

# FIND ME FRIENDS: GPS TRACKER



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**BORANG PENGESAHAN STATUS LAPORAN AKHIR PROJEK SARJANA MUDA  
(PSM)**

JUDUL: Find Me Friends: GPS Tracker

SESI PENGAJIAN: 2016/2017

Saya SYAFIDA BINTI MOHAMMAD HASBULLAH mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hak milik 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: Lot 873, Kpg Petanak Daya,  
96400 Mukah, Sarawak

Tarikh: \_\_\_\_\_

\_\_\_\_\_  
(TANDATANGAN PENYELIA)

Nama penyelia: En. Muhammad Suhaizan  
bin Sulong

Tarikh: \_\_\_\_\_



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FIND ME FRIENDS: GPS TRACKER

SYAFIDA BINTI MOHAMMAD HASBULLAH



This thesis is submitted in partial fulfilment of the requirements for the Bachelor of Computer  
Science (Software Development)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

## DECLARATION

I hereby declare that this project report entitled

### **FIND ME FRIENDS: GPS TRACKER**

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



STUDENT : \_\_\_\_\_ Date: \_\_\_\_\_

(SYAFIDA BINTI MOHAMMAD  
HASBULLAH)

SUPERVISOR : \_\_\_\_\_ Date: \_\_\_\_\_

(MUHAMMAD SUHAIZAN BIN  
SULONG)

**DEDICATION**

To my beloved parents.



## ACKNOWLEDGEMENT

First and foremost, I would like to thank my supervisor, Mr. Muhammad Suhaizan bin Sulong, for his assistance and guidance throughout the project. He has helped me make it through the times when my ideas come to a standstill, and when I lack motivation to finish the project. Things were difficult, but I made it to the end. So, thank you.

Most importantly, I would like to thank my parents and my family for their unwavering support for me throughout this journey. They have sacrificed a lot for me to be where I am today and I am forever in their debt. Their encouragement and the love they gave for me is what keeps me going during the days in the university.



## ABSTRACT

The Find Me Friends: GPS Tracker is developed to address the problem of finding friends without the hassle of manually searching them or trying to contact them. The idea is to utilise the Global Positioning System (GPS) functionality embedded in a smartphone to detect friends or family by using this application. The project is developed in a period of 3 months. The project is inspired by the problem of finding people in a bustling place. The situation is similar when searching for a friend in a wide area. The project is also concern on the security of the person being search by exposing their location. The objective of this project is to build an application for tracking and locating, to study the GPS functionality of a smartphone, and to study the efficiency of GPS on a smartphone. The project is developed using Rapid Application Development methodology. The project mainly used the GPS detection method to locate the user's phone. The GPS detection method is the result of trilateration technique to determine the user's position, speed of their movement, and elevation from the ground. The technique involves the use of three or more satellites to pinpoint the user's location. The data from each satellite is used to narrow down where the user is from the overlapping area where the two sphere of the satellite detection meets. The use of more satellites will increase the accuracy of the location detection. Other than GPS detection, web service is also very crucial to the project as it handles the request from the application in the user's mobile phone. The web service handles the user authentication, location retrieval and storage, and friends' approval. The project is expected to grasp how GPS functionality in smartphone works. It will also hopefully be able to help people to find their family or friends and be more alert to their surroundings.



## ABSTRAK

Aplikasi Find Me Friends: GPS Tracker dibangunkan untuk menangani masalah mencari rakan-rakan tanpa perlu mencari secara manual atau cuba untuk menghubungi mereka. Idea ini adalah untuk menggunakan fungsi Sistem Kedudukan Global (GPS) terdapat dalam telefon pintar untuk mengesan rakan-rakan atau keluarga dengan menggunakan aplikasi ini. Projek ini dibangunkan dalam tempoh 3 bulan. Projek ini diilhamkan oleh masalah mencari orang di tempat yang sibuk. Keadaan ini adalah sama ketika mencari rakan di kawasan yang luas. Projek ini juga fokus kepada kebimbangan mengenai keselamatan orang yang dicari apabila mendedahkan lokasi mereka. Objektif projek ini adalah untuk membina sebuah aplikasi untuk mengesan dan mencari, untuk mengkaji fungsi GPS telefon pintar, dan untuk mengkaji kecekapan GPS pada telefon pintar. Projek ini dibangunkan menggunakan metodologi Pembangunan Aplikasi Rapid. Projek ini terutamanya digunakan kaedah pengesanan GPS untuk mengesan telefon pengguna. Kaedah pengesanan GPS adalah hasil daripada teknik pengigasegian untuk menentukan kedudukan pengguna, kelajuan pergerakan mereka, dan ketinggian dari tanah. Teknik ini melibatkan penggunaan tiga atau lebih satelit untuk menentukan lokasi pengguna. Data dari setiap satelit digunakan untuk mengecilkan mana pengguna adalah dari kawasan bertindih antara dua kawasan pengesanan satelit. Penggunaan lebih banyak satelit akan meningkatkan ketepatan pengesanan lokasi. Selain daripada pengesanan GPS, perkhidmatan web juga sangat penting kepada projek ini kerana ia mengendalikan permintaan daripada aplikasi dalam telefon mudah alih pengguna. Perkhidmatan web mengendalikan pengesanan pengguna, lokasi semula dan penyimpanan, dan kelulusan rakan-rakan. Projek itu dijangka untuk memahami bagaimana fungsi GPS dalam kerja-kerja telefon pintar. Ia juga diharap dapat membantu orang ramai untuk mencari keluarga atau rakan-rakan mereka dan menjadi lebih peka kepada persekitaran.

## TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	TABLE OF CONTENTS	vii
	LIST OF TABLES	x
	LIST OF FIGURES	xii
	LIST OF ABBREVIATIONS	xiii
	LIST OF ATTACHMENTS	xv
	INTRODUCTION	1
1.1	Introduction	1
1.2	Problem statement(s)	2
1.3	Objective	2
1.4	Scope	2
1.5	Project significance	3
1.6	Expected output	3
1.7	Conclusion	4
	LITERATURE REVIEW AND PROJECT METHODOLOGY	5
2.1	Introduction	5
2.2	Facts and Findings	6
2.2.1	Tracking Technology	6
2.2.2	Application of Tracking Technology for Tracking People	8
2.2.3	Existing System	10
2.2.4	Comparison Between the Existing System	17
2.2.5	Technique	18
2.3	Project Methodology	19
2.3.1	Planning	20
2.3.2	Analysis	21
2.3.4	Design	21
2.3.5	Implementation	22

2.3.6	Testing	22
2.3.7	Deployment	23
2.4	Project Requirements	23
2.4.1	Software Requirement	23
2.4.2	Hardware Requirement	24
2.5	Project Schedule and Milestones	24
2.6	Conclusion	27
<b>ANALYSIS</b>		28
3.1	Introduction	28
3.2	Problem Analysis	29
3.2.1	How Current System Works	29
3.2.2	Analysis of Problem Statements	30
3.3	Requirement Analysis	31
3.3.1	Data Requirement	31
3.3.2	Functional Requirement	33
3.3.3	Non-Functional Requirement	33
3.3.4	Others Requirement	34
3.4	Conclusion	36
<b>DESIGN</b>		37
4.1	Introduction	37
4.2	High-Level Design	38
4.2.1	System Architecture	38
4.2.2	User Interface Design	44
4.2.3	Database Design	50
4.3	Detailed Design	55
4.3.1	Software Design	55
4.3.2	Physical Database Design	65
4.4	Conclusion	67
<b>CHAPTER V</b>		69
<b>IMPLEMENTATION</b>		69
5.1	Introduction	69
5.2	Software Development Environment Setup	70
5.3	Software Configuration Management	71
5.3.1	Configuration Environment Setup	71
5.3.2	Version Control Procedure	72

5.4	Implementation Status	73
5.5	Conclusion	73
TESTING		75
6.1	Introduction	75
6.2	Test Plan	76
6.2.1	Test Organization	76
6.2.2	Test Environment	76
6.2.3	Test Schedule	77
6.3	Test Strategy	78
6.3.1	Classes of Tests	79
6.4.	Test Design	80
6.4.1	Test Description	80
6.4.2	Test Data	83
6.5	Test Result and Analysis	86
6.6	Conclusion	88
CHAPTER VII		89
CONCLUSION		89
7.1	Observation on Weaknesses and Strengths	89
7.2	Proposition for Improvement	90
7.3	Project Contribution	91
7.3	Conclusion	91
REFERENCES		92
BIBLIOGRAPHY		95
APPENDICES		96

## LIST OF TABLES

TABLE	TITLE	PAGE
	<b>Table 2.1: Comparison between the existing system</b>	<b>17</b>
	<b>Table 2.2: High-level project schedule</b>	<b>24</b>
	<b>Table 2.3: Project milestone</b>	<b>26</b>
	<b>Table 3.1: User table</b>	<b>31</b>
	<b>Table 3.2: Friend table</b>	<b>32</b>
	<b>Table 3.3: Location table</b>	<b>32</b>
	<b>Table 3.4: List of software and its justification of usage</b>	<b>34</b>
	<b>Table 3.5: List of hardware and its justification of usage</b>	<b>35</b>
	<b>Table 4.1: User table</b>	<b>52</b>
	<b>Table 4.2: Friend table</b>	<b>53</b>
	<b>Table 4.3: Location table</b>	<b>53</b>
	<b>Table 4.4: User class</b>	<b>55</b>
	<b>Table 4.5: Register class</b>	<b>56</b>
	<b>Table 4.6: Login class</b>	<b>58</b>
	<b>Table 4.7: User class</b>	<b>59</b>
	<b>Table 4.8: Friend class</b>	<b>60</b>
	<b>Table 4.9: User class</b>	<b>61</b>
	<b>Table 4.10: Location class</b>	<b>62</b>
	<b>Table 4.11: User class</b>	<b>63</b>
	<b>Table 4.12: Calculate class</b>	<b>63</b>
	<b>Table 4.13: Location class</b>	<b>64</b>
	<b>Table 5.1: Version control procedure</b>	<b>72</b>
	<b>Table 5.2: Find Me Friends: GPS Tracker implementation status</b>	<b>73</b>
	<b>Table 6.1: Testing device configuration</b>	<b>77</b>
	<b>Table 6.2: Test schedule</b>	<b>78</b>
	<b>Table 6.3: Test classes</b>	<b>79</b>
	<b>Table 6.4: Test case</b>	<b>80</b>
	<b>Table 6.5: Test data</b>	<b>83</b>

