOS application is track the bus outside of the campus. Therefore the application is useless for the students. Next, the NIU use website based bus tracking system. This web is function similar with RICE University, but it has his own feature which allows the user to know the attraction around the bus stop.

Based on the analysis, the bus application should have the feature to allow the user to set the alarm for the bus, and track the bus.

3.2.3 Design

System Architecture Design

Figure 3.1 shows the flow of the product. First, the bus driver need to use android application to get the GPS position of the bus and sent the data to the database server in the internet. When the student uses the android application to track the bus, the application will request the location of the bus. The project design details will be discussed in Chapter 4.

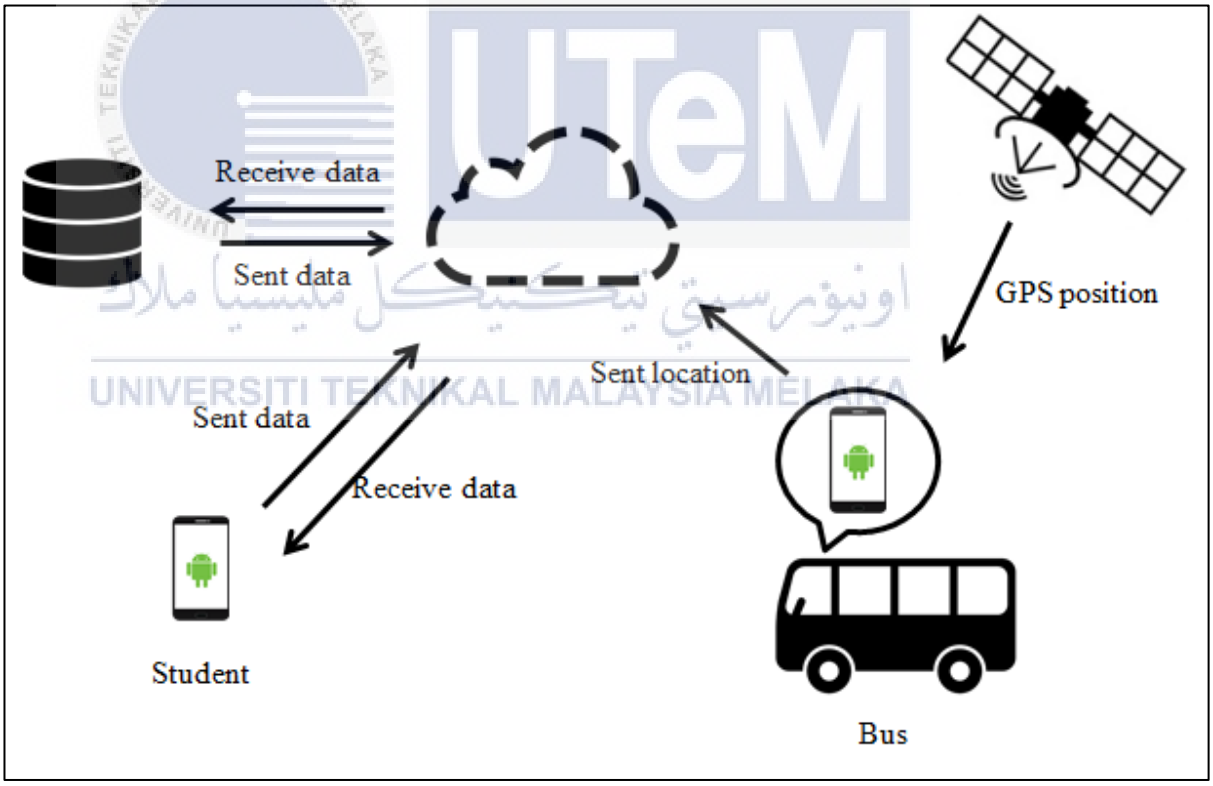


Figure 3.1 System Architecture

3.2.3 Implementation

Project implementation describes how to develop the system in reality. This project is implemented using Android Studio to create the android application. The detail of implementation is discussed in chapter 5.

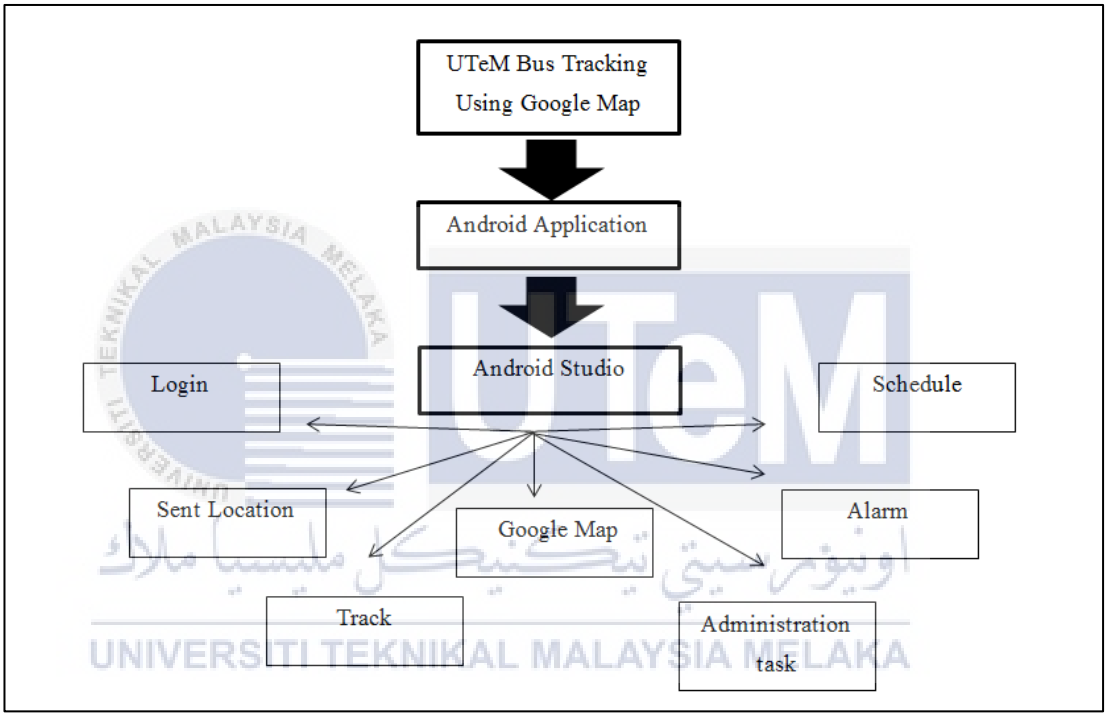


Figure 3.2 Functionality of UTeM Bus Tracking Using Google Map

Figure 3.2 show the functionality of the UTeM Bus Tracking System using Google Map. Android Studio is used to create the android application. This android application allows the bus driver to login into the system. After successfully login into the system, the driver will sent the bus location to the database server. By using the location sent to the server, the student can track the bus location. This should be show in the Google Map because it is intuitive for the student to use. Besides, the student can set the alarm clock for the time before the bus reached the certain bus

stop. Next, the student can view the bus schedule in the android application. Lastly, the student can sent complaint to the server when the student is not satisfy with the bus service.

3.2.4 Testing

Testing is an important process to make sure the system is running well. In order to execute the testing for the project, the procedure must be plan well and the result is recorded correctly. The software is separate to many modules to test. Each module is test with difference approach to test the stability of the module. One bus driver is needed to test the bus driver modules and 10 students are required to test the student modules. After the testing, the testers need to fill in the questionnaires to record the result. The testing detail is discussed in the Chapter 6.

3.3 Theory Structure

Theory structure explains from a big picture to the specific scope. Each part is briefly explained to give a basic idea. Figure 3.3 shows the theory structure of this project.

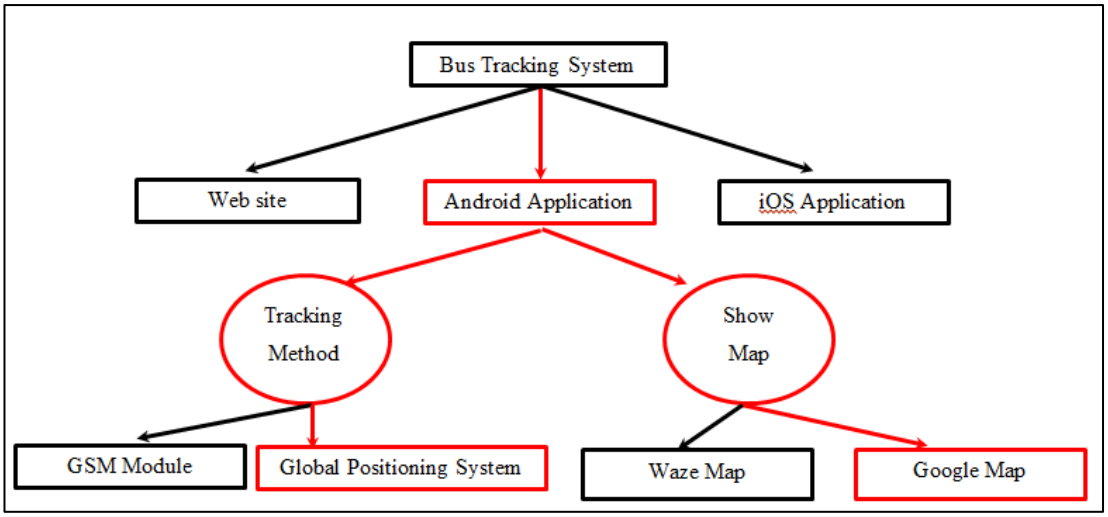


Figure 3.3 Summary of theory structure

3.3.1 Bus Tracking System

A bus tracking system is the combination of the use of vehicle bus and the software that retrieve the data, software that manage the data to do function. Website, android application, and iOS application is the example for the bus tracking system.

3.3.2 Tracking Method

Tracking method is the way to retrieve the location of the vehicle. There are two commonly used methods which are Global Positioning System (GPS) and GMS Module. GMS Module is a specification of wireless network infrastructure which the object's position is determined using signal strength and triangulation from base stations. While GPS is satellite-based navigation system.

3.3.3 Show Map

Show Map is the method to show the vehicle position.

3.4 Methodology

In this project, waterfall model is implemented. This methodology is the suitable as this project is need to meet the user expectation and within time estimation.

Waterfall is a six-phase software development cycle. The six phases are Requirement Analysis, System Design, Implementation, Testing, Deployment and Maintenance. The figure 3.4 shows the phases in waterfall model.

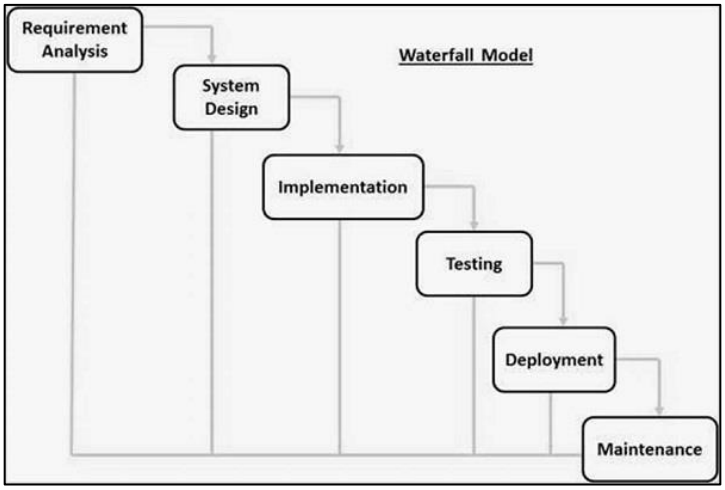


Figure 3.4 Phases in Waterfall Model

3.5 Research Technique

To develop the UTeM Bus Tracking Using Google Map, android studio is chosen as the technique to create the android application. Android Studio is the official integrated development environment (IDE) for the Android platform. In android studio, GPS is used to retrieve the location of the bus. Besides, Google Map API is used to implement the Google Map feature into android application. Moreover, Google Maps Distance Matrix is used to get the bus estimated time of arrival.

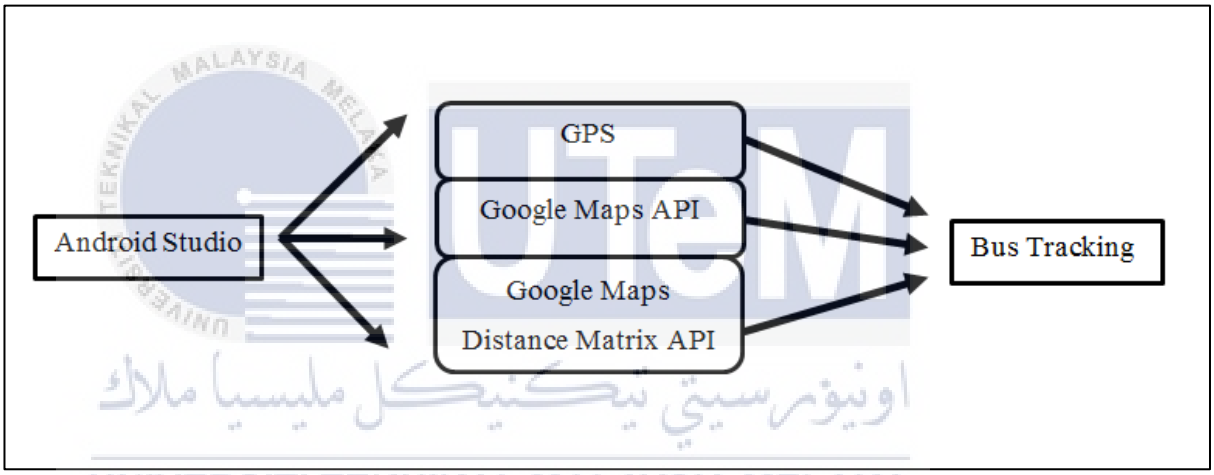


Figure 3.5 Research Technique

3.6 Research Framework

In order to carry out the project, a framework is design to have a big picture about the project. First, identify the problem. In this project, the problem is the student does not know the bus ETA and have no idea about bus schedule. After the problem is identified, research is done to analysis the solution to overcome the

problems in Malaysia and foreign country. Next, study is conducted to gain the knowledge about the theory in this project such as IDE to create android application, what is technique to track the location, and the method to calculate the ETA. This knowledge is important as theory is needed to implement the method in the project. The strategies in this project are including the waterfall methodology, android studio, Google Map API, alarm system. These strategies are chosen in order to develop the bus tracking system.

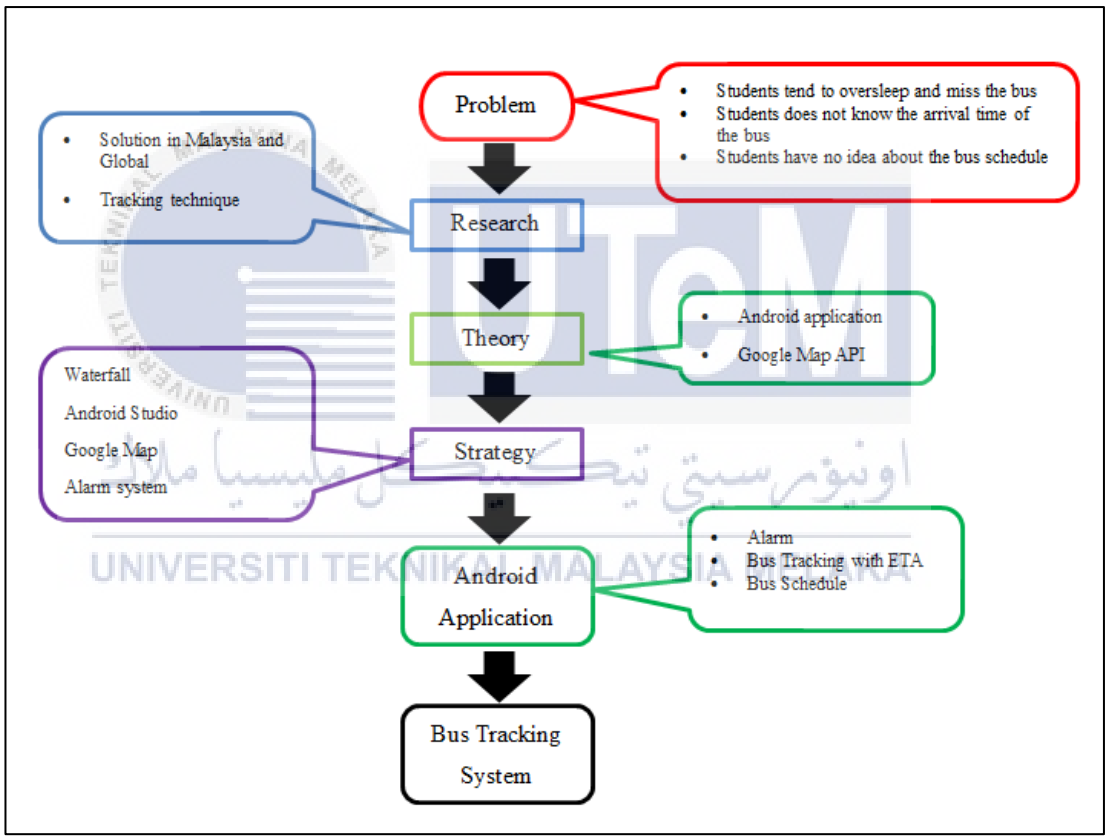


Figure 3.6 Project Framework

3.7 Research Requirement

To carry out this project, some hardware and software requirement have to be fulfilled to ensure that the project can be progress smoothly.

Hardware requirements:

In this project, there is hardware needed to complete the project. This hardware is needed to conduct the project.

1. Laptop with at least 4GB RAM, this is the recommended for the android studio to ensure the development process run smooth without any delay or crash due to lack of RAM.

2. Android phone with minimum android sdk version of 10, this is needed to use the Google Map API.

Software requirement:

In this project, software is needed to develop that android application and web application.

1. Android Studio. To create android application

2. Android SDK: To provide the API libraries and necessary developer tool for android studio

3. Wamp Server: To provide the connection to database, and act as a database in local area network.

3.8 Project Milestone

Project Milestone is used as reference point to monitor the project's progress and marks the major activity in a project.

In order to make sure the flow of this project progress smoothly, the project milestone is created and well planned to ensure all the activities in the project are able to be completed within the project timeline.

Gantt chart will be able to track the time of every progression of the chapter to ensure that all tasks can be completed on the given time. Figure 3.7 shows the summary of the Gantt Chart table.

Task Name	Start Date	End Date	Duration
Proposal PSM	02/13/17	02/17/17	1 Week
Chapter 1: Introduction	02/17/17	03/03/17	1 Week
Chapter 2: Literature Review	03/06/17	03/17/17	2 Weeks
Chapter 3: Project Methodology	03/20/17	03/24/17	1 Week
Chapter 4: Analysis and Design	03/27/17	04/10/17	3 Weeks
Project Demo	04/17/17	04/21/17	1 Week
PSM 1: Final Presentation	05/22/17	05/26/17	1 Week
Chapter 5: Implementation	05/29/17	06/19/17	4 Weeks
Chapter 6:Testing	06/20/17	06/30/17	2 Weeks
Chapter 7:Conclusion	07/03/17	07/07/17	1 Weeks
PSM 2: Final Demonstration (Product & Report)	08/07/17	08/24/17	3 Weeks

Figure 3.7 Summary of Gantt Chart table

3.9 Conclusion

As conclusion, this chapter discusses the methodology used in this project. Waterfall model consists of several phrase and each phrase has own objective in order to done the project within the time limit. Besides, the milestone is set to keep track of the progress of the project.

Chapter 4

DESIGN

4.1 Introduction

In this chapter defines the results of the analysis of the preliminary design and the result of the detailed design. It will also focus on the analysis of the requirements of the project.

The requirements include the hardware and software needed on this project. The block diagram architecture and proper analysis in detail for this project is stated to ensure the project can be completed and well designed.

4.2 Problem Analysis

The student now cannot know the location of the bus as the bus may be late or leave earlier. Therefore, bus tracking system will be implemented in this project is

a software application create by Android studio to keep track of the bus. Besides, this project also implements alarm system to prevent the student overslept and miss the bus.

4.3 Requirement Analysis

Requirement analysis is the process of determining user requirement or expectation for a system.

4.3.1 Data Requirement

Figure 4.1 show the flow of data in the system. The bus driver application retrieves the GPS location of bus by using the GPS receiver. Then the bus driver application sent its location to the database via internet. The location data is saved into the database. When the student wants to track the bus, it will request the bus location information from the database server via Internet.

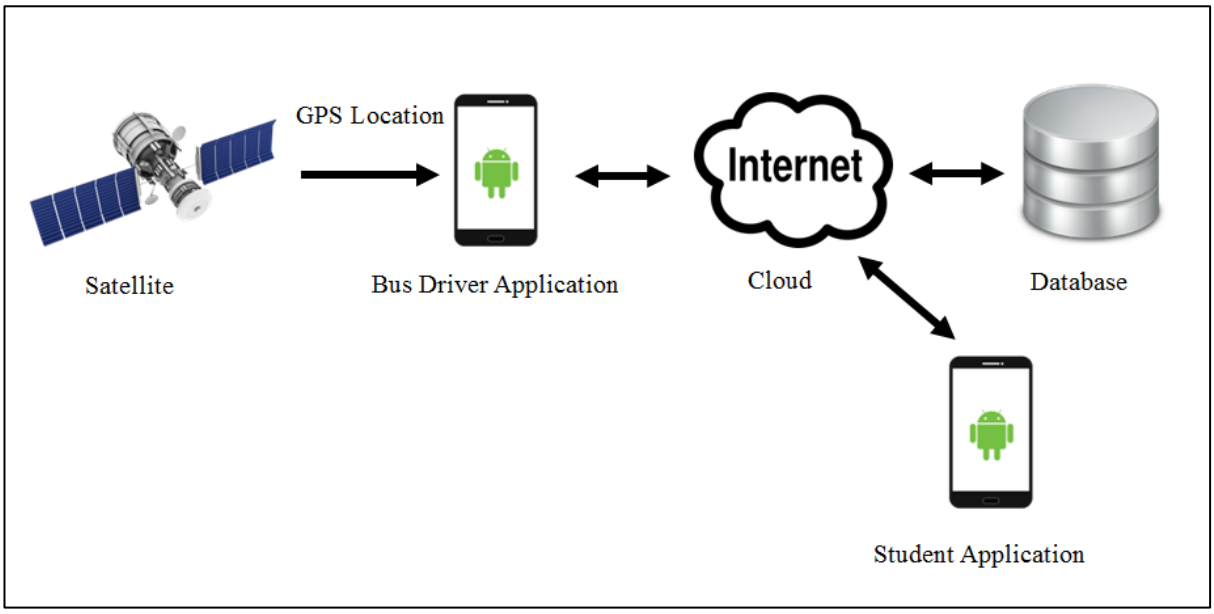


Figure 4.1 Data Flow

4.3.2 Functional requirement

Figure 4.2 shows the block diagram for this project. The project is separated into several blocks namely GPS location block, driver android phone block, student android phone block.

1. GPS location block

There are 31 active satellites on the orbit and most of the place is view by 6 satellites. When GPS receiver like an android phone request its location, the satellite will work with other satellite to calculate the location.

2. Driver Android Phone

The driver android phone is installed with an android application which will request the GPS location. After, the application retrieve the location data it will sent the data to the database via internet.

3. Student Android Phone

The student android phone is installed with and android application which can request the location data from the database. After the application get the location data, the bus location will display in Google Map. Besides, the application will calculate the bus ETA to let the student know when the bus will arrive.

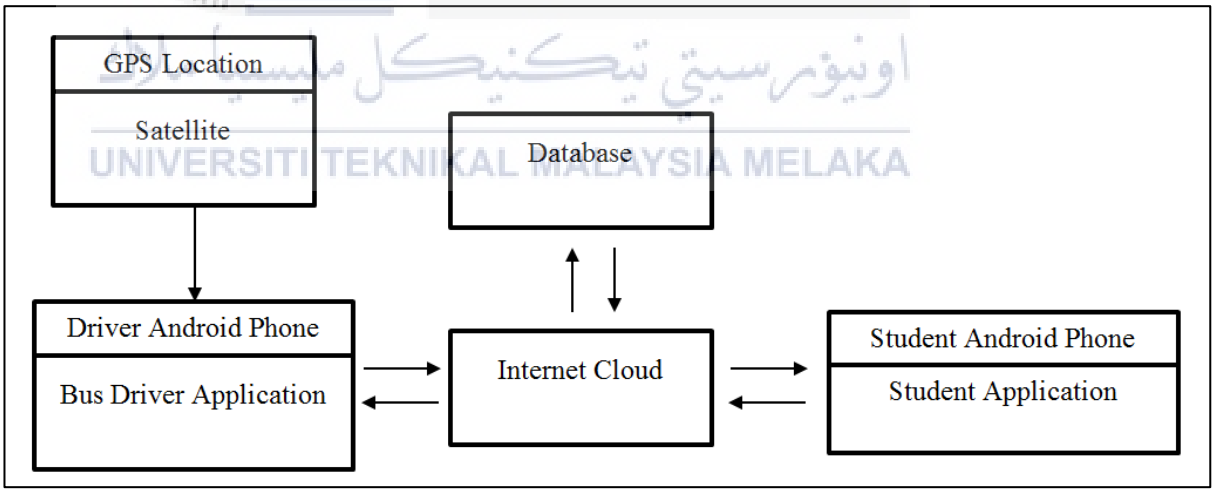


Figure 4.2 Block Diagram

4.4 Hardware requirement

In order to do the project some hardware equipment are needed. This equipment plays a big role as tool to complete the project

1. Lenovo G40, AMD 48-6410 APU, 6.00 GB RAM

A laptop play an important role in this project as it can use to develop software and write project report. A minimum RM of 6GB is needed to run the Android Studio smoothly. If the RAM is less than 6.000GB, the emulator will take a very long time to start. These can cause the progress to be delay.

Figure 4.3 Lenovo G40

2. XiaoMi Redmi Note 2

An android phone is needed as the testing need to be done in android phone. Redmi note 2 is recommended as it has androidSDKVersion of 19. This androidSDKVersion is very crucial as the Google Map API need a

minimum of androidSDKVersion of 14 to run. If less than androidSDKVersion of 14, the Google Map will not be generated.

Figure 4.4 Redmi Note 2

4.5 Software Requirement

In order to implement the project the software are needed. This software plays important role as tool to develop the system.

1. Android Studio

Android Studio is an official IDE for Android platform. Android Studio allow the user to develop their own application. Besides, many Google API key can be easily used in Android Studio. Figure 4.5 shows the icon for android studio.

Figure 4.5 Android Studio

2. Android SDK

Android SDK allows mobile software developers the opportunity to tinker with the platform and create new and interesting work. The kit contains everything to start building apps. The Android SDK also comes with an emulated virtual device that is fully functional to for testing and debugging.

Figure 4.6 Android SDK

3. Wamp Server

Wamp Server has Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language. These components are crucial to establish

connection between the devices with the server. Besides, it also has database which can save the information.

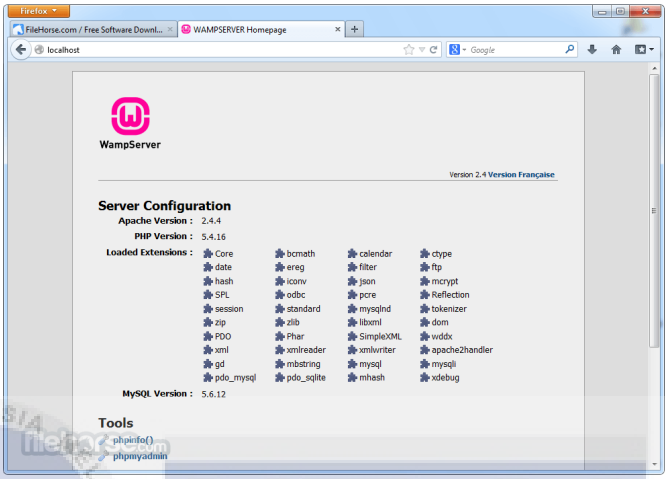


Figure 4.7 Wamp Server

4.6 High Level Design

4.6.1 System Architecture

Figure 4.8 shows the system architecture of the UTeM Bus Tracking Using Google Map. Android phone is used as GPS receiver to get the GPS location from the satellites. This information play a huge role as the student need to get this data to know the bus location. Therefore the data is store in the database and the student can easily retrieve it using the android application.

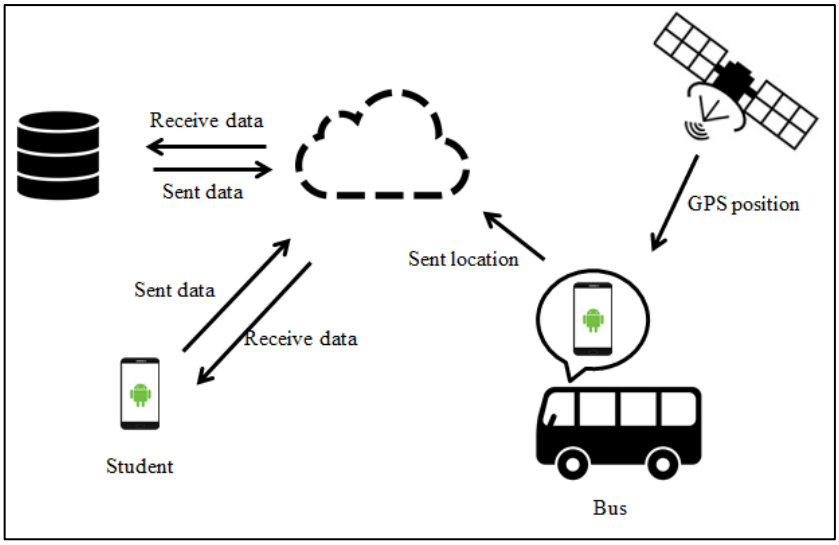


Figure 4.8 System Architecture

4.6.2 Interface Design

A proposed user interface is designed for the UTeM Bus Tracking Using Google Map. These interfaces allow the user to have some basic idea on how the system is works. Figures show the user interface of the project.

Figure 4.9 shows the main page interface for the system. There is a short description shown. Besides, the menu button at the top left corner will make the navigator drawer to appear.

Function:	
Menu Button: 	Move to navigation drawer interface (Figure 10)

Figure 4.9 MainPage Interface

Figure 4.10 shows the navigator drawer. There are three options, which are login, map and track, and view schedule. If either of the option is selected the page will move to the selected interface, such as select login option to go login interface.

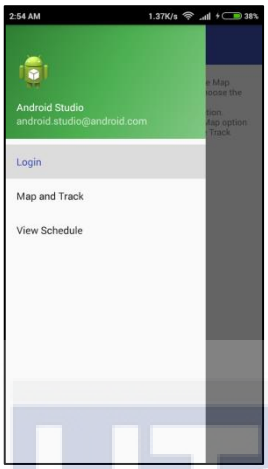


Figure 4.10 Navigation Drawer Interface

	Function:
Login option: Login	Move to Login Form interface (Figure 4.11)
Map and Track option: Map and Track	Move to Map and Track interface (Figure 4.17)
View Schedule option: View Schedule	Move to View Schedule interface (Figure 4.18)

Figure 4.11 shows login interface. The user needs to fill in the username and password in order to login by pressing Login button. There are two type of user for the login function, which are admin and driver. Upon successful login, the system will bring the user to respective type of user.

Figure 4.11 Login Form

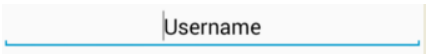
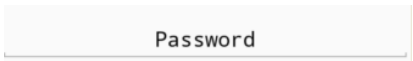
Function:	
Username text field: 	Text field to insert the username for login purpose.
Password text field: 	Text field to insert the password for login purpose
Login button: 	Validate the username and password with the server. If the username and password match with the server, move to the Admin Action interface (Figure 4.12) for Admin position, or move to Driver Action interface (Figure 4.16) for driver position.

Figure 4.12 show the Admin Action Interface. There are four options in this interface. When user clicks either “ADD”, “UPDATE”, or “DELETE” button, the system will move to the selected interface. While click “LOG OUT” button will end the user session and move to the main page interface.

Figure 4.12 Admin Action Interface

Function:	
ADD button: 	Move to Admin Add Interface (Figure 4.13)
UPDATE button: 	Move to Admin Update Interface (Figure 4.14)
Delete button: 	Move to Admin Delete Interface (Figure 4.15)
LOG OUT button: 	End the user login session and move to MainPage interface (Figure 4.9)

Figure 4.13 shows the Admin Add interface. User need to fill in all the information in the textfield to add the information into the database when click the "ADD" button. While clicks the "BACK" button will move to Main Page Interface.

Figure 4.13 Admin Add Interface

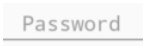
	Function:
Staff ID text field: 	Text field to insert the Staff ID, character and number can be inserted.
Name text field: 	Text field to insert the Name, only character can be inserted.
Password text field: 	Text field to insert Password, character and number can be inserted. .
Position drop down menu:	Drop down menu with "Admin" option and "Driver" option to be selected as staff position.
Route id drop down menu:	Drop down menu with "EP_BR_KT_KI", "SU_KT_KI", "SU_KB", "MK_MT_KI", and "MK_KB" options to be selected for route id.
ADD button:	When click, add all the information in the text fields and drop down menu into the server.
BACK button:	Move to Admin Action Interface (Figure 4.12)

Figure 4.14 shows the Admin Update Interface. User can change the information in the database by change the informaion shown in the field when click the “UPDATE” button. While clicks the “BACK” button will move to Main Page Interface.

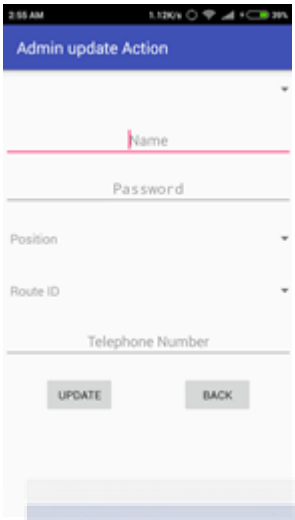


Figure 4.14 Admin Update Interface

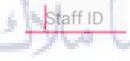
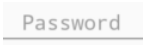
	Function:
Staff ID drop down menu: 	Drop down menu consist the staff information in the server. If any Staff ID item is selected, all the other fields and drop down menus will change based on the information in the server.
Name text field: 	Text field to hold the Name, only character can be inserted.
Password text field: 	Text field to hold Password, character and number can be inserted. .
Position drop down menu:	Drop down menu with “Admin” option and “Driver” option to be selected as staff position for update purpose.
Route id drop down menu:	Drop down menu with “EP_BR_KT_KI”, “SU_KT_KI”, “SU_KB”, “MK_MT_KI”, and “MK_KB” options to be selected for update route id.
UPDATE button: 	When click, update all the information in the text fields and drop down menu into the server for the selected staff id.
BACK button:	Move to Admin Action Interface (Figure 4.12)

Figure 4.15 shows the Admin Delete Interface. The user can remove the information from the database by select the staff id shown in the drop down menu then clicks the “DELETE” button. While click “BACK” button will move to Main Page Interface.

Figure 4.15 Admin Delete Interface

Function	
Staff ID drop down menu: admin	Drop down menu consist the staff information in the server. If any Staff ID item is selected, all the other fields will change based on the information in the server to display the selected staff information.
DELETE button: DELETE	When click, the selected staff id with the corresponding staff information will be deleted in the server.
BACK button: BACK	Move to Admin Action Interface (Figure 4.12)

Figure 4.16 shows the Driver Action Interface. The driver can start sending his/her GPS location to the server every four second by click the “Send Location”. Besides, the driver needs to click “STOP SENDING LOCATION” to stop sending the driver GPS location into server. Moreover, driver can click the “LOGOUT” button to end driver’s session and move to MainPage Interface.