**Table 6.6: Test Step**

84

**Table 6.7: Test Result**

86



## LIST OF FIGURES

DIAGRAM	TITLE	PAGE
	Figure 2.1: Requesting friend's location	10
	Figure 2.2: Sending alerts	11
	Figure 2.3: Sharing location and alert on Facebook	11
	Figure 2.4: User's location tracking history	12
	Figure 2.5: Life360 home screen	13
	Figure 2.6: Help alert	14
	Figure 2.7: User option	15
	Figure 2.8: Family Locator navigation menu	16
	Figure 2.9: Create zone	17
	Figure 2.10: Rapid Application Methodology	20
	Figure 4.1: Find Me Friends: GPS Tracker architecture	38
	Figure 4.2: Class diagram for user authentication	39
	Figure 4.3: Class diagram for friend authentication	39
	Figure 4.4: Class diagram for user detection	40
	Figure 4.5: Class diagram for distance calculation	40
	Figure 4.6: Sequence diagram for user authentication	41
	Figure 4.7: Sequence diagram for friend authentication	42
	Figure 4.8: Sequence diagram for user detection	43
	Figure 4.9: Sequence diagram for distance calculation	44
	Figure 4.10: Use case for Find Me Friends: GPS Tracker	45
	Figure 4.11: Login screen	46
	Figure 4.12: Register screen	47
	Figure 4.13: Search friend screen	47
	Figure 4.14: User location screen	48
	Figure 4.15: Friend list screen	49
	Figure 4.16: Distance calculation result	49
	Figure 4.17: Conceptual database design	50
	Figure 4.18: Logical database design	51
	Figure 5.1: Deployment diagram	70

## LIST OF ABBREVIATIONS

GPS	-	Global Positioning System
RFID	-	Radio Frequency Identification
UWB	-	Ultra-wideband
WLAN	-	Wireless Local Area Network
WHO	-	World Health Organization
IBM	-	International Business Machines
RPM	-	Remote patient monitoring
RSSI	-	Received signal strength indication
BLE	-	Bluetooth low energy
SIM	-	Subscriber identification module
SMS	-	Short Message Service
Plc	-	Public limited company
GSM	-	Global System for Mobile communication
3G	-	Third generation
WCDMA	-	Wide Band Code Division Multiple Access
CDMA	-	Code Division Multiple Access
IDE	-	Integrated development environment
UML	-	Unified Modelling Language
FTP	-	File transfer protocol
PHP	-	Hypertext pre-processor
MS	-	Microsoft
OS	-	Operating system
CPU	-	Central processing unit
RAM	-	Random access memory
ID	-	Identification
XML	-	Extensible markup language
REST	-	Representational State Transfer
GUI	-	Graphical user interface
HTTP	-	Hypertext transfer protocol
JSON	-	JavaScript Object Notation
IDE	-	Integrated development environment



API	-	Application program interface
VCS	-	Version control system
FMF: GT	-	Find Me Friends: GPS Tracker



**LIST OF ATTACHMENTS**

<b>ATTACHMENT</b>	<b>TITLE</b>	<b>PAGE</b>
A	Proposal Form	97
B	User Manual	105
C	Find Me Friends: GPS Tracker screenshot	106
D	Gantt Chart	108



## CHAPTER I

### INTRODUCTION



#### 1.1 Introduction

Global Positioning System (GPS) has made location tracking relatively easy. GPS has been used for determining a position, navigating from one point to another, monitoring the movement of an object, creating maps, and providing precise timing. The versatility of GPS and availability of GPS-equipped devices has expanded the use of GPS into our everyday lives.

Find Me Friends: GPS Tracker is an Android application that use the GPS function to locate other user's location and display it on a map for tracking. The application can be used to find the location of a friend in a big, crowded place by locating their smartphone with this application on. This project is made up of four modules that are user authentication, friend authentication, user detection, and distance calculation.

## 1.2 Problem statement(s)

### i. Problem statement 1: Locating a person in a crowded place is difficult.

- A crowd of people cannot be avoided at the festivals, or shopping malls, or parks. In this situation, locating a person is challenging. In cases involving children, the children may get into some accident, or worst-case scenario, they might get kidnapped.

### ii. Problem statement 2: Locating a person in a massive expanse takes too much time.

- Time is not something that is to be wasted when things run on schedule. The other person may wander too far off from the person and they would want to find them quickly to comply to the schedule they have.

### iii. Problem statement 3: Exposing the exact location of a person can be very dangerous.

- Nowadays, data is very valuable. Exposing information such as one's current location to just about anyone can be bad. People can take advantage of this maybe by kidnapping, or robbing the person.

## 1.3 Objective

- Objective 1: To develop an application for user locating and tracking.
- Objective 2: To explore the GPS function on smartphone.
- Objective 3: To evaluate the tracking capabilities of GPS on smartphone.

## 1.4 Scope

The targeted user of this application is public. The application will consist of four modules that are user authentication, friend authentication, user detection, and distance calculation. This

application will be developed within 3-months period. The project is expected to finish in May 2017.

### 1.5 Project significance

This project will hopefully be able to help better understand how GPS on smartphone works. This project is also hopefully will be able to help people to keep track of their family or friends and increase people's awareness to their surroundings. This application has the potential to be used during pilgrimage in Mecca where a *Tabung Haji* employee can keep track of the pilgrims under their supervision. This application also has the potential to be used by parents to keep track of their children's location when they are not in the vicinity of the parents. Besides that, the application also can be used by prisoner on probation period, where the police can track their location using the location of their smart phone.

### 1.6 Expected output

This project is expected to produce the following outputs:

- Output 1: Android application that explore the use of GPS on smartphone.
- Output 2: Android application that ease the task of locating and tracking a person.
- Output 3: Android application that provide a secure way of sharing location.

## 1.7 Conclusion

As a conclusion, this chapter discuss about the background of the Find Me Friends: GPS Tracker. Find Me Friends: GPS Tracker is a GPS-based application developed for Android smartphone. This application is aimed to be used by people that wish to track the location of their family, friends, or people under their care. This project will explore the use of GPS on smartphone. This project will hopefully be able to assess the advantage and limitation of GPS capability on a smartphone. The next activity is to research facts and findings related to the Find Me Friends: GPS Tracker and the methodology which will be used to develop this application.



## CHAPTER II

### LITERATURE REVIEW AND PROJECT METHODOLOGY



#### 2.1 Introduction

This chapter is going to discuss about the facts and findings regarding the Find Me Friends: GPS Tracker to understand the concept, theories, and technologies related to the application and its domain. This chapter is also going to analyse the existing systems that are similar to the application that is being developed. The selected methodology for the development of this project is also going to be discussed. Next, the software, hardware, and other requirements of this project is going to be listed. Lastly, the project schedule and milestone are going to be listed.

## 2.2 Facts and Findings

This section is going to discuss about the domain of this project, tracking technology, the use of tracking technology, and the advantages of this technology in 2.2.1 through 2.2.3.

### 2.2.1 Tracking Technology

#### 2.2.1.1 Introduction to Tracking Technology

The domain of this project is tracking technology. According to Oxford Dictionary online, track is an act of following the trail or movement of a person or object, often to find or mark their course. Tracking technology, on the other hand, is an application of a collection of technology to create a tracking system. Some of the technology used in tracking system are Global Positioning System (GPS), Radio Frequency Identification (RFID), Cellular-Based, Ultra-wideband (UWB), Wireless Local Area Network (WLAN), and Bluetooth (Hui Liu et al., 2007).

Among these technologies, GPS is widely used for outdoor environment positioning system such as vehicle tracking, navigation and law enforcement. However, the system became limited when it is used indoor due to the limited or poor satellite signal coverage. Despite this limiting factor, GPS is the most direct way to track a smartphone because the presence of GPS as built-in features in most smartphones today (Hao Tang et al., 2016). WLAN also got many attentions because of its availability in the real-world environment. It is used in WLAN location fingerprinting.



### 2.2.1.2 Advantages of Tracking Technology

The extensive use of tracking technology nowadays has both its advantages and disadvantages. The idea of being track or monitored by anyone does not usually sit well with many people. Some even said that it breaches their privacy and violate their rights. However, using technology to track or monitor people or object does not have to be scary or creepy as always portrayed in movies or television series. It can be very beneficial and can improve one's lifestyle or care for their family.

Tracking technology can help in consumerism. Tracking technology is used to identify consumer preference and provide the best service or product that meets the consumer demands and needs. Facebook is infamous for its tracking the user web browsing behaviour to provide advertisement that the user can relate to or can be useful to the user. While it may be a nuisance to some, the move proved to be a successful marketing technique to companies who advertise their business on Facebook.

Tracking technology is also can come in handy for personal navigation when a person is lost or is trying to find a location. In these situations, the built-in GPS capabilities in the smartphones and mobile application such as Google Maps are very useful. Application such as Google Map utilize the GPS capability of the smartphone to locate the position of the user, and later offer suggestions for personal navigation. The information also can be used to plan a journey, estimate the travel time, and choose the best travel route.

Besides that, tracking technology is also very useful in emergency situations such as tracking a lost or missing person. The information gathered is usually gathered from a device used by the targeted person such as a smartphone. The data are collected from GPS receiver of the smartphone and from the mobile telecommunication provider to determine the device's location relative to nearby cell tower. The information can later be used by the police force, fire department and hospitals to begin the rescue mission.

## 2.2.2 Application of Tracking Technology for Tracking People

### 2.2.2.1 Tracking Technology for Tracking Disabled Persons

The 2008 Disability Act defined disabled person as those who have long term physical, mental, intellectual or sensory impairments which, in interaction with various barriers, may hinder their full and effective participation in society. World Health Organization estimates that there are over 1 billion people globally that experience disability. Disabled persons for the most part have poor health, low education achievements, fewer economic opportunities, and higher rate of poverty than their normal counterparts. Due to this, many activists and organization such as WHO are actively finding ways for disabled persons to have better opportunity in society.

Many systems and approaches were developed to aid and assist disabled persons to enhance their quality of life. Study conducted by Khaled Salhi et al. (2016) was aimed to create a secure navigation of powered wheelchairs for disabled person with mobility impairment. The wheelchair is for indoor environment use only. On the navigation part, of the system focused on moving the wheelchair to follow the caregiver and to detect and avoid obstacles. The tracking part of the system is to track the caregiver and the other wheelchair. The system use the concept of marching elephant which is to follow the cue of the leader of the platoon.