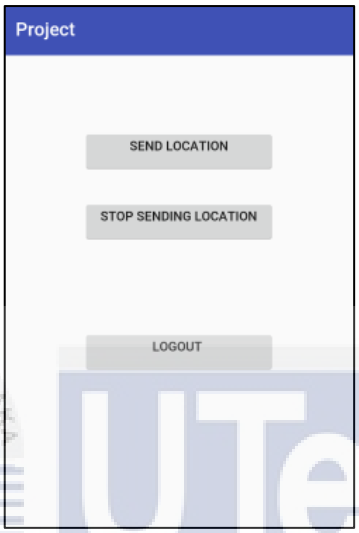


Figure 4.16 Driver Action Interface

	Function
SEND LOCATION button: 	When click, the system will get the GPS location and send it to server
STOP SENDING LOCATION button: 	When click after clicked the “SEND LOCATION”, will stop sending the GPS location to the server.
LOGOUT button: 	When click, move to MainPage Interface (Figure 4.9)

Figure 4.17 shows the Map and Track Interface. User needs to choose the desire bus route and bus stop then clicks the “TRACK” button to start tracking the bus. While clicks “STOP” button will stop the tracking process. Besides, the user can set the alarm by checked any check box.

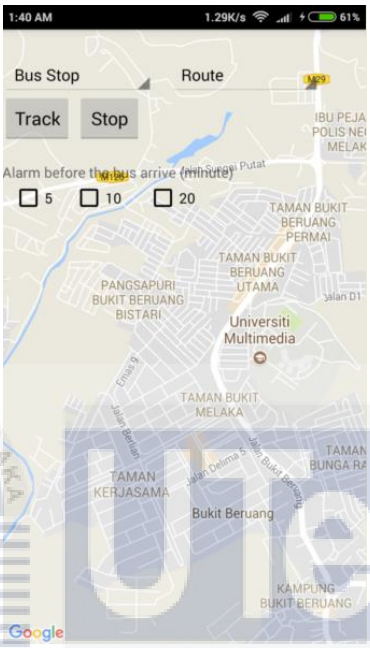
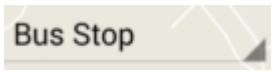


Figure 4.17 Map and Track Interface

Function	
BusStop drop down menu: 	Drop down menu consist the list of the bus stop. The items in the drop down menu are “KI”, ”KT”, “KB”, “EP”, “BR”, “SU”, “MK”. These items are important to determine student wait at which bus stop.
Bus Route drop down menu: 	Drop down menu consist the list of the bus route. The items in the drop down menu are “EP_BR_KT_KI”, ”SU_KT_KI”, “SU_KB”, “MK_KT_KI”, and “MK_KB”. These items are important to determine which bus route to track. When item is selected, the corresponding bus route and the bus stop icon is show in the Google Map.

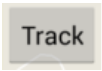
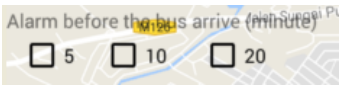
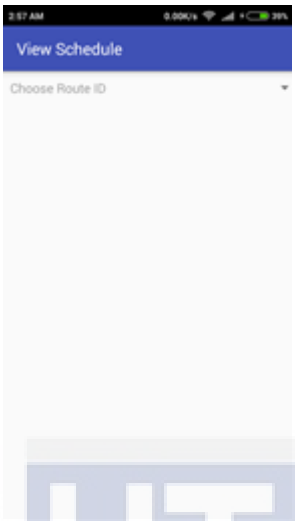
Track button: 	When clicked, the GPS location for the running bus is obtained. Next, the GPS location of the bus and bus stop is used to calculate the bus arrival time and show at the top bus stop icon.
Stop button: 	When click after clicked the Track button, the system will stop requesting bus GPS location and no calculation in bus arrival time. Finally, system clears all the bus route and bus stop icon in the Google Map.
Alarm check boxes: 	The application need to track the bus before check any alarm check box. When Checkbox 5 is checked, the alarm will ring when the bus arrival time is less than 5 minutes. When Checkbox 10 is checked, the alarm will ring when the bus arrival time is less than 10 minutes. When Checkbox 20 is checked, the alarm will ring when the bus arrival time is less than 20 minutes.

Figure 4.18 shows the view schedule interface. User can choose the route id from the drop down menu to view the desire bus schedule.



Function	
Choose Route ID drop down menu: Choose Route ID	This drop down menu consists "EP_BR_KT_KI","SU_KT_KI", "SU_KB", "MK_KT_KI", and "MK_KB". When any item is selected, the bus schedule for the chosen route id will be shown.

4.7 Conclusion

Analysis and design is one of the important phrases to implement in a project. All software and hardware requirements need to be identified and studied before carry out a project. This chapter is the pre-preparation stage for the implementation and it also includes the flow of the overall system so that to have a better understanding before implement it. The next chapter Implementation will discuss how the project to be implement and the output expected for this project.

Chapter 5

IMPLEMENTATION

5.1 Introduction

In this chapter, the process of how to implement the UTeM Bus Tracking System using Google Map is discussed. The process involve is focus on how to setup this project in term of hardware and software.

5.2 Software Development Environment Setup

Development environment is a set of technique and programming tools used to develop a software product. The software development environment setup involves the hardware, software, and network configuration. There are three kinds of users involve in this project, which are bus admin, bus driver and student. All of the users communicate to server using the Android smart phone which installed with the project.apk. Figure 5.1 shows the deployment diagram for UTeM Bus Tracking System using Google Map.

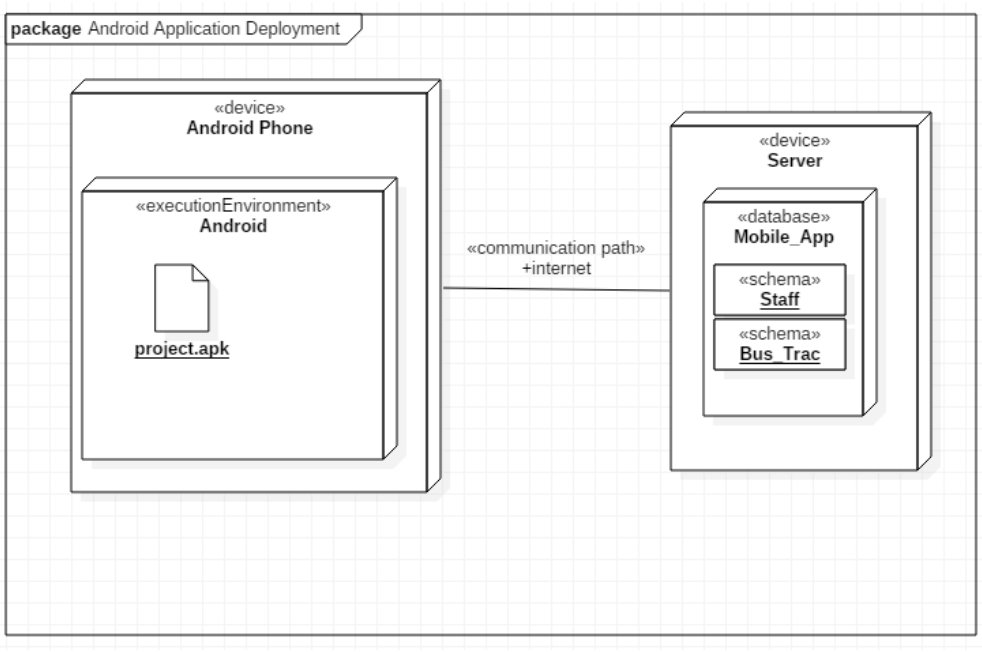


Figure 5.1 Deployment diagram

The hardware configuration is the details and system resource settings allotted for a specific device. In this project, there are two main hardware devices, which are the android smart phone and the server. The hardware configuration for devices used in this project is shown in Table 5.1.

Table 5.1 Hardware configuration

Item	Requirement	Hardware Configuration
Laptop (Lenovo G40)	Processor	AMD A8-6410
	RAM	6GB
	NIC	Realtek PCIe GBE Family Controller
Android smart phone (Redmi note 2)	Android version	4.4.4 KTU84P
	Processor	ARMv7 Processor rev 3

	CPU	Quad-core Max 1.6GHz
	RAM	2GB
	GPS	Available
Server (Smartgreen.my)	Server type	MySQL
	Server version	5.6.30-cll-lve - MySQL Community Server (GPL)
	Protocol Version	10
	phpMyAdmin	4.0.10.14

The software configurations that are used to setup an android development environment are as follow:

a. Android Studio

Android Studio is an official IDE for Android platform. Android Studio allows the user to develop own application. Besides, many Google API key can be easily used in Android Studio.

b. Android SDK

Android SDK allows mobile software developers the opportunity to tinker with the platform and create new and interesting work. The kit contains everything to start

building apps. The Android SDK also comes with an emulated virtual device that is fully functional to for testing and debugging.

c. Wamp Server

Wamp Server has Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language. These components are used to connect the device with the server. Besides, it also has database which can save the information.

d. Web Hosting Server

Web Hosting Server works like Wamp Server but unlike Wamp Server which only work on localhost and local area network. Web Hosting Server can be accessed through wide area network. However, there are some limitations in free web hosting server as there is a limited capacity for the database and the availability of the server uptime.

5.3 Software Configuration Management (SCM)

SCM refers to the software engineering discipline, which consists of standard procedures that are used to manage all the changes that are introduced to the software. Having SCM in a project can help to improve the productivity by controlling its basic components such as coding and test data. Besides that SCM also helps to track

and control the changed that are made to the particular software though reliable version control. SCM consists of two major components which are configuration environment setup and version control procedure.

5.3.1 Configuration environment setup

UTeM Bus Tracking using Google Map is an application creates using Android Studio. Android Studio is running in Java Runtime Environment (JRE). Therefore JRE need to be setup to run Android Studio. Besides, this application needs to communicate with server to exchange information. Hence, there is a need to create a class in the java file in the android and php file in the server to exchange the information. Following show the methods to setup the JRE and database connection.

A. Setup Java Runtime Environment

1. Download the JRE file from http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
2. Install the JRE.
3. Find the path to the JRE bin file in the local disk C, and copy the path.
4. Search “system” in the window explorer and then choose advanced system setting.
5. Click “Environment Variable” button on advanced tab.
6. Search for the “Path” in the system variable and click on edit.
7. Add “;” and paste the path copied at step 3 at the end of the statement.

8. Click “OK” and “Apply”

B. Connection from android application to web hosting server

1. Create a java class name “BackgroundTask” which responsible to connect to database.
2. Declare the URL link to access the database. Figure 5.2 shows examples of URL.

```
String login_url = "http://smartgreen.my/[TARISA]/class%20session/bus/login.php";
String sentLocation_url ="http://smartgreen.my/[TARISA]/class%20session/bus/sentLocation.php";
String getLocation_url = "http://smartgreen.my/[TARISA]/class%20session/bus/getLocation.php";
String addStaff_url ="http://smartgreen.my/[TARISA]/class%20session/bus/addStaff.php";
String deleteStaff_url="http://smartgreen.my/[TARISA]/class%20session/bus/delStaff.php";
```

Figure 5.2 URL link

3. Access the database using “HttpURLConnection” method. Figure 5.3 shows the example of access the login.php using “HttpURLConnection”.

```
try {

    URL url = new URL(login_url);
    HttpURLConnection httpURLConnection=(HttpURLConnection)url.openConnection();
    httpURLConnection.setRequestMethod("POST");
    httpURLConnection.setDoOutput(true);
    httpURLConnection.setDoInput(true);
    OutputStream os = httpURLConnection.getOutputStream();

    BufferedWriter bufferedWriter= new BufferedWriter(new OutputStreamWriter(os,"UTF-8"));
    String data = URLEncoder.encode("login_name","UTF-8")+"="+URLEncoder.encode(login_name,"UTF-8")+"&"+
            URLEncoder.encode("login_pass","UTF-8")+"="+URLEncoder.encode(login_pass,"UTF-8");

    bufferedWriter.write(data);
    bufferedWriter.flush();
    bufferedWriter.close();
    os.close();

    InputStream inputStream=httpURLConnection.getInputStream();
    BufferedReader bufferedReader= new BufferedReader(new InputStreamReader(inputStream,"iso-8859-1"));

    StringBuilder stringBuilder = new StringBuilder();
    while((JSON_STRING = bufferedReader.readLine())!=null){
        stringBuilder.append(JSON_STRING+"\n");
    }
    bufferedReader.close();
    inputStream.close();
    httpURLConnection.disconnect();

    return stringBuilder.toString().trim();

} catch (MalformedURLException e) {
```

Figure 5.3 HttpURLConnection method

C. Connection from web hosting to android application

1. Create a database name “Bus_Track” using SQL statement “ CREATE DATABASE *Bus_Track*; ”
2. Create necessary table using SQL statement. Figure 5.4 , shows the SQL statement to create Table Staff.

```
CREATE TABLE IF NOT EXISTS `Staff` (
  `Staff_ID` varchar(8) NOT NULL,
  `Staff_Name` varchar(30) NOT NULL,
  `Staff_Post` varchar(10) NOT NULL,
  `Pwd` varchar(10) NOT NULL,
  `Tel_No` int(11) NOT NULL,
  `Route_ID` varchar(15) NOT NULL,
  PRIMARY KEY (`Staff_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Figure 5.4 SQL to create table Staff

3. Create a php file named "init.php" to manage the connection to database. Figure 5.5 shows the init.php file.


```php
<?php

$db_name='smarnmy1_mobileApp';
$mysql_user='PsM2zero1tujuh';
$mysql_pass='PsM20one7';
$server_name='smartgreen.my';

$con=mysqli_connect($server_name,$mysql_user,$mysql_pass,$db_name);

if(!$con)
{
    echo"<h3>failed to connected</h3>";
}
else
{
    //echo"<h3>connected</h3>";
}

?>
```

Figure 5.5 init.php code

4. Create php file with the function to access the database and get the require data. Figure 5.6 shows the “login.php” code which manage the login from android application.

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

 $user_name=@$_POST["login_name"];
 $user_pass=@$_POST["login_pass"];
 //$user_name='admin';
 //$user_pass='admin';

 $sql_query="SELECT `Staff_ID`, `Staff_Name`, `Staff_Post`, `Route_ID`
 FROM `Staff` WHERE `Staff_Name` ='$user_name' and `Pwd` = '$user_pass';";

 $result=mysqli_query($con,$sql_query);

if(mysqli_num_rows($result)>0)
{
    $row=mysqli_fetch_assoc($result);

    $id=$row["Staff_ID"];
    $name=$row["Staff_Name"];
    $post=$row["Staff_Post"];
    $routeid=$row["Route_ID"];

    //echo"Login Success...";
    echo"$id-$name-$post-$routeid ";
}
else
{
    echo "Login Failed ... Try Again" ;
}
 mysqli_close($con);
?>
```


Figure 5.6 login.php coding

5.3.2 Version Control Procedure

Version control procedure describes the procedure to control UTeM Bus Tracking using Google Map source code management. Version control procedure involves several steps. The initial development of the android application is performed without version control involved. After the initial development is complete, the application is put under the version control procedure. Developer and users' feedback regarding the functionality of the system will be acquired and from time to time. Table 5.2 shows the proposed version of UTeM Bus Tracking using Google Map.

Table 5.2 Proposed Version Control Procedure for UTeM Bus Tracking using Google Map

Version	Description
BusTrac v1	Initial version, the application allows the user to login, logout, sent location into database. Beside, admin also can add, update and delete the staff data in the Wamp Server (localhost).
BusTrac v2	Second version. The Google Map is used. Besides, user can choose the route and bus stop to estimate the bus arrival time. When user chooses the route option, the Google Map will show the respectively route and available bus stop marker. However, only able to estimate the bus arrival time for one bus.
BusTrac v3	The alarm service and bus schedule is added into the application.
BusTrac v4	Full version. Most of the bug in the application is solved and web hosting is used instead of Wamp Server to allow the application to work in wide area network.

5.4 Implementation Status

Implementation status refers to status of development for each module or functionality. Table 5.3 shows the implementation status for each module.

Table 5.3 Implementation Status

No.	Module/ Functionality	Description	Duration to complete
1	Interface Layout	Develop the interfaces for the android application	One week
2	Database Design	Create the database and tables. Relate the tables using relationship	One week
3	Administrative Module	Add function to the admin interface which allows the admin to add, update and delete staff.	Four days
4	Driver Module	Add function into the driver interface. Allowing the user to send location to the database	Two days
5	Google Map Module	Add Google Map into the tracking interface. Polyline is used to draw the route.	One week
6	Track Module	Add track function into the tracking interface.	One week
7	Alarm Module	Add alarm service into the tracking interface	One week

5.5 Conclusion

This chapter explain the steps to implement this project which are hardware configuration, software configuration, environment setup, version control procedure and implementation status. These steps are important to prevent unexpected event, which harm the project implementation. Moreover, the developers able to keep track and manage the changes in the source code through the software configuration management.

Chapter 6

TESTING

6.1 Introduction

This chapter explain the testing carry out for UTeM Bus Tracking using Google Map. The testing is conducted to ensure that the developed system achieve the user requirement and expectation.

6.2 Test Plan

Test plan is a detailed document that describes the overall testing approach used in the system. It is used to test the expected result of the technical tests records to be performed. Typically, test plan state the type of test, what test data will be used, what results are to be expected and the actual result of the test. Besides, it also defines the objectives and scope of the testing phases and identifies the methodology that will be used to conduct the testing.

6.2.1 Test Organization

Test organization is the personnel who are involved in the testing of the functionality of the android application. The personnel involve are developer, and user. Table 6.1 shows the responsibility of personnel in the testing.