### 2.2.2.2 Tracking Technology for Tracking Employees

Employee monitoring have existed as early as 1888 by Computing Tabulating Recording Corporation, which is now known as International Business Machines (IBM). The corporation created the first time clock intended to record the employees' attendance by punching in and out of work. The employee monitoring system have evolved a lot since then. Nowadays, the employee monitoring system includes internet usage, email, phone use, and even employee's

location tracking. These employee monitoring system is developed to lessen wasted time while at work, improve employee insights, increase security of the organization, and lessen administrative work.

Sasisthrani Chandrasekaran et al. (2016) conducted a study to create a method to track employee in restricted area of an organization. The system's focuses on location tracking and authentication. The application itself is built on Android Operating System. The location tracking uses the GPS on the smartphone of the employee, a reference map, and applies Loyd's algorithm. The GPS is used to locate the coordinate of the employee's device, while the reference map is used together with Loyd's algorithm to compute employee's location in the given area. The authentication uses biometric technique to ensure the integrity of the employee.

#### 2.2.2.3 Tracking Technology in Remote Healthcare Systems

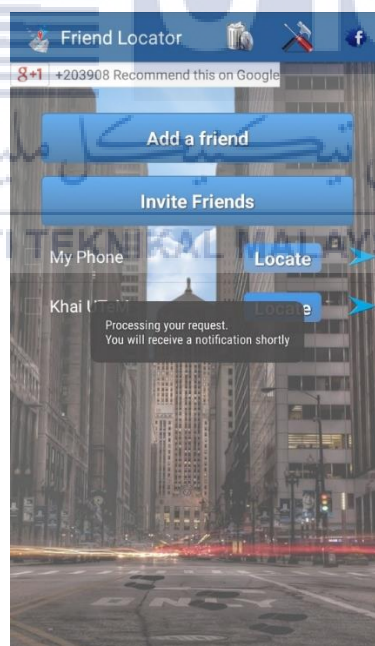
Remote patient monitoring (RPM) is a system to monitor patients outside of primary care setting, such as hospital, to healthcare provider by using technology. The RPM is often used with patients with high medical needs such as the elderly and the chronically ill. RPM allows the patients to stay at home longer while being under continuous monitoring by their physicians. RPM also allows the physician to detect any health issues and intervene if the need arise. RPM provides better access to healthcare, improve the quality of healthcare, and provides assurance and support to the patient and family.

A study conducted by Po-Chou Liang et al. (2016) focuses tracking the patient's movement because monitoring daily movement patterns of patient can show early sign or worsening health condition. The system is built on Apple's iOS and comprises of several low-cost sensors. The system is composed of step detection, indoor radio-based location tracking using received signal strength indication (RSSI) from three Bluetooth low energy (BLE) sensors, and Kalman filter for the localization algorithm. The Agent Module of the system is deployed in an iPhone 5. The result of the study shows the average estimation error of up to 0.47m.

### 2.2.3 Existing System

#### 2.2.3.1 Application 1: Friend Locator

Friend Locator is an application that tracks a friend's location using their mobile phone. The features of this application include its ability to receive notification about the movement of friends, track and locate a person by mobile number, track the phone location using GPS and cell tracking, and its service is totally free. The application works in background and is said to work even if the SIM is removed or replaced. Friend Locator has a simple and easily understandable interface. However, the application does not work. It can track the user's location but it is not able to track the friend's location. The notification is only about the user's location and not the friend's location. The alert message sent is not received by the friend. The application features are good but some of its features does not work.



**Figure 2.1: Requesting friend's location**

To start using the location tracking function of the application, the user must create an account. Then, to add friend to track, the user must add friend into their friend list. If the friend has not created an account, the user can invite the other user to download, install and register the

application. Then, the user can start to track their friend using the locate button. A Toast will appear to show that the request is processed and the user will be notified once the application has successfully tracked the friend (refer Figure 2.1).

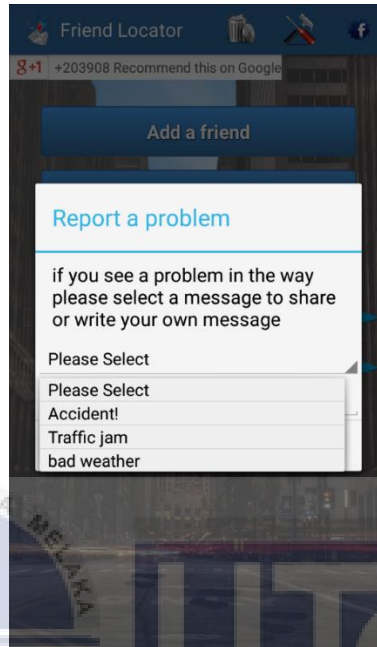


Figure 2.2: Sending alerts

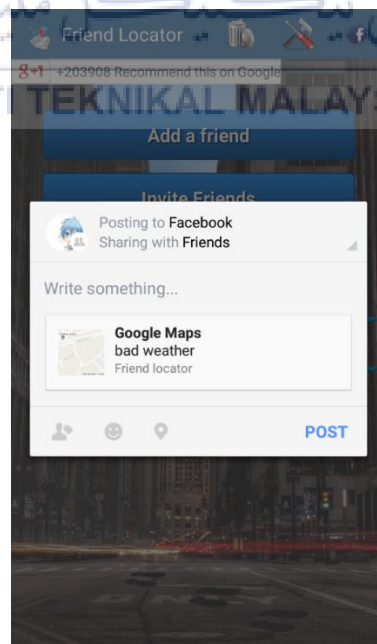


Figure 2.3: Sharing location and alert on Facebook

Figure 2.2 shows how the application can send alert to friends via Facebook plugin. This plugin will automatically detect the account in the user's Facebook application in the user's device. The user has the option to choose a list of predefined messages or they can write their own message. The message or the problem will be posted to the user's Facebook timeline as a status. Figure 2.3 shows how the message, and the location will be posted on Facebook.

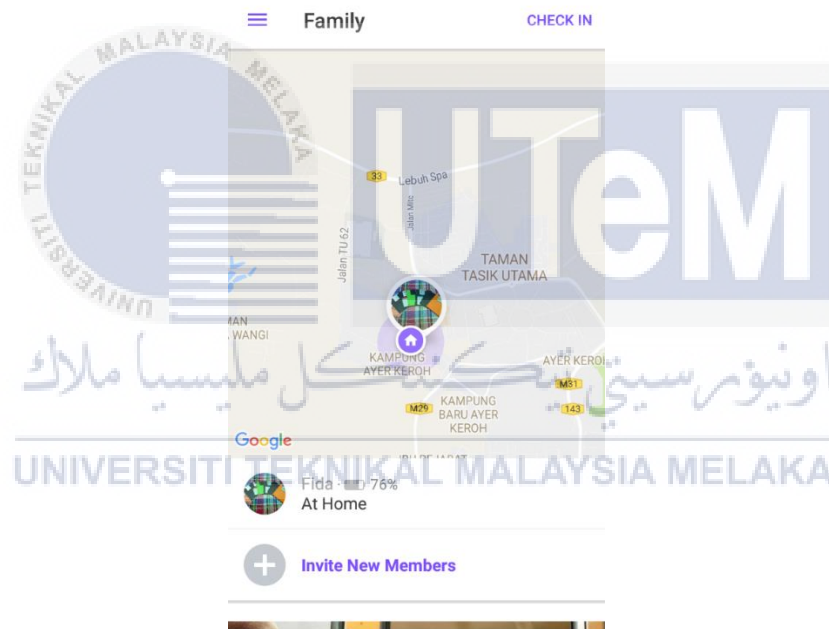


**Figure 2.4: User's location tracking history**

This application also provides the location history of the user themselves and their friends (refer Figure 2.4). In the history, the user can view the address of their location, the battery level, the date and time, and problem reported by the user at the time of tracking, if any. In the same screen, there is the option to delete the location history, a button to stop location tracking and a button to view the user locations on map. However, the map function is not working properly as it only shows the user's current location only.

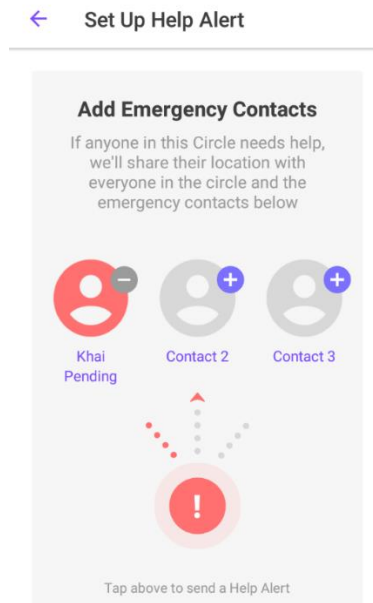
### 2.2.3.2 Application 2: Life360

Life360 Family Locator is another application that utilise GPS on mobile device to track friends and family location. Life360 Family Locator main features are it can organise family and friend list in groups called “Circles”, chatting with the circles, and receive real-time notifications when the circle members arrive or leave destinations. The application’s interface is easy to navigate and learn for first time users. The application delivers what it is promised but some features such as location history is only available for premium users. Some of its most interesting features is it can show the phone battery percentage of the circle members, send alert messages via notification and SMS to emergency contact numbers, and share list, such as shopping list, or to-do list, with circle members.



**Figure 2.5: Life360 home screen**

Figure 2.5 shows what are the elements that the user will see on the application. The user will be able to immediately see their own location and their circle members’ location and the status of the other member’s phone. To differentiate each user on the map, the pin displayed the user’s profile picture. They are also able to check in and the other member will be notified of their current location. They also can invite new members to their circle.



**Figure 2.6: Help alert**

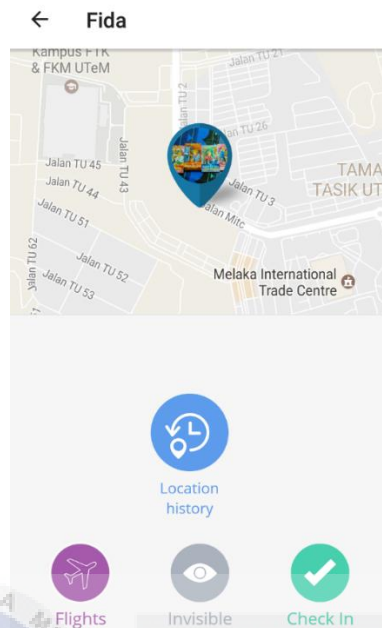
The help alert of the application helps the user to send emergency message to the emergency contacts assigned by the user in the event of an emergency. The user simply has to tap on the exclamation button (refer Figure 2.6) to send the emergency message. The emergency message is sent in the form of email and SMS to the emergency contacts. The email contains the latitude and longitude of the user's current location and the time of the help alert is pressed. After the email is sent, the application prompts the user to call emergency helpline at 112. The SMS, on the other hand, only contains a message to ask the other user to be their emergency contact.