Table 6.1 personnel responsibility

No.	Name	Personnel role	Responsibility
1	Chen Jian Tat	Developer	Responsible for overall testing. Testing involve include unit testing and integrate test. This is to ensure the application can be used and fulfil the user expectation.
2.	Siah Bing Sheng Gabriel Phan Kai Jie Tan Chia Yi Chin Jia Jun Hui Yoke Ling Raj Kumar Azza Hamza Pamendeep Kaur Sidhu Lim Lean Yee Tham Lee Yan	User	Executing the testing on the application and give the review after testing the application

6.2.2 Test Environment

The test environment is the place where the testing is conducted. Since the application is developed for the UTeM bus tracking, the test location is conducted at the route across UTeM bus route to the UTeM hostel. Besides, the hardware required is android phone. This android smart phone need to be has at least Android 4.4.4 to install and run the application.

6.2.3 Test Schedule

Test schedule need to be planned carefully. This is to ensure the testing can be conducted in timely manner without disrupting the users' time. Table 6.2 shows the test schedule.

Table 6.2 Test schedule

No.	Testing Type	Description	Date
1	Unit Testing	Testing conducted by developer to ensure that each module in the system is worked as defined.	8/7/2017
2	Integration Testing	Testing conducted by developer to ensure that each module in the system can work together.	22/7/2017

3	User Acceptance Test	Testing conducted by users to know the user satisfaction toward the application	5/8/2017

6.3 Test Strategy

This section describes about the test strategy used for this project. There are black box testing and white box testing. Black box testing is chosen because the user does not need to know the source code of the android application. Therefore, the tester does not need any IT knowledge to test this application. The black box testing technique chosen is the equivalence partitioning. The tester will be given a list of valid and invalid partition to test the application. This can reduce the time to test, as only one test need to be done in both valid partition and invalid partition.

6.3.1 Classes of tests.

The class of tests are used to define the class of the testing conducted for this project. There are three classes of tests used in this project which are unit testing, integration testing and user acceptance testing. Table 6.3 shows the classes of tests.

Table 6.3 Classes of tests

No.	Class of test	Description
1	Unit test	To ensure that the program of each module is work able to meet the user requirement.
2	Integration test	To ensure that each module in the system able to operate together when the all module is combined as a full system.
3	User Acceptance Test	To ensure that the end user is able to gain satisfaction when using the system. The sample of questionnaire is attached in Appendix B.

6.4 Test Design

Test design is about writing the test suites which specify the test conditions for a test case. In this project, the test design consists of test description and test data. Both elements are required to create test case for testing purpose. Figure 6.1 to Figure 6.10 show the interfaces in the application and the detail description for that particular interface. There is Unit Test and Integrated Test created for the Test Design.

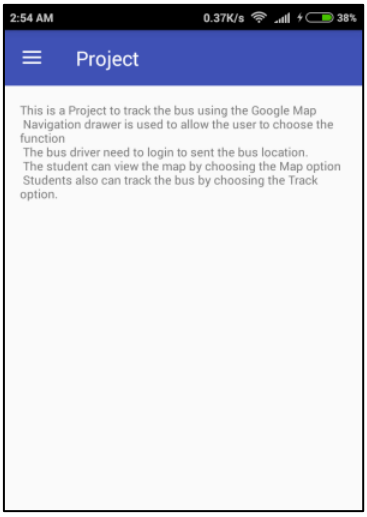


Figure 6.1MainPage Interface

	Function:
Menu Button: 	Move to navigation drawer interface (Figure 6.2)

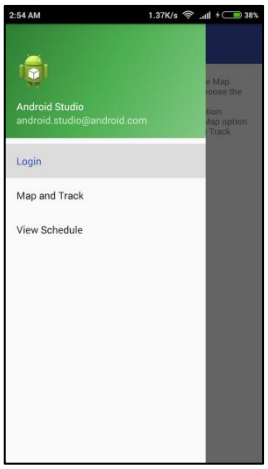


Figure 6.2 Navigation Drawer Interface

	Function:
Login option: Login	Move to Login Form interface (Figure 6.3)
Map and Track option: Map and Track	Move to Map and Track interface (Figure 6.9)
View Schedule option: View Schedule	Move to View Schedule interface (Figure 6.10)

Figure 6.3 Login Form Interface

	Function:
Username text field: Username	Text field to insert the username for login purpose.
Password text field: Password	Text field to insert the password for login purpose
Login button: Login	Validate the username and password with the server. If the username and password match with the server, move to the Admin Action interface (Figure 6.4) for Admin position, or move to Driver Action interface (Figure 6.8) for driver position.

Figure 6.4 Admin Action Interface

	Function:
ADD button: 	Move to Admin Add Interface (Figure 6.5)
UPDATE button: 	Move to Admin Update Interface (Figure 6.6)
Delete button: 	Move to Admin Delete Interface (Figure 6.7)
LOG OUT button: 	End the user login session and move to MainPage interface (Figure 6.1)

Figure 6.5 Admin Add Interface

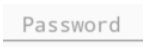
Function:	
Staff ID text field: 	Text field to insert the Staff ID, character and number can be inserted.
Name text field: 	Text field to insert the Name, only character can be inserted.
Password text field: 	Text field to insert Password, character and number can be inserted. .
Position drop down menu:	Drop down menu with “Admin” option and “Driver” option to be selected as staff position.
Route id drop down menu:	Drop down menu with “EP_BR_KT_KI”, “SU_KT_KI”, “SU_KB”, “MK_MT_KI”, and “MK_KB” options to be selected for route id.
ADD button:	When click, add all the information in the text fields and drop down menu into the server.
BACK button:	Move to Admin Action Interface (Figure 6.4)

Figure 6.6 Admin Update Interface

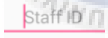
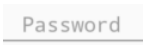
Function:	
Staff ID drop down menu: 	Drop down menu consist the staff information in the server. If any Staff ID item is selected, all the other fields and drop down menus will change based on the information in the server.
Name text field: 	Text field to hold the Name, only character can be inserted.
Password text field: 	Text field to hold Password, character and number can be inserted. .
Position drop down menu:	Drop down menu with "Admin" option and "Driver" option to be selected as staff position for update purpose.
Route id drop down menu:	Drop down menu with "EP_BR_KT_KI", "SU_KT_KI", "SU_KB", "MK_MT_KI", and "MK_KB" options to be selected for update route id.
UPDATE button: 	When click, update all the information in the text fields and drop down menu into the server for the selected staff id.
BACK button:	Move to Admin Action Interface (Figure 6.4)

Figure 6.7 Admin Delete Interface

Function	
Staff ID drop down menu:	Drop down menu consist the staff information in the server. If any Staff ID item is selected, all the other fields will change based on the information in the server to display the selected staff information.
DELETE button:	When click, the selected staff id with the corresponding staff information will be deleted in the server.
BACK button:	Move to Admin Action Interface (Figure 6.4)

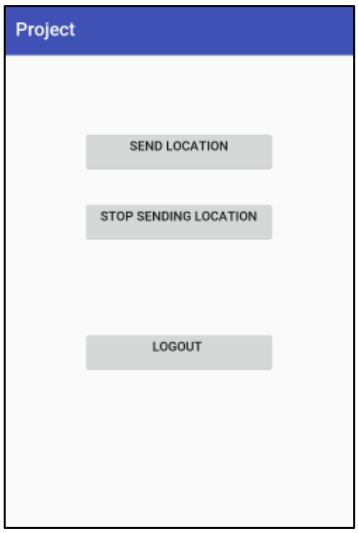
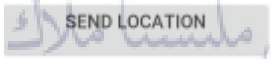


Figure 6.8 Driver Action Interface

Function	
SEND LOCATION button: 	When click, the system will get the GPS location and send it to server
STOP SENDING LOCATION button: 	When click after clicked the "SEND LOCATION", will stop sending the GPS location to the server.
LOGOUT button: 	When click, move to MainPage Interface (Figure 6.1)

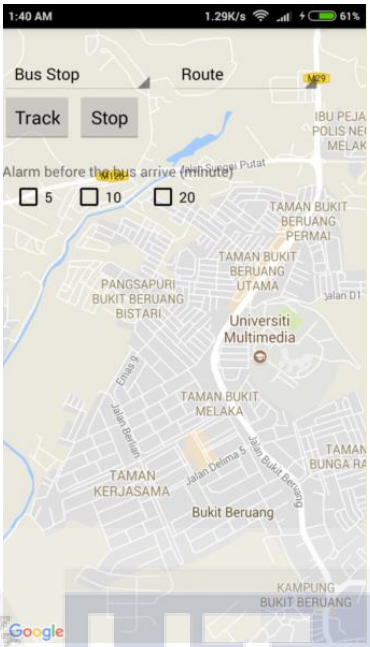
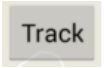
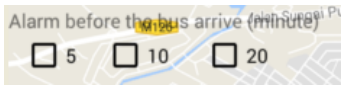


Figure 6.9 Map and Track Interface

Function	
BusStop drop down menu: 	Drop down menu consist the list of the bus stop. The items in the drop down menu are “KI”, ”KT”, “KB”, “EP”, “BR”, “SU”, “MK”. These items are important to determine student wait at which bus stop.
Bus Route drop down menu: 	Drop down menu consist the list of the bus route. The items in the drop down menu are “EP_BR_KT_KI”, ”SU_KT_KI”, “SU_KB”, “MK_KT_KI”, and “MK_KB”. When item is selected, the corresponding bus route and the bus stop icon is show in the Google Map.
Track button: 	When clicked, the GPS location for the running bus is obtained. Next, the GPS location of the bus and bus stop is used to calculate the bus arrival time and show at the top bus stop icon.

Stop button: 	When click after clicked the Track button, the system will stop requesting bus GPS location and no calculation in bus arrival time. Finally, system clears all the bus route and bus stop icon in the Google Map.
Alarm check boxes: 	The application need to track the bus before check any alarm check box. When Checkbox 5 is checked, the alarm will ring when the bus arrival time is less than 5 minutes. When Checkbox 10 is checked, the alarm will ring when the bus arrival time is less than 10 minutes. When Checkbox 20 is checked, the alarm will ring when the bus arrival time is less than 20 minutes.

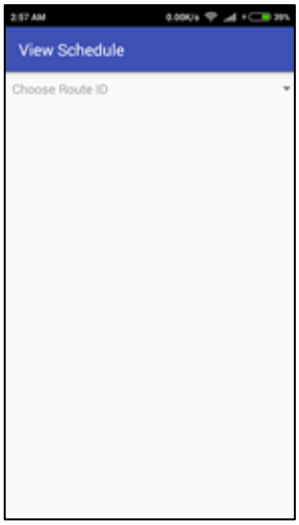


Figure 6.10 View Schedule Interface

Function	
Choose Route ID drop down menu: 	This drop down menu consists "EP_BR_KT_KI","SU_KT_KI", "SU_KB", "MK_KT_KI", and "MK_KB". When any item is selected, the bus schedule for the chosen route id will be shown.

6.4.1 Unit Test

User Login Test

Table 6.4 shows the test design to test the system when the user login using invalid username and password combination.

Table 6.4 Unit Test 1.1

Test Case ID	BusTrac_UTP_1.1
Test Name	UTP_user_inv_login
Test Description	To test the system when the user login using invalid username and password combination.
Test Requirement	To validate admin can login into the system.
Pre-Condition	User opens the login form page.
Input/Test Data	*Remark : the data is in [“username”, “password”] Valid combination : [“admin”,”admin”], [“chen”,”chen”], [“driver”,driver”] Invalid combination: [“”,””], [“admin”,””], [“”,”admin”], etc
Flow/Steps	1. Choose the login option in the main interface. 2. The system shows the login interface. 3. Insert the invalid combination test data into the field. 4. Press the login button.

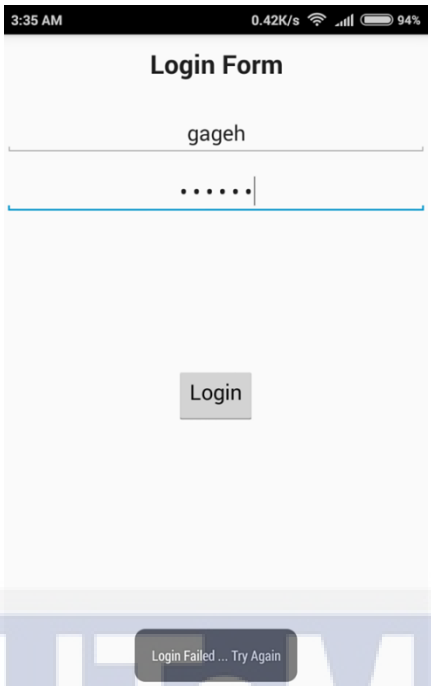
Expected Result	The system will popup message “Login Fail … Try Again”. Figure 6.11 shows the expected result.  **Figure 6.11 Login Fail**
Cycle 1	Pass
Cycle 2	Pass

Table 6.5 shows the test design to test the system when the user login as admin using valid username and password combination.

Table 6.5 Unit Test 1.2

Test Case ID	BusTrac_UTP_1.2
Test Name	UTP_admin_v_login
Test Description	To test the system when the user login as admin using valid username and password combination.
Test Requirement	To validate admin user can login into the system.
Pre-Condition	User opens the login form page.
Input/Test Data	*Remark : the data is in [“username”, “password”] Valid combination : [“admin”,”admin”], [“chen”,”chen”] Invalid combination: [“”,””], [“admin”,””], [“”,”admin”], etc
Flow/Steps	1. Choose the login option in the main interface. 2. The system shows the login interface. 3. Insert the valid combination test data into the field. 4. Press the login button.
Expected Result	The system will popup message “Login success …” and move to admin action page. Figure 6.12 shows the expected result.

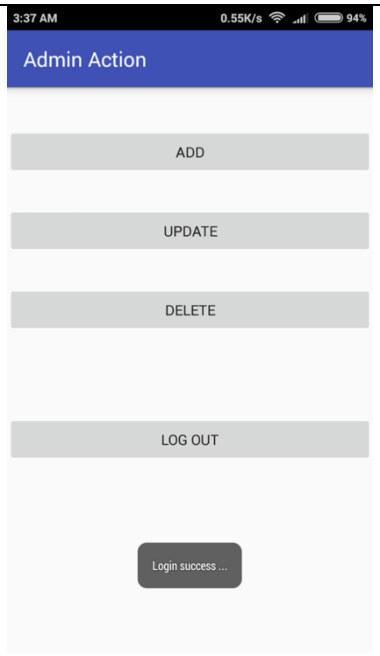
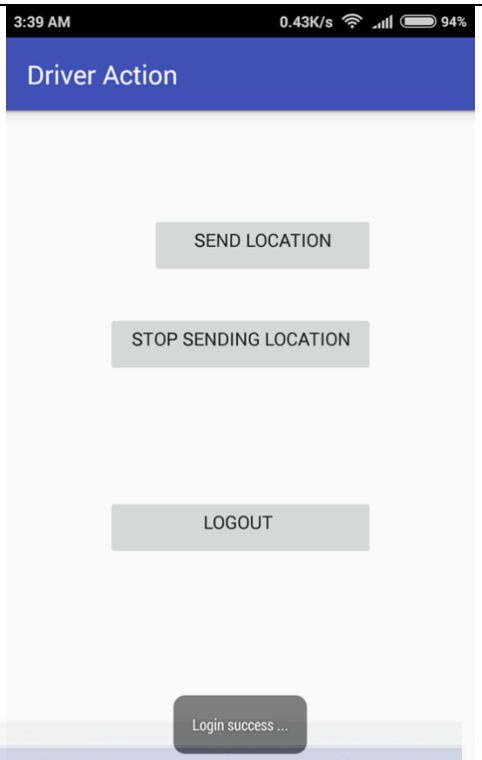
	 **Figure 6.12 Admin Login Success**	
Cycle 1	Pass	
Cycle 2	Pass	

Table 6.6 shows the test design to test the system when the user login as driver using valid username and password combination.

Table 6.6 Unit Test 1.3

Test Case ID	BusTrac_UTP_1.3
Test Name	UTP_driver_v_login
Test Description	To test the system when the user login as driver using valid username and password combination.
Test Requirement	To validate user can login into the system.
Pre-Condition	User opens the login form page.
Input/Test Data	*Remark : the data is in [“username”, “password”] Valid combination : [“driver”,driver”] Invalid combination: [“”,””], [“driver”,””], [“”,”driver”], etc
Flow/Steps	1. Choose the login option in the main interface. 2. The system shows the login interface. 3. Insert the valid combination test data into the field. 4. Press the login button.
Expected Result	The system will popup message “Login success …” and move to driver action page. Figure 6.13 shows the expected result.

	 **Figure 6.13 Driver Login Success**
Cycle 1	Pass
Cycle 2	Pass

Admin Action Test

Table 6.7 shows the test design to test the system when the user use invalid staff information combination when the add button is pressed.

Table 6.7 Unit Test 2.1

Test Case ID	BusTrac_UTP_2.1
Test Name	UTP_admin_inv_addStaff_1
Test Description	To test the system when the user use invalid staff information combination when the add button is pressed.
Test Requirement	To validate user can add staff into the system.
Pre-Condition	User must be in Admin Add Interface.
Input/Test Data	*Remark : The data is in order of [“Staff ID” , “Name” , ”Password” , ”Position” , “Route ID” , ”Tel No.”] and data type: character = Ch, number = No Valid combination : [“Ch+No”,Ch”,”Ch+No” ,”Ch” , “Ch” , “No”] Invalid combination: [“Ch”,Ch+No”,”No” ,”Ch” , “Ch” , “No”], [“Ch+No”,Ch”,”No” ,”No” , “No” , “No”], etc
Flow/Steps	1. Fill in the invalid combination. 2. Press ADD button
Expected Result	The system will popup message “Please insert all correct information to proceed”. Figure 6.14 shows the expected result.

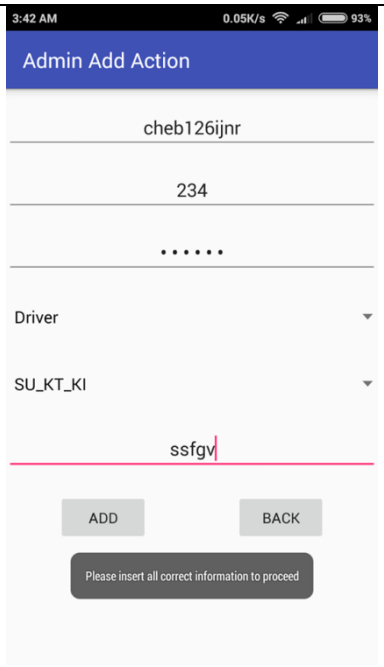
	 **Figure 6.14 Admin Add Failure 1**
Cycle 1	Pass
Cycle 2	Pass

Table 6.8 shows the test design to test the system when the user use invalid or existing staff id with valid combination when the add button is pressed.