### 2.2.3.3 Application 3: Family Locator

Family Locator – Phone Tracker is another application that tracks user's device. This application features are showing real-time location of family and friends, sending and receiving SOS with exact location, receiving notification when family or friends arrive at certain destination, and receiving alerts when family or friends leave safe zone. These features are available for free. Similar to Life360, the location history is only available as premium features. Aside from the simple interface, the application has beautiful and interactive user interface.

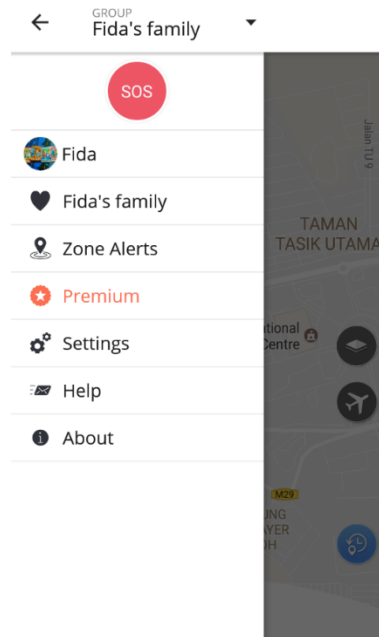


The extra features of this application is the invisible mode where user can turn off the tracking option, track flight based on its flight number, and the alert notification.



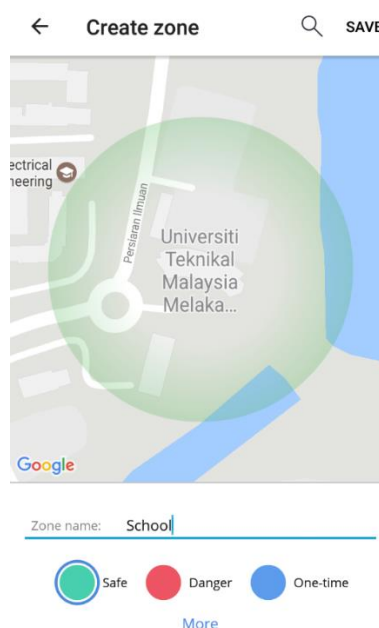
**Figure 2.7: User option**

Figure 2.7 shows the user option that is available on the user home screen when the user click on their location's pin. This application provides extra function that is the invisible mode, where other users cannot track their location unless the mode is switched off. The application also provides the option to check in where the other users will be notified of their current location. The special feature in this application is the flight locator. The user only has to enter the flight number and the application will display the details about the flight such as the departure destination, arrival destination, the date and time of the flight.



**Figure 2.8: Family Locator navigation menu**

The most interesting function in the Family Locator navigation menu is the SOS button at the top (refer Figure 2.8). The button is used to send SOS message to the user's emergency contacts. The emergency contacts are notified of the user's distress via email. The email requires the emergency contact to open the application in order to see the user's position. Other than that, the navigation menu also offers the user to view different groups that the user joined.



### Figure 2.9: Create zone

Another interesting function that the application has is the function to create zone. There are three categories of zone; safe zone, danger zone, and one-time zone (Figure 2.9). These zones are used when the user that is being tracked entered the zone, the other member that created the zone will immediately be notified. This function is especially useful when parents need to know the location of their children, whether they are in a safe place or dangerous place, as deemed by the parents, without the hassle of having to constantly contacting their children.

#### 2.2.4 Comparison Between the Existing System

The comparison of features and additional features between all the applications in the existing system is summarized in the following table.

Table 2.1: Comparison between the existing system

	<b>Friend Locator: Phone Tracker</b>	<b>Family Locator – GPS Tracker</b>	<b>Family Locator – Phone Tracker</b>
<b>Platform</b>			
<b>Mobile</b>	Yes	Yes	Yes
<b>Web</b>	No	No	No
<b>Operating system</b>			
<b>iOS</b>	No	Yes	No
<b>Android</b>	Yes	Yes	Yes
<b>Features</b>			
<b>GPS tracking</b>	Available	Available	Available
<b>Map view</b>	Not available	Available	Available

<b>Location notification</b>	When friends are nearby	When Circle Members arrive, or leave destinations	When friends or family reach a destination
<b>SOS message</b>	Problem reporting (accident, bad weather, traffic jam) via Facebook post	SOS message with current location is sent via email and push notification to emergency contact	SOS message is sent via email and push notification to emergency contact
<b>Location history</b>	Available and unlimited	Available up to seven days before for free	Available up to two days before for free
<b>Special features</b>			
<b>Cellular tracking</b>	Available	Not available	Not available
<b>Tracking notification</b>	User is alerted when they are being tracked	Not available	Not available
<b>Stop real-time tracking</b>	Not available	Available, location sharing via check-in	Available
<b>Chat</b>	Not available	Not available	Available

### 2.2.5 Technique

Based on previous studies conducted on location tracking, there are several other techniques to locate a person. Each technology that is used in location tracking is limited to a degree, thus giving advantage to another technology. The technique that will be used in the proposed system is using GPS on smartphone. The technique is used because smartphone is widely used by the

society today and GPS is the simplest technique for tracking. Among other alternative technique for tracking is using WLAN, Bluetooth and RFID.