Table 6.8 Unit Testing 2.2

Test Case ID	BusTrac_UTP_2.2
Test Name	UTP_admin_inv_addStaff_2
Test Description	To test the system when the user use invalid or existing staff id with valid combination when the add button is pressed.
Test Requirement	To validate user can add staff into the system.
Pre-Condition	User must be I Admin Add Interface.
Input/Test Data	*Remark : The data is in order of [“Staff ID” , “Name” , ”Password” , ”Position” , “Route ID” , ”Tel No.”] and data type: character = Ch, number = No Invalid staff id : “admin”, “chen” Valid combination : [“Ch+No”,Ch”,”Ch+No” ,”Ch” , “Ch” , “No”] Invalid combination: [“Ch”,Ch+No”,”No” ,”Ch” , “Ch” , “No”], [“Ch+No”,Ch”,”No” ,”No” , “No” , “No”], etc
Flow/Steps	1. Fill in the invalid staff id and valid combination. 2. Press ADD button
Expected Result	The system will popup message “Failed to insert data”. Figure 6.15 shows the expected result.

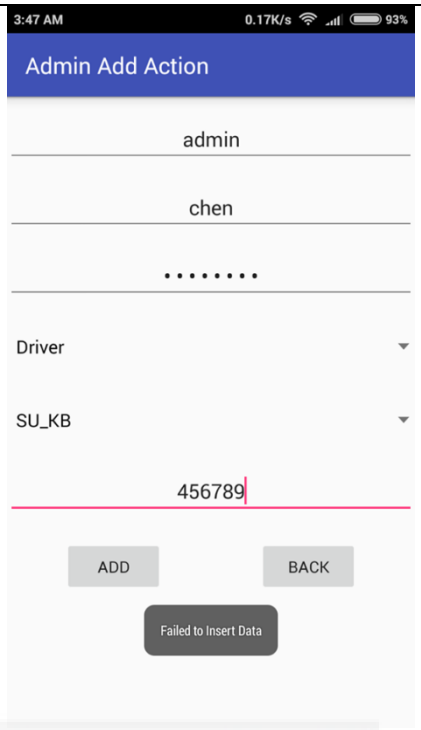
	 **Figure 6.15 Admin Add Failure 2**	
Cycle 1	Pass	
Cycle 2	Pass	

Table 6.9 shows the test design to test the system when the user use valid or non-existing staff id with valid combination when the add button is pressed.

Table 6.9 Unit Testing 2.3

Test Case ID	BusTrac_UTP_2.3
Test Name	UTP_admin_v_addStaff
Test Description	To test the system when the user use valid or non-existing staff id with valid combination when the add button is pressed.
Test Requirement	To validate user can add staff into the system.
Pre-Condition	User must be in Admin Add Interface
Input/Test Data	*Remark : The data is in order of [“Staff ID” , “Name” , ”Password” , ”Position” , “Route ID” , ”Tel No.”] and data type: character = Ch, number = No Valid staff id : Other except “admin”and “chen” Invalid staff id : “admin”, “chen” Valid combination : [“Ch+No”,Ch”,”Ch+No” ,”Ch” , “Ch” , “No”] Invalid combination: [“Ch”,Ch+No”,”No” ,”Ch” , “Ch” , “No”], [“Ch+No”,Ch”,”No” ,”No” , “No” , “No”], etc
Flow/Steps	1. Fill in the valid staff id and valid combination. 2. Press ADD button
Expected Result	The system wills popup message “Data insertion success”. Figure 6.16 shows the expected result.

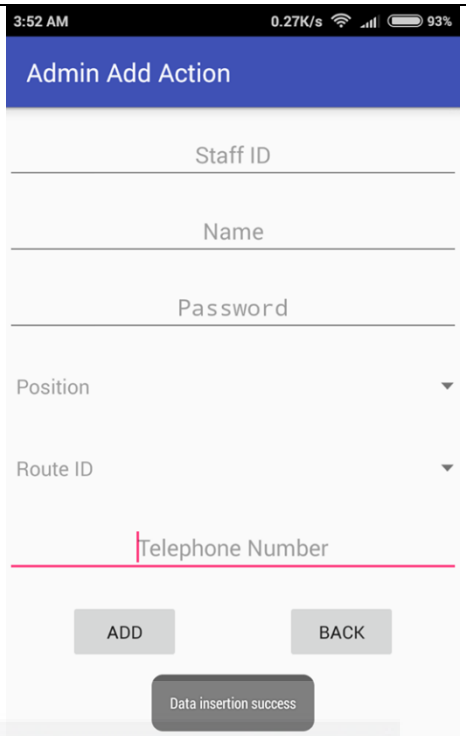
	 **Figure 6.16 Admin Add Success**
Cycle 1	Pass
Cycle 2	Pass

Table 6.10 shows the test design to test the system when the user update the staff information in the database using invalid combination when the add button is pressed.

Table 6.10 Unit Testing 2.4

Test Case ID	BusTrac_UTP_2.4
Test Name	UTP_admin_inv_updateStaff
Test Description	To test the system when the user update the staff information in the database using invalid combination when the add button is pressed.
Test Requirement	To validate user can update staff into the system.
Pre-Condition	User must be in Admin Update Interface.
Input/Test Data	*Remark : The data is in [Name" , "Password" , "Position" , "Route ID" , "Tel No."] and data type: character = Ch, number = No Valid combination : [Ch","Ch+No" ,"Ch" , "Ch" , "No"] Invalid combination: ["Ch+No","No" ,"Ch" , "Ch" , "No"], ["Ch+No",Ch","No" ,"No" , "No"], etc
Flow/Steps	1. Choose staff id which need to be updated. 2. Change the staff information into invalid combination. 3. Press UPDATE button.
Expected Result	The system will popup message "Please insert all correct information to update". Figure 6.17 shows the expected result.

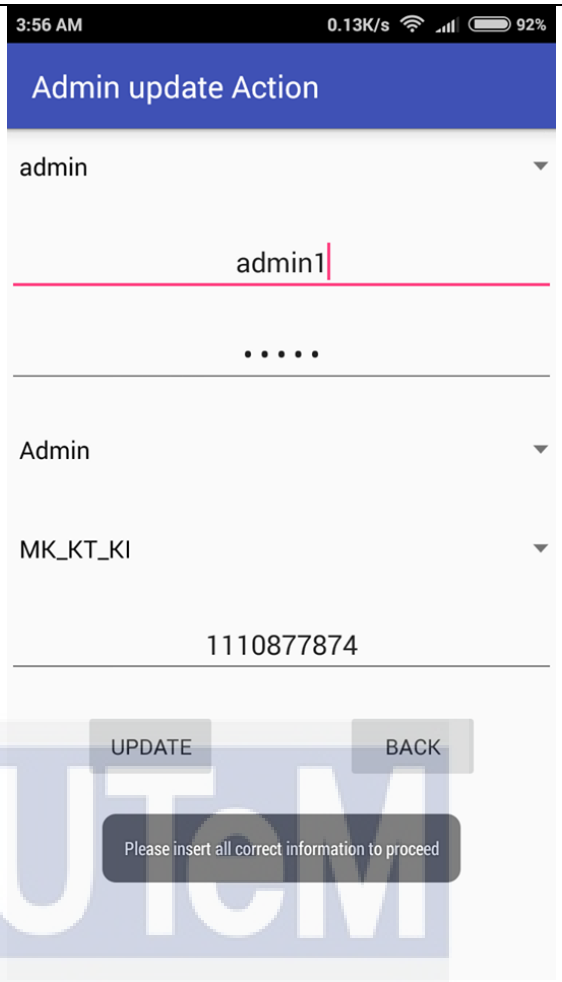
	 **Figure 6.17 Admin Update Failure**
Cycle 1	Pass
Cycle 2	Pass

Table 6.11 shows the test design to test the system when the user update the staff information in the database using valid combination when the add button is pressed.

Table 6.11 Unit Testing 2.5

Test Case ID	BusTrac_UTP_2.5
Test Name	UTP_admin_v_updateStaff
Test Description	To test the system when the user update the staff information in the database using valid combination when the add button is pressed.
Test Requirement	To validate user can update staff into the system.
Pre-Condition	User must be in Admin Update Interface
Input/Test Data	*Remark : The data is in [Name" , "Password" , "Position" , "Route ID" , "Tel No."] and data type: character = Ch, number = No Valid combination : [Ch","Ch+No" ,"Ch" , "Ch" , "No"] Invalid combination: ["Ch+No","No" ,"Ch" , "Ch" , "No"], ["Ch+No",Ch","No" ,"No" , "No"], etc
Flow/Steps	1. Choose the staff Id which need to be updated.. 2. Change the staff information into valid combination. 3. Press UPDATE button.
Expected Result	The system will popup message "Successfully update data". Figure 6.18 shows the expected result.

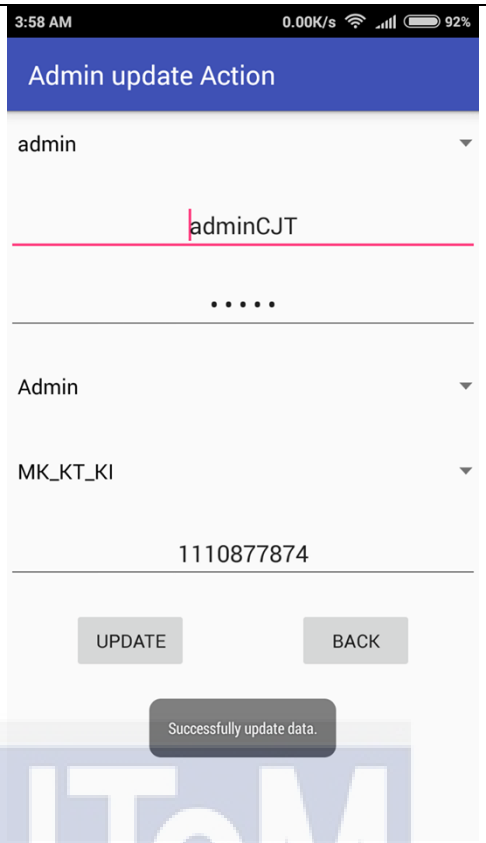
	 **Figure 6.18 Admin Update Success**
Cycle 1	Pass
Cycle 2	Pass

Table 6.12 shows the test design to test the system when the user can delete the staff information in the database when the delete button is pressed.

Table 6.12 Unit Testing 2.6

Test Case ID	BusTrac_UTP_2.6
Test Name	UTP_admin_v_deleteStaff
Test Description	To test the system when the user can delete the staff information in the database when the delete button is pressed.
Test Requirement	To validate user can delete staff into the system.
Pre-Condition	User must be in Admin Delete Interface
Input/Test Data	Valid Option: the Staff id show in drop down list. Invalid Option: Fields except staff id.
Flow/Steps	1. Choose the staff id that needs to be deleted. 2. Press DELETE button.
Expected Result	The system will popup message “Delete Data Successfully”. Figure 6.19 shows the expected result.

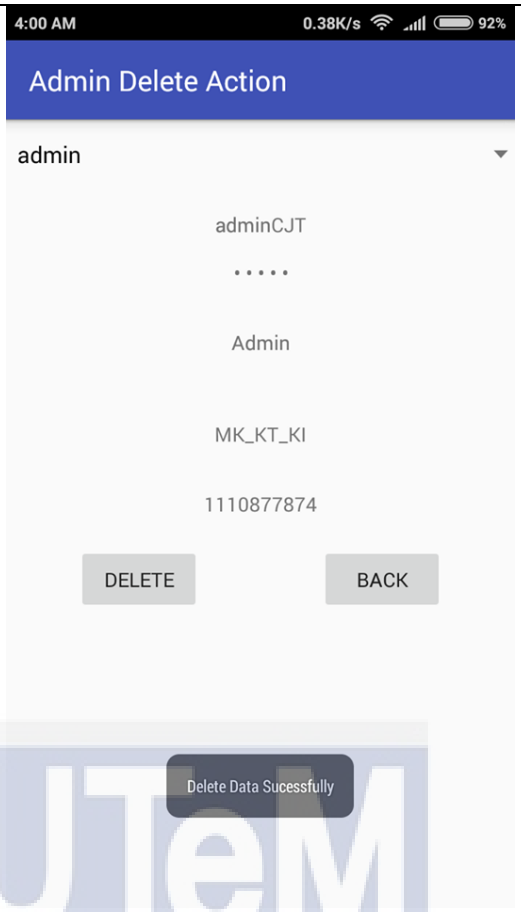
<table>
<tr><td></td><td> 

Figure 6.19 Admin Delete Success</td></tr>
<tr><td>Cycle 1</td><td>Pass</td></tr>
<tr><td>Cycle 2</td><td>Pass</td></tr>
</table>

Driver Action Test

Table 6.13 shows the test design to test the system when users want to send the location to database

Table 6.13 Unit Testing 3.1

Test Case ID	BusTrac_UTP_3.1
Test Name	UTP_sendLlocation
Test Description	To test the system when user wants to send the location to database
Test Requirement	To validate user can send the data into the system.
Pre-Condition	User opens the login page and login as driver.
Input/Test Data	Valid Option: SEND LOCATION Invalid Option: STOP SENDING LOCATION
Flow/Steps	1. Click the SEND LOCATION option
Expected Result	The system will popup message “Start to send location to server” and the data is insert into the database server. Figure 6.20 shows the expected result.

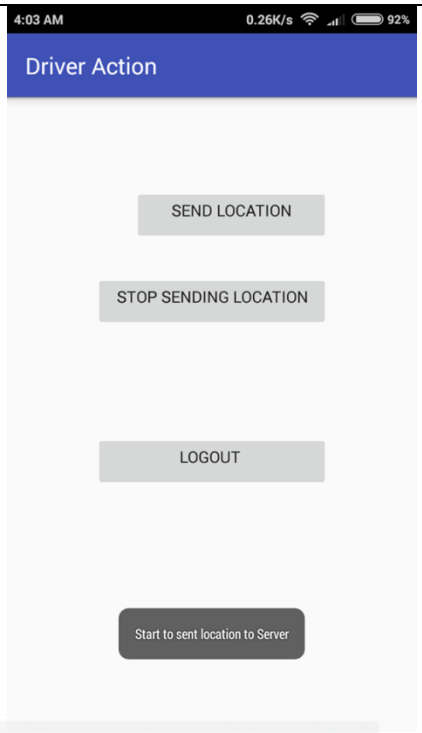
	 **Figure 6.20 Driver Send Location Success**
Cycle 1	Pass
Cycle 2	Pass

Table 6.14 shows the test design to test the system when user wants to stop sending the location to server.

Table 6.14 Unit Testing 3.2

Test Case ID	BusTrac_UTP_3.2
Test Name	UTP_stopLlocation
Test Description	To test the system when user want to stop sending the location to server
Test Requirement	To validate user can stop sending the data into the system.
Pre-Condition	User opens the login page and login as driver and start sending location to the server.
Input/Test Data	Valid Option: STOP SENDING LOCATION Invalid Option: SEND LOCATION
Flow/Steps	1. click the valid option button
Expected Result	The system will popup message “Stopping the sending” and the data is stop insert into the database server. Figure 6.21 shows the expected result.

	 **Figure 6.21 Driver Stop Sending Location**
Cycle 1	Pass
Cycle 2	Pass

Tracking test

Table 6.15 shows the test design to test the system when user wants to track the bus.

Table 6.15 Unit Testing 4.1

Test Case ID	BusTrac_UTP_4.1
Test Name	UTP_inv_busTrack_1
Test Description	To test the system when user wants to track the bus.
Test Requirement	To validate user can track the bus.
Pre-Condition	User opens the Map and Track page.
Input/Test Data	Valid Option: Select the any item from both "Bus Stop" and "Route" drop down list. Invalid Option: Does not select any item from "Bus Stop" nor "Route" drop down list, etc.
Flow/Steps	1. Choose the invalid option 2. Press Track button
Expected Result	The system will popup message "Start tracking" and then popup message "Please choose desire bus stop". Figure 6.22 shows the expected result.

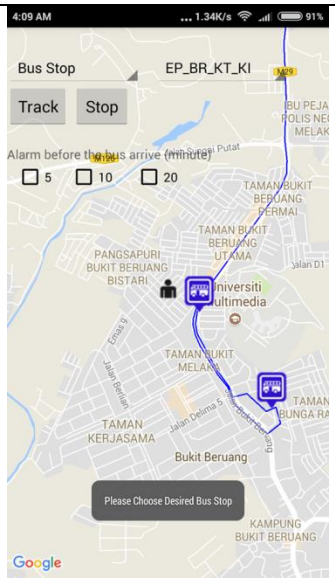
	 **Figure 6.22 Student Track Bus Failure 1**
Cycle 1	Pass
Cycle 2	Pass

Table 6.16 shows the test design to test the system when user wants to track the bus.

Table 6.16 Unit Testing 4.2

Test Case ID	BusTrac_UTP_4.2
Test Name	UTP_inv_busTrack_2
Test Description	To test the system when user wants to track the bus.
Test Requirement	To validate user can track the bus.
Pre-Condition	User opens the Map and Track page.
Input/Test Data	Valid Option: Select the item from both "Route" with corresponding "Bus Stop" Invalid Option: Does not select the item from both "Route" with corresponding "Bus Stop", etc.
Flow/Steps	1. Choose the invalid option 2. Press Track button
Expected Result	The system will popup message "Start tracking" and then popup message "Please select correct bus stop and route". Figure 6.23 shows the expected result.

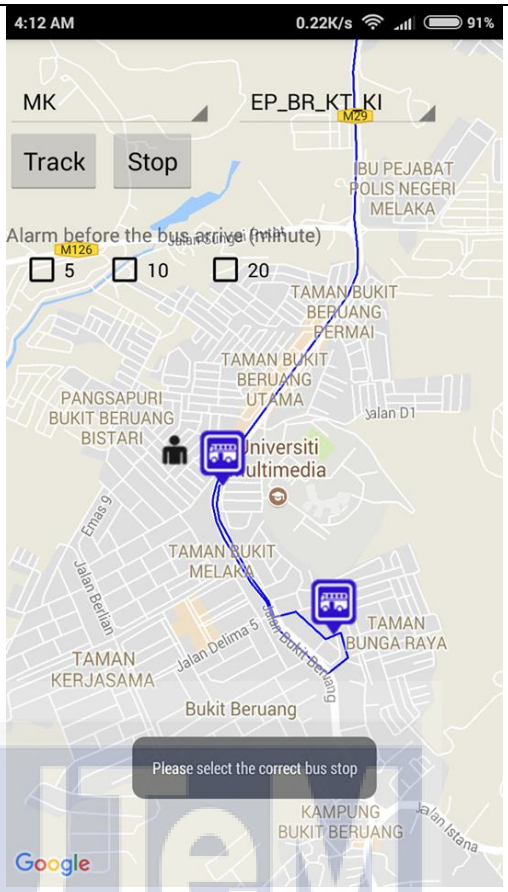
	 **Figure 6.23 Student Track Bus Failure 2**
Cycle 1	Fail
Cycle 2	Pass

Table 6.17 shows the test design to test the system when user wants to track the bus.

Table 6.17 Unit Testing 4.3

Test Case ID	BusTrac_UTP_4.3
Test Name	UTP_v_busTrack
Test Description	To test the system when user wants to track the bus.
Test Requirement	To validate user can track the bus.
Pre-Condition	User opens the Map and Track page.
Input/Test Data	Valid Option: Select the any item from both "Bus Stop" and "Route" which is relevant from the drop down lists Invalid Option: Does not select any item from "Bus Stop" nor "Route" drop down lists, etc
Flow/Steps	1. Choose the valid option 2. Press Track button
Expected Result	The system will popup message "Start tracking" and then redirect the Google Map to the bus stop location and bus arrival time is appear in info window. Figure 6.24 shows the expected result.

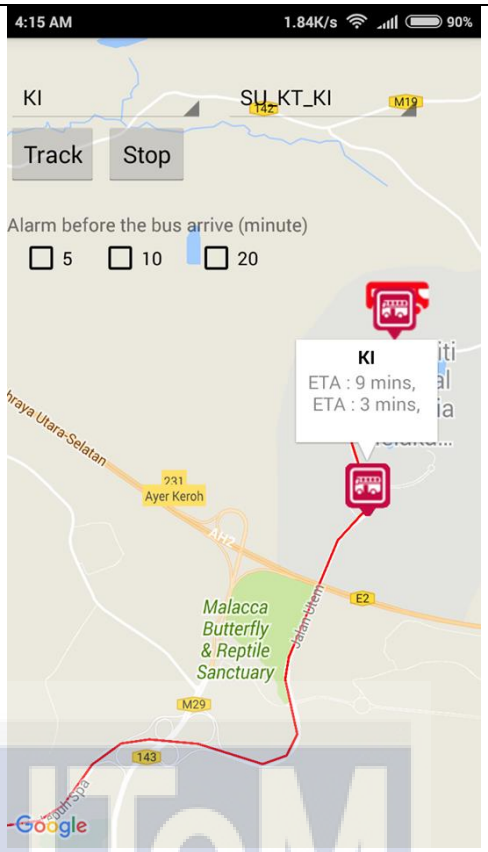
	 **Figure 6.24 Student Track Bus Success**
Cycle 1	Fail
Cycle 2	Pass

Table 6.18 shows the test design to test the system when user wants to stop tracking the bus.

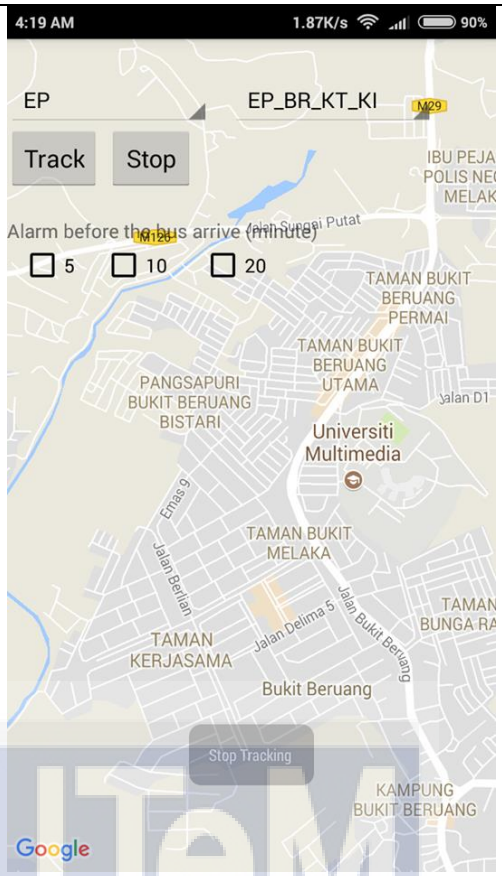
Table 6.18 Unit Testing 4.4

Test Case ID	BusTrac_UTP_4.4
Test Name	UTP_inv_stopTrack
Test Description	To test the system when user want to stop tracking the bus.
Test Requirement	To validate user can stop tracking the bus.
Pre-Condition	User opens the Map and Track page.
Input/Test Data	Valid Option: Start tracking the bus first, then press STOP button. Invalid Option: Direct press STOP button
Flow/Steps	1. Choose the invalid option
Expected Result	Nothing happen in the Google Map
Cycle 1	Fail
Cycle 2	Pass

Table 6.19 shows the test design to test the system when user wants to stop tracking the bus.

Table 6.19 Unit Testing 4.5

Test Case ID	BusTrac_UTP_4.5
Test Name	UTP_v_stopTrack
Test Description	To test the system when user want to stop tracking the bus.
Test Requirement	To validate user can stop tracking the bus.
Pre-Condition	User opens the Map and Track page and start tracking the bus.
Input/Test Data	Valid Option: Start tracking the bus first, then press STOP button. Invalid Option: Direct press STOP button
Flow/Steps	1. Start tracking the bus. 2. Press STOP button
Expected Result	The system will popup message “Stop Tracking” and the bus markers in the map are clear. Figure 6.25 shows the expected result.

	 **Figure 6.25 Student Stop Track Bus Success**
Cycle 1	Pass
Cycle 2	Pass

Alarm test

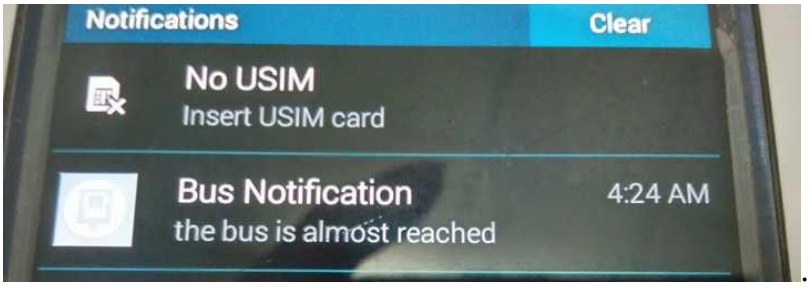
Table 6.20 shows the test design to test the system when user wants to set alarm for the bus tracking.

Table 6.20 Unit Testing 5.1

Test Case ID	BusTrac_UTP_5.1
Test Name	UTP_inv_alarm
Test Description	To test the system when user wants to set alarm for the bus tracking.
Test Requirement	To validate user can set alarm for the bus tracking.
Pre-Condition	User opens the Map and Track page.
Input/Test Data	Valid Option: Start tracking the bus first, then press any Alarm check box. Invalid Option: Direct press any Alarm check box
Flow/Steps	1. Choose the invalid option
Expected Result	Nothing happen in the Google Map
Cycle 1	Fail
Cycle 2	Pass

Table 6.21 shows the test design to test the system when user wants to set alarm for the bus tracking.

Table 6.21 Unit Testing 5.2

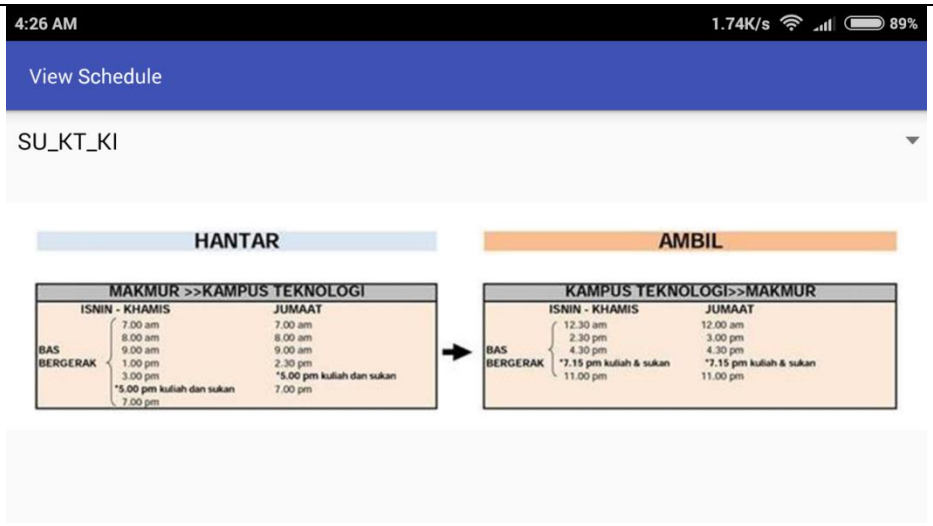
Test Case ID	BusTrac_UTP_5.2
Test Name	UTP_v_alarm
Test Description	To test the system when user wants to set alarm for the bus tracking.
Test Requirement	To validate user can set alarm for the bus tracking.
Pre-Condition	User opens the Map and Track page and start tracking the bus.
Input/Test Data	Valid Option: Start tracking the bus first, and then press any Alarm check box. Invalid Option: Direct press any Alarm check box
Flow/Steps	1. Choose the valid option
Expected Result	Alarm is triggered when the time chosen is less than the bus arrival time and notification is created. Figure 6.26 shows the expected result.  . **Figure 6.26 Bus Alarm Notification and Alarm**
Cycle 1	Fail
Cycle 2	Pass

View Schedule

Table 6.22 shows the test design to test the system when user can view the bus schedule.

Table 6.22 Unit Testing 6.1

Test Case ID	BusTrac_UTP_6.1
Test Name	UTP_v_schedule
Test Description	To test the system when user can view the bus schedule.
Test Requirement	To validate user can view the bus schedule.
Pre-Condition	User opens the View Schedule Page.
Input/Test Data	Valid Option: Choose any item from the route id drop down list. Invalid Option: -
Flow/Steps	1. Choose any item from the route id drop down list.
Expected Result	The corresponding bus schedule appears. Figure 6.27 shows the expected result.

	 **Figure 6.27 Student View Schedule**
Cycle 1	Pass
Cycle 2	Pass

6.4.2 Integrated Testing

Table 6.23 shows the test design to test the system to change from main menu to login page.

Table 6.23 Integrated Testing 1

Test Case ID	BusTrac_ITP_1
Test Name	ITP_MainMenu_Login
Test Description	To test the system to change from main menu to login page.
Test Requirement	To validate the function of login option
Pre-Condition	User is at the main menu.
Input/Test Data	-
Flow/Steps	1. Choose the Login option
Expected Result	The system change the interface from main menu to login page
Cycle 1	Pass
Cycle 2	Pass

Table 6.24 shows the test design to test the system to whether it can maintain the admin session after successfully login.

Table 6.24 Integrated Testing 2

Test Case ID	BusTrac_ITP_2
Test Name	ITP_MainMenu_LoginPage_AdminPage
Test Description	To test the system to whether it can maintain the admin session after successfully login.
Test Requirement	To validate the admin session in the system
Pre-Condition	User successfully login the system as admin.
Input/Test Data	-
Flow/Steps	1. User chooses the login option to change to login page. 2. User fills in the correct admin login information and press login button. 3. After the system validate the login information, the system move to admin main page and prompt "Login Success …" 4. Close the application. 5. Open the application again. 6. Choose the login option at the main page.
Expected Result	The system changes from the main page to admin main page by passing the login page.
Cycle 1	Fail
Cycle 2	Pass

Table 6.25 shows the test design to test the system to whether it can maintain the driver session after successfully login.

Table 6.25 Integrated Testing 3

Test Case ID	BusTrac_ITP_3
Test Name	ITP_MainMenu_LoginPage_DriverPage
Test Description	To test the system to whether it can maintain the driver session after successfully login
Test Requirement	To validate the driver session in the system
Pre-Condition	User successfully login the system as driver.
Input/Test Data	-
Flow/Steps	1. User chooses the login option to change to login page. 2. User fills in the correct driver login information and press login button. 3. After the system validate the login information, the system move to driver main page and prompt "Login Success …" 4. Close the application. 5. Open the application again. 6. Choose the login option at the main page.
Expected Result	The system changes from the main page to driver main page by passing the login page.
Cycle 1	Fail
Cycle 2	Pass

Table 6.26 shows the test design to test the system to whether it can change to Add Staff page.

Table 6.26 Integrated Testing 4

Test Case ID	BusTrac_ITP_4
Test Name	ITP_AdminPage_AdminAdd
Test Description	To test the system to whether it can change to Add Staff page.
Test Requirement	To validate the function of Add Staff button at Admin Main Page.
Pre-Condition	User successfully login the system as admin.
Input/Test Data	-
Flow/Steps	1. Selects the Add Staff Button
Expected Result	The system changes from the admin main page to add staff page.
Cycle 1	Pass
Cycle 2	Pass

Table 6.27 shows the test design to test the system to whether it can change to Update Staff page.

Table 6.27 Integrated Testing 5

Test Case ID	BusTrac_ITP_5
Test Name	ITP_AdminPage_AdminUpdate
Test Description	To test the system to whether it can change to Update Staff page.
Test Requirement	To validate the function of Update Staff button at Admin Main Page.
Pre-Condition	User successfully login the system as admin.
Input/Test Data	-
Flow/Steps	1. Selects the Update Staff Button
Expected Result	The system changes from the admin main page to update staff page.
Cycle 1	Pass
Cycle 2	Pass

Table 6.28 shows the test design to test the system to whether it can change to Delete Staff page.

Table 6.28 Integrated Testing 6

Test Case ID	BusTrac_ITP_6
Test Name	ITP_AdminPage_AdminDelete
Test Description	To test the system to whether it can change to Delete Staff page.
Test Requirement	To validate the function of Delete Staff button at Admin Main Page.
Pre-Condition	User successfully login the system as admin.
Input/Test Data	-
Flow/Steps	1. Selects the Delete Staff Button
Expected Result	The system changes from the admin main page to delete staff page.
Cycle 1	Pass
Cycle 2	Pass

Table 6.29 shows the test design to test the system to close the admin session after logout.

Table 6.29 Integrated Testing 7

Test Case ID	BusTrac_ITP_7
Test Name	ITP_AdminPage_MainPage
Test Description	To test the system to close the admin session after logout.
Test Requirement	To validate the function of Logout button at Admin Main Page.
Pre-Condition	User successfully login the system as admin.
Input/Test Data	-
Flow/Steps	1. Selects the Logout Button. 2. The system change from Admin Main Page to Main Menu Page. 3. Select Login option at the Main Menu Page.
Expected Result	The system changes from the admin main page to Main Menu page. After that, choose the login option will bring to login page rather than by passing the login page.
Cycle 1	Pass
Cycle 2	Pass

Table 6.30 shows the test design to test the system to whether close the driver session after logout.

Table 6.30 Integrated Testing 8

Test Case ID	BusTrac_ITP_8
Test Name	ITP_AdminPage_MainPage
Test Description	To test the system to whether close the driver session after logout.
Test Requirement	To validate the function of Logout button at Driver Main Page.
Pre-Condition	User successfully login the system as driver.
Input/Test Data	-
Flow/Steps	1. Selects the Logout Button. 2. The system change from Driver Main Page to Main Menu Page. 3. Select Login option at the Main Menu Page.
Expected Result	The system changes from the Driver Main Page to Main Menu page. After that, choose the login option will bring to login page rather than by passing the login page.
Cycle 1	Pass
Cycle 2	Pass

Table 6.31 shows the test design to test the system to change from main menu to track page.

Table 6.31 Integrated Testing 9

Test Case ID	BusTrac_ITP_9
Test Name	ITP_MainMenu_TrackPage
Test Description	To test the system to change from main menu to track page.
Test Requirement	To validate the function of Map and Track option in the Main Menu Page.
Pre-Condition	User is at the main menu.
Input/Test Data	-
Flow/Steps	1. Choose the Map and Track option
Expected Result	The system change the interface from main menu to track page
Cycle 1	Pass
Cycle 2	Pass

Table 6.32 shows the test design to test the system to change from main menu to View Schedule page.

Table 6.32 Integrated Testing 10

Test Case ID	BusTrac_ITP_10
Test Name	ITP_MainMenu_ViewSchedule
Test Description	To test the system to change from main menu to View Schedule page.
Test Requirement	To validate the function of Schedule option
Pre-Condition	User is at the main menu.
Input/Test Data	-
Flow/Steps	1. Choose the Schedule option
Expected Result	The system change the interface from main menu to View Schedule page
Cycle 1	Pass
Cycle 2	Pass

6.5 Test Result and Analysis

This section discuss about analyse the result obtain from the unit testing, integrated testing and user acceptance test. Following show the result and analysis for the testing.

a. Unit Testing

Unit Testing Cycle 1

There are 17 Unit Testing Case. However there are five testing fail to meet the expected result. The first fail test case is the application can track the bus with unrelated route and bus stop. This is due to the mistake from the developer as the developer forgot the logical error. Next fail test case is the application only able to calculate the bus arrival time for one bus. This is also due to the incapable of the developer to create an array to hold the bus arrival time. The third error is alarm setting in the bus track fail to ring even though the estimated time is passed. Moreover, there are two fail cases which are, press the alarm or press the stop tracking without start tracking the bus. These two cases can cause the application to crash as there are many exceptions do not handle with care. Table 6.4 shows the result of the Unit Testing Cycle 1

Table 6.33 Unit Testing Cycle 1

Unit Testing Cycle 1	
Pass (%)	Fail (%)
70.58	29.42

Unit Test Cycle 2:

The result obtain from Unit Testing Cycle 1 is used as reference. Correction has been made after the Unit Testing Cycle 1. Therefore there are no failed in Unit Testing Cycle 2.Table 6.5 shows the result for the unit testing cycle 2.

Table 6.34 Unit Testing Cycle 2

Unit Testing Cycle 2	
Pass (%)	Fail (%)
100	0

b. Integrated Testing

Integrated Testing Cycle 1

There are 10 test case had conducted and 2 of the case are failed to meet the expected outcome. Both cases are the admin session and driver session. The message are prompted even the user already login the system. Table 6.6 shows the result for the integrated testing cycle 2.

Table 6.35 Integrated Testing Cycle 1

Integrated Testing Cycle 1	
Pass (%)	Fail (%)
80	20

The result obtained from the Integrated Testing Cycle 1 is used. Some code adjustment had been made to solve the failed case from Integrated Testing Cycle 1. Table 6.7 shows the result for the integrated testing cycle 2.

Table 6.36 Integrated Testing Cycle 2.

Integrated Testing Cycle 2	
Pass (%)	Fail (%)
100	0

c. User Acceptance Testing

10 users are test as students. This is due to the main objective of this system is to help the student to track the bus and provide user satisfaction when using the system. Result shows that most of the users are satisfy with the system. Table 6.8 shows the user acceptance testing result.

Table 6.37 User Acceptance Testing

User Acceptance Testing					
Description	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
The application is intuitive and easy to use	0	1	3	5	1
The Google Map show the bus location correctly	0	0	4	4	2
The bus arrival time is accurate and easy to see	0	2	3	5	0
The alarm is useful to alert the user when the bus is almost reach	1	1	2	5	1
User can view the bus schedule easily and clearly	0	2	2	6	0
User is satisfy using this application to track the bus	0	2	5	3	0

6.6 Conclusion

In conclusion, the testing helps the developer to identify the issues or bug in the system. It also ensures the system is work properly. Therefore it is very important to carry out the testing before deploy the system into public.

Chapter 7

CONCLUSION

7.1 Introduction

This final chapter discuss about the conclusion for this project. This chapter start with the observation on the weakness and strength of the system. Based on the observation, there are some suggestions for the future improvement.

7.2 Project Summarization

This project is about bus tracking system – UTeM Bus Track. This system is implemented using android application and web hosting to manage the information in order to fulfil the user requirement. This system provides a bus tracking system to student which helps them to know the location of the bus.

The first objective for this project is to develop a system that can keep track on the bus location. In order to achieve this, an online database is needed in the web hosting site to allow the driver to send the GPS location to it. The student can use the system to track the bus location through accessing the data in the database.

The second objective is to calculate the bus arrival time. In order to accomplish this, Google Maps Distance Matrix API is used. This API is used in the android application to calculate the bus arrival time. The API uses the origin position, destination position and waypoints to generate JSON contain the distance and arrival time of the bus.

The last objective is to notify the student when the bus is almost reaches. This can be done by using the AlarmManager, BroadcastReceiver, and Service in the android application. When the phone need to alert, the AlarmManager will sent the message to start alarm to BroadcastReceiver, then the BroadcastReceive will start the Service to start the AlarmService.

The system has been tested and the system meets all objective. From the testing, the strength and the weakness of the system is observer. Besides, SWOT analysis is carried out to understand the value of this project. Figure 7.1 show the SWOT analysis for this project.

Strength	Weakness
• Able to track the bus location. • Able to estimate the bus arrival time. • Able to set the alert when the bus is almost arrive. • Able to view the bus schedule.	• Only support android. • Only able to use the Google Maps Distance Matrix with maximum usage of 2,500 elements per day. • Strong internet connection is required. • Bus icon movement is not smooth in the application
Opportunities	Threat
• More addition function can be added in future.	• Internet connection is crucial to ensure the user satisfaction.

Figure 7.1 SWOT Analysis

7.3 Project Contribution

UTeM Bus tracking using Google Map is aim to provide a bus tracking system in UTeM. This can benefits the students with the ability to allow the student to track the bus. Besides, the student also can set the alarm to alert student when the bus is almost reached, and view the bus schedule.

7.4 Project Limitation

In this project there are some limitations, such as good internet connection is needed to use the application. Besides, the application only available in android, which make other user can not enjoy this application. Moreover, there is a need of strong internet connection to run the application. Lastly, there is a limit use of the Google Map Distance Matrix API which mean after the API key is used up the bus arrival time will not be calculate.

7.5 Future work

There are many weaknesses in this system such as the application only support Android smart phone, the limited Google Maps Distance Matrix API usage to calculate the bus arrival time and strong internet connection are needed. Therefore some improvement can be made to overcome this weakness in the future work. Following shows the example of the future works.

a. Improve the application interoperability.

Currently, the system only support in android. This is unfair for the IOS users. Therefore, it is recommended to develop application which support IOS smart phone.

b. Develop bus arrival time algorithm

The current system uses the Google Maps Distance Matrix API to calculate the bus arrival time. This API has limited quota on how many

elements can runs. Therefore, there is a need to make an algorithm to calculate the bus arrival time.

c. Install kiosk at the bus stop

Currently, the student need to use own android device to track the bus. Besides, some students may not have android device. Therefore, the solution is to install a kiosk at the bus stop to allow the student to the bus without using own device.

7.6 Conclusion

By doing the planning, analysis, design, implementation and testing the system are finally complete. Based on the testing result, it can be assume that the system has fulfilled the requirement. The application can help the students to track the UTeM buses and prevent the student from missing the bus. Although most of the testing results are positive, there are still many improvement can be done to improve the system. I hope this system can contribute to the UTeM students to have a better shuttle service during their study.

REFERENCE

[1] Jianye Liu. (2011). "Research on Development of Android Applications"

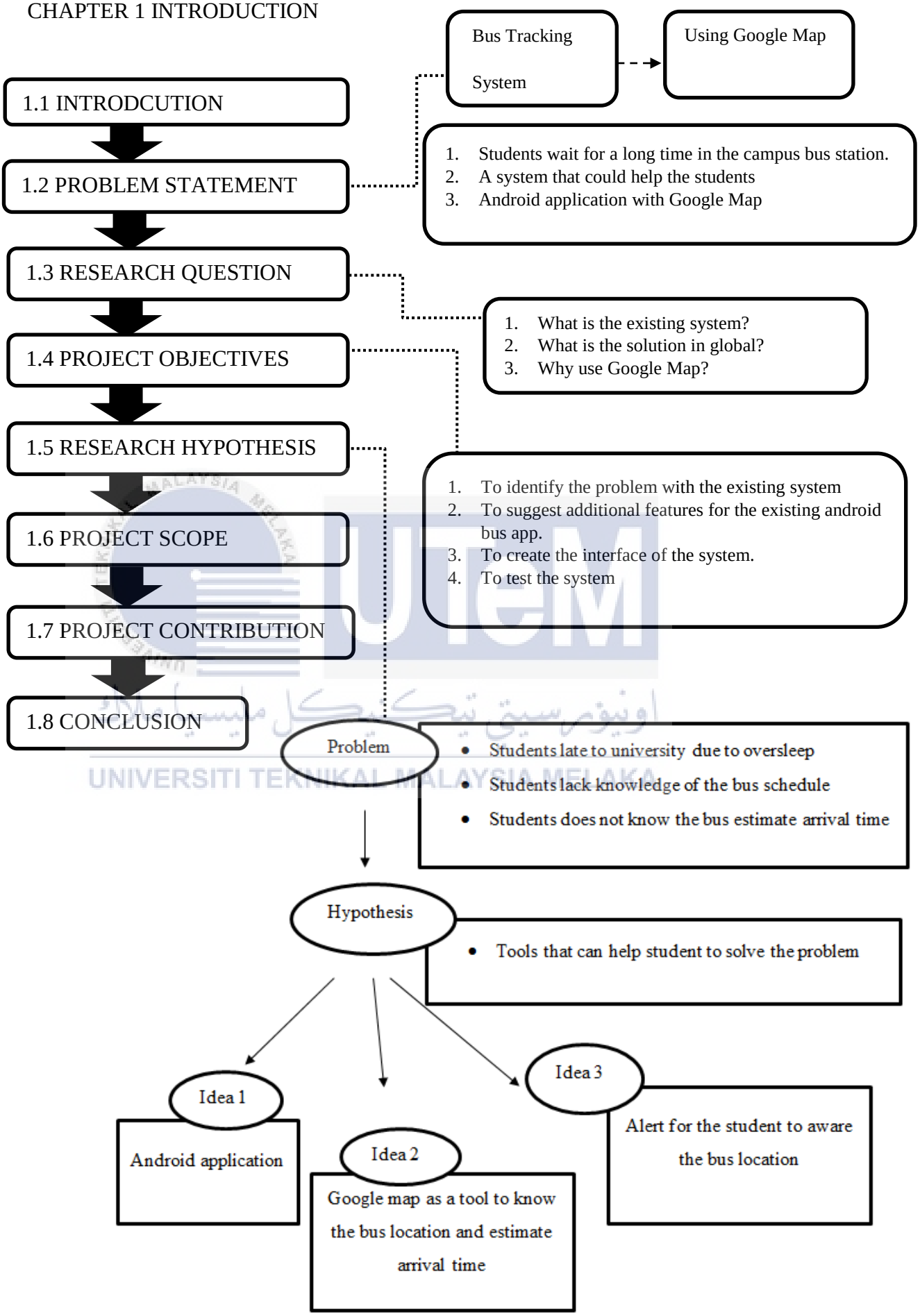
[2] Muthumurugesan, D. , Nalini S. , Vinodini R. (2013) "Smart Way to Track the Location in Android Operating System"

[3]UTHM Public Shuttle Tracking by KATSANA. (2015, November 23) Retrieved from https://uthm.katsana.com/#

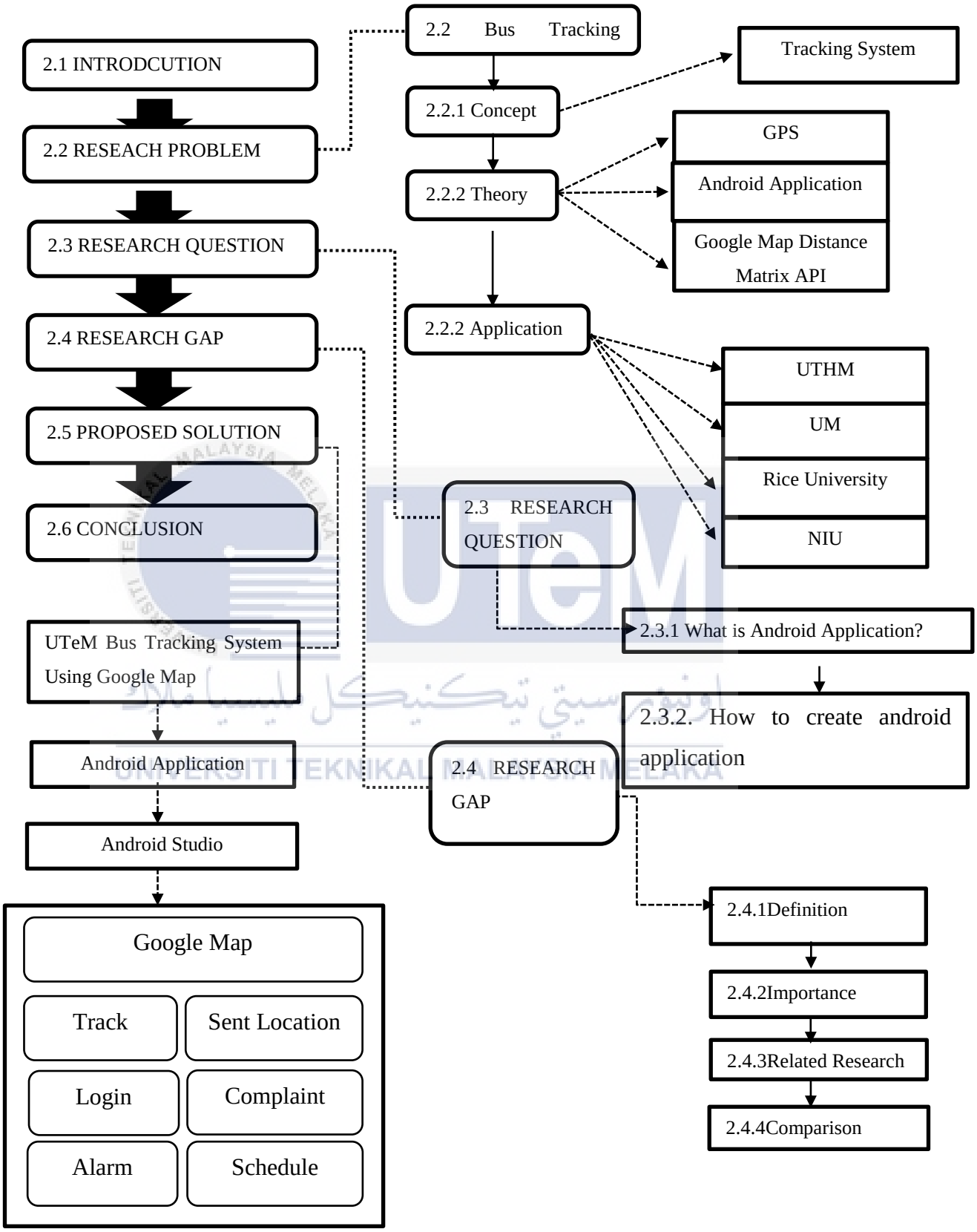
APPENDIX A

CHAPTER SUMMARY

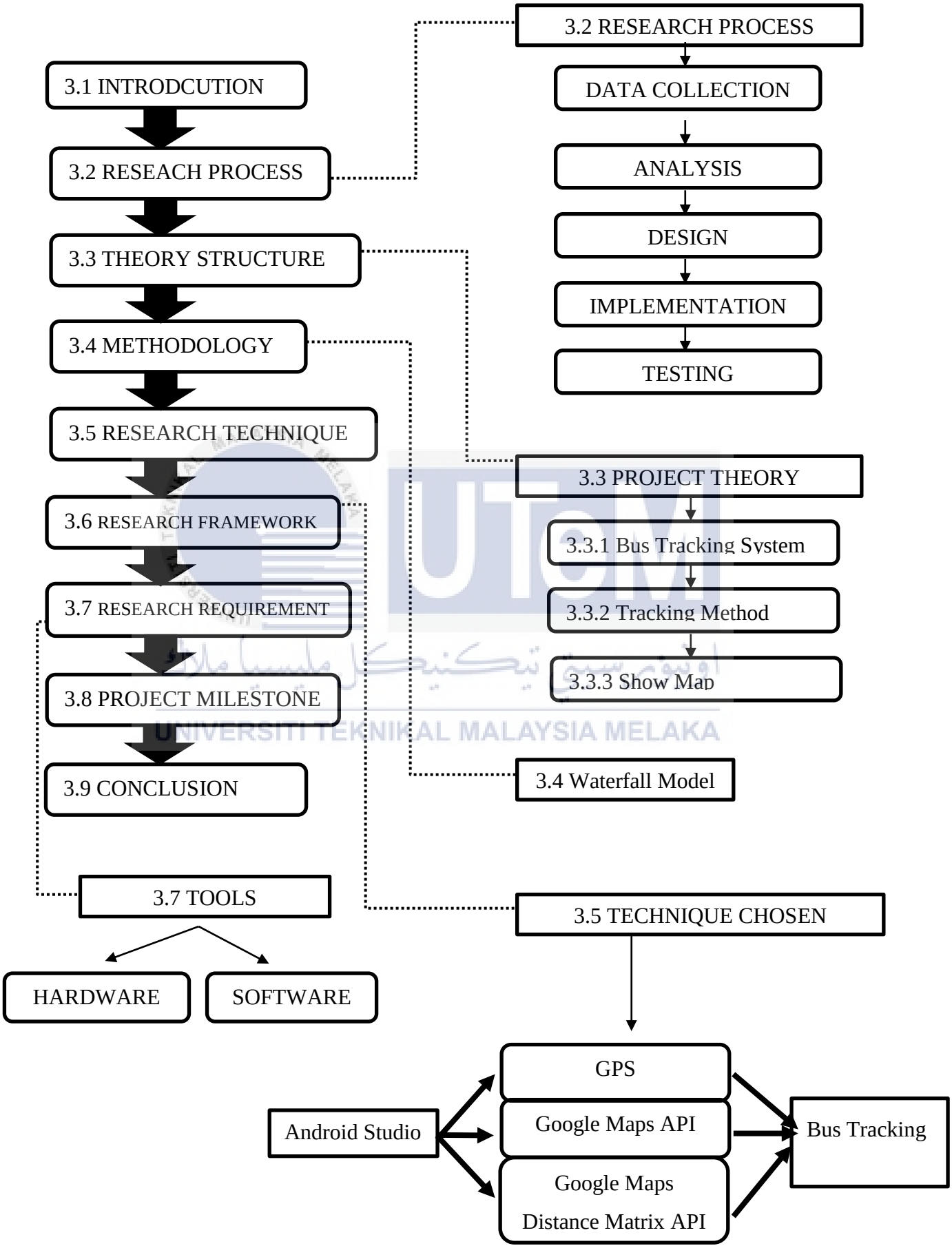
CHAPTER 1 INTRODUCTION



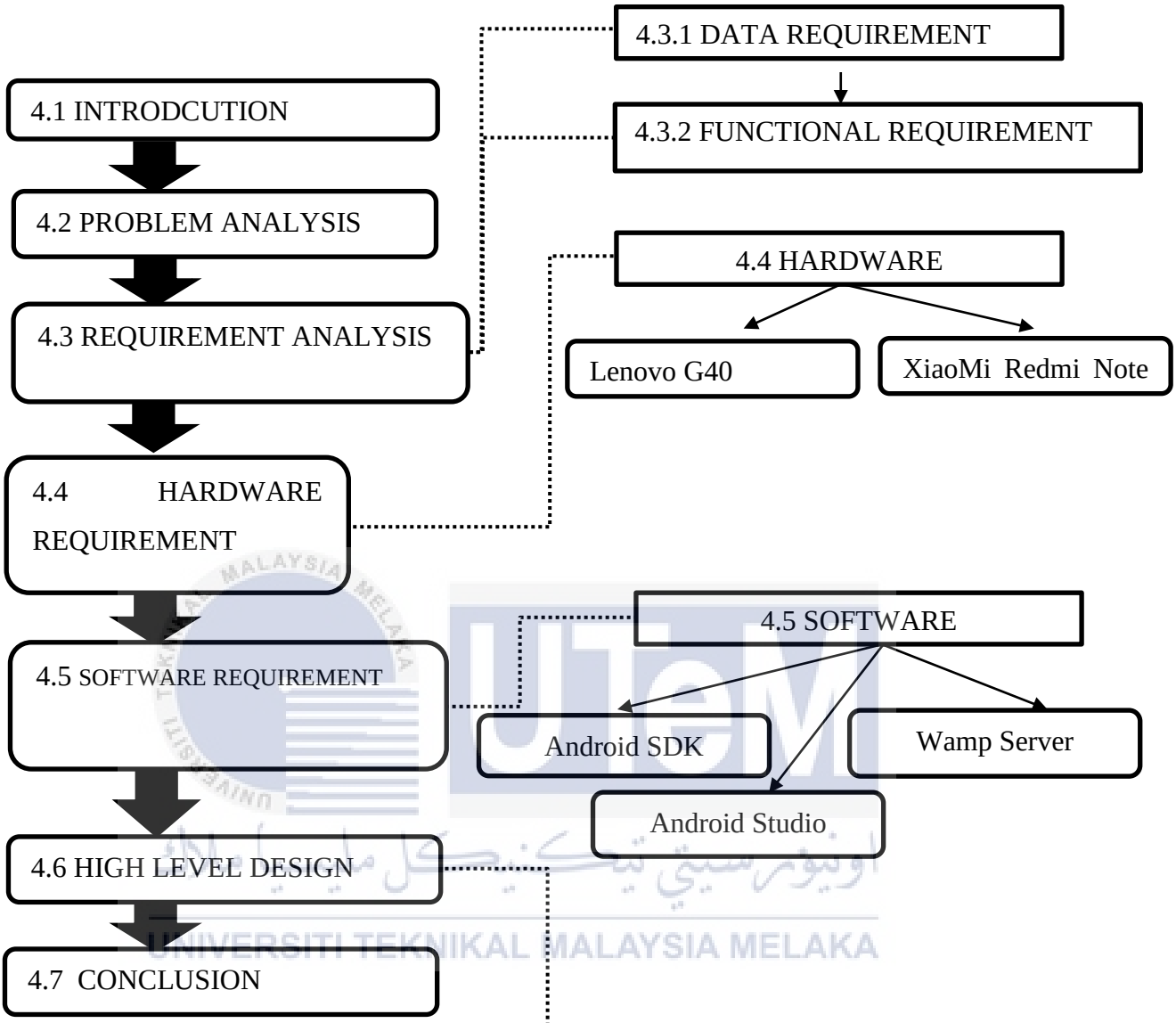
CHAPTER 2: LITERATURE REVIEW

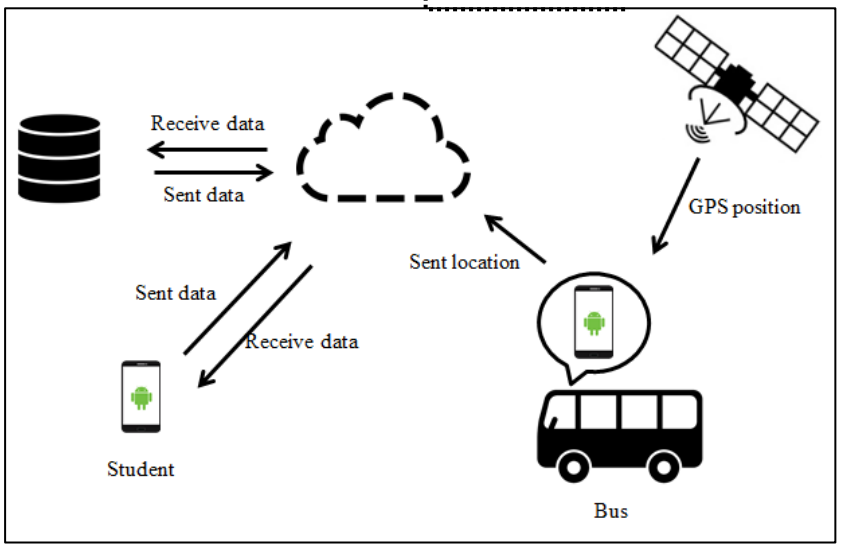


Chapter 3 Methodology

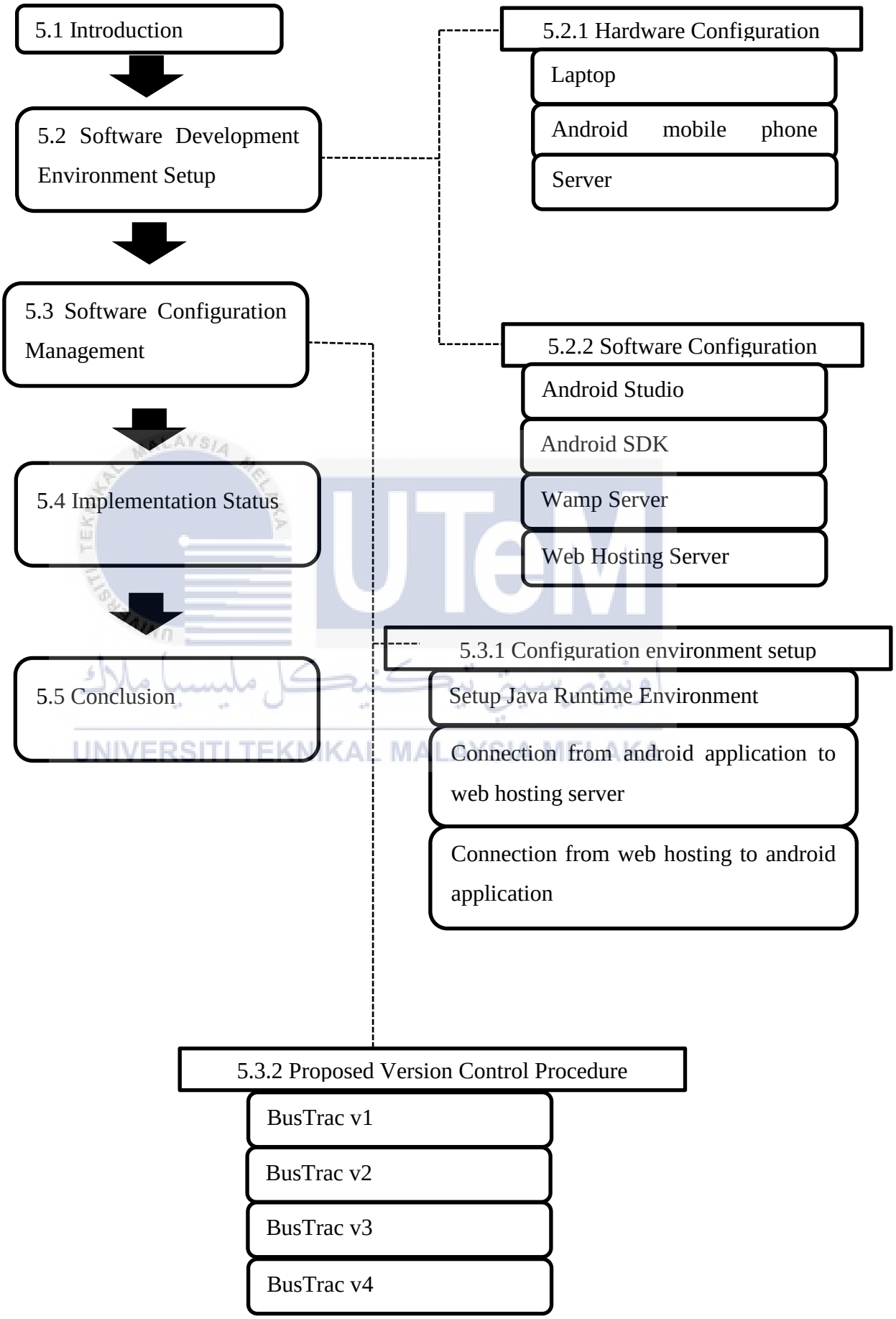


Chapter 4 Analysis and Design

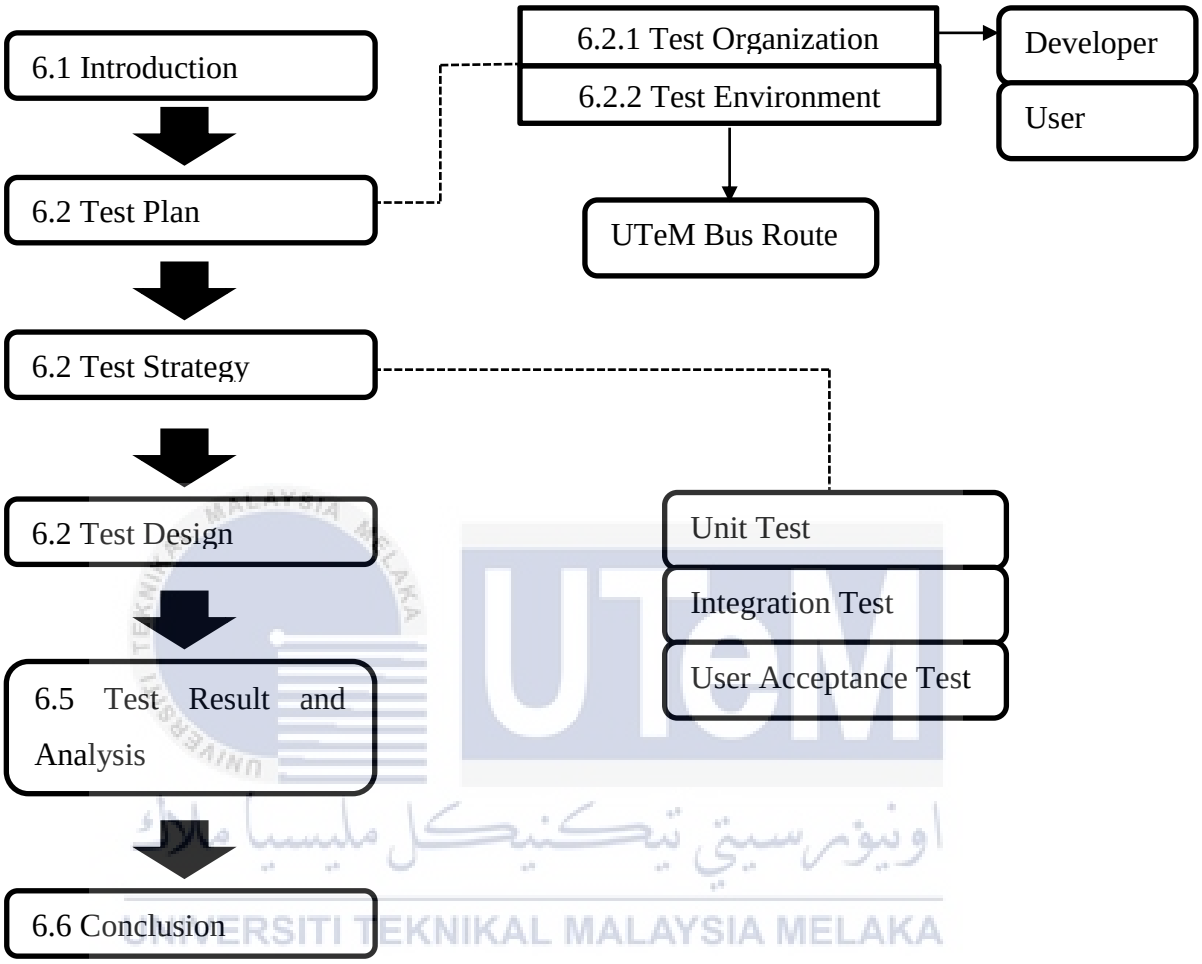




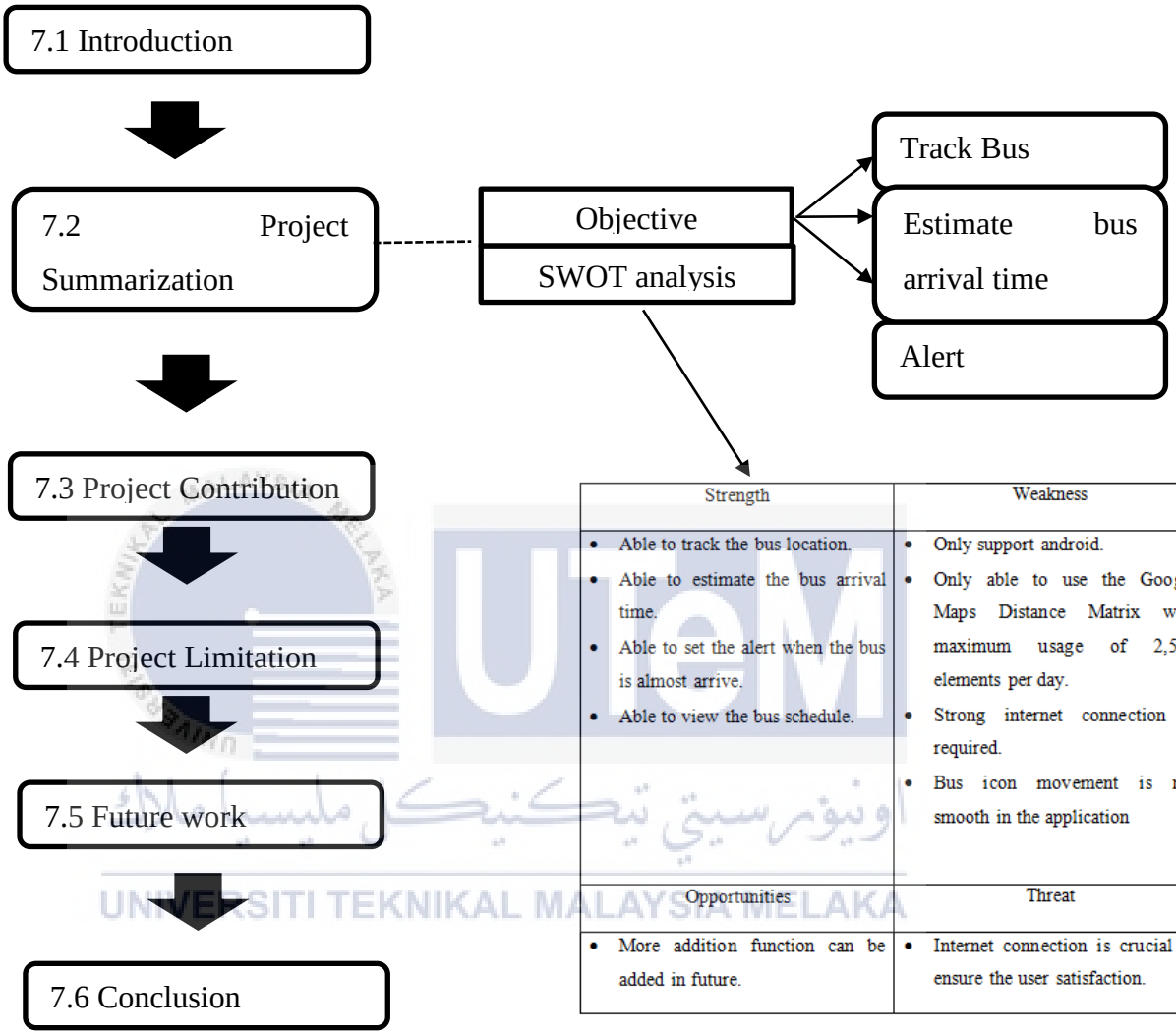
Chapter 5 Implementation



Chapter 6 Testing



Chapter 7 Conclusion



Strength	Weakness
• Able to track the bus location. • Able to estimate the bus arrival time. • Able to set the alert when the bus is almost arrive. • Able to view the bus schedule.	• Only support android. • Only able to use the Google Maps Distance Matrix with maximum usage of 2,500 elements per day. • Strong internet connection is required. • Bus icon movement is not smooth in the application
Opportunities	**Threat**
• More addition function can be added in future.	• Internet connection is crucial to ensure the user satisfaction.

Appendix B

User Acceptance Test

QUESTIONAIRE

Hello. The purpose of this questionnaire is to gather the user information in order to develop a bus tracking application. Your response will only be used for survey purposes. In case you have any questions regarding the survey, please contact Chen Jian Tat at 011-1087 7874. Thank you very much for your time and suggestions.

Part A: General Information

1. Name: _____

2. Gender:

☐ Male ☐ Female

3. Age: _____

Part B:

The following items describe user satisfaction. Indicate your agreement or disagreement with the following statements by ticking (✓) your response using this scale:

1	2	3	4	5
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

User Acceptance Test:

Description	1	2	3	4	5
The application is intuitive and easy to use					
The Google Map show the bus location correctly					
The bus arrival time is accurate and easy to see					
The alarm is useful to alert the user when the bus is almost reach					
User can view the bus schedule easily and clearly					
User is satisfy using this application to track the bus					

Thank you.