There are several WLAN localization techniques exists today. The most commonly used technique is WLAN fingerprinting. WLAN fingerprinting is a localization technique that associate an individual position with a distinct feature and apply them to pinpoint absolute locations of target (Jiwei Li, 2012). This technique is not used for the proposed system because the user of the proposed system may not be connected to a WiFi for a length of time. Thus, this can limit the ability to track the person.

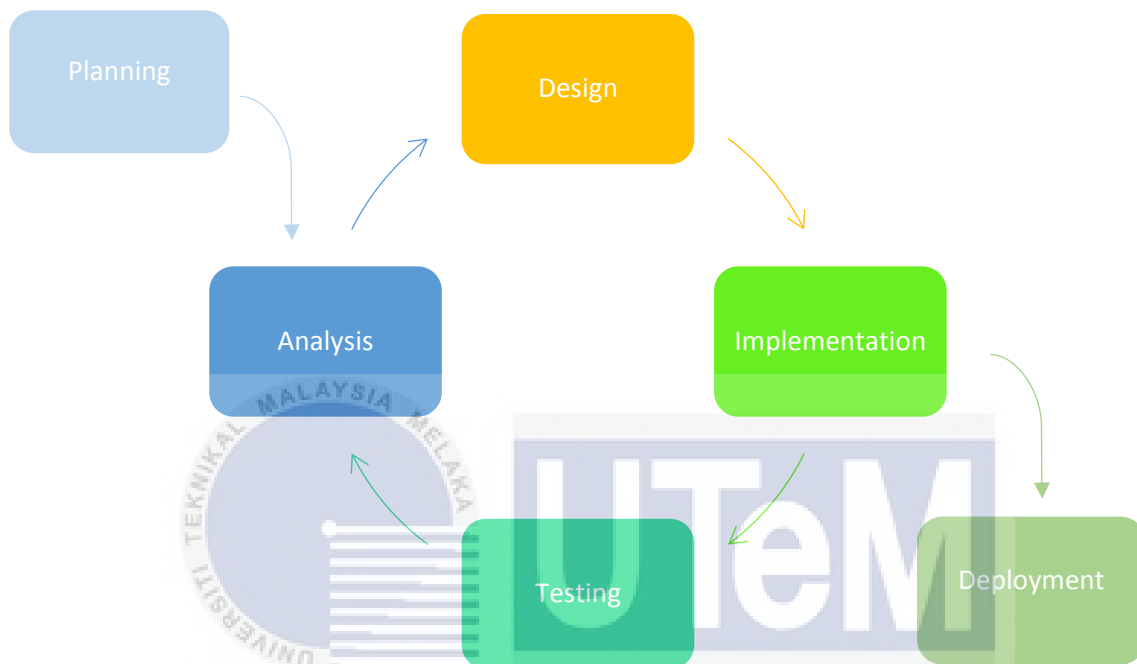
On the other hand, Bluetooth and RFID main disadvantage is to create a Find Me Friends: GPS Tracker using both techniques require extra sensors to detect the user's location. Bluetooth, for example, require several Bluetooth receivers to be strategically placed in a certain area to track the position of any discoverable devices. The Bluetooth receiver will search for any discoverable devices, record and send data back to the main system. However, the use of sensors for these techniques to work will limit the area that the application can work.

### 2.3 Project Methodology

Leo R. Vijayarathy et al. states that “software development methodology provides a framework for planning, executing, and managing the process of developing software system”. Therefore, choosing the right methodology for a software development project is very crucial to the project management and the success of the project. The software development methodology provides guideline on the execution of the project from start to finish.

This project will be developed using Rapid Application methodology. There are many ways in which the methodology is applied depending on the requirement and constraint of the project itself. For this project, the phases that will be used is planning, analysis, design, implementation, testing and deployment. This methodology is chosen for this project because of the project is a small-scale project and the short time constraint is the main limiting factor

The phases applied for the development of the project is illustrated in Figure 2.10. Further details of the activity of the phases will be explained in Chapter 5. Section 2.3.1 to 2.3.6 will explain the general idea of each phase.



**Figure 2.10: Rapid Application Methodology**


### 2.3.1 Planning

The planning of this project starts with drafting the outline for the proposal of the project. Then choose a topic, determine the problem statements and objectives of this project. Identify the target user, define the scope of the project and the platform to be used. Divide the application into modules for implementation phase. Submit proposal to supervisor and wait for approval. After that, proceed to completion of Chapter 1 and do literature review for the topic of the project. Also, describe the selected methodology for the development of the project and identify the project requirements which is the software requirement, hardware requirement and other requirement of this project.

### 2.3.2 Analysis

The analysis involves describing the current situation related to the project and show how the current system works in form of diagram. The problem statements identified in Chapter 1 will be explained. Then, do the requirement analysis. The requirement analysis involves data requirement, functional requirement and non-functional requirement. For data requirement, identify the system input, output, and the data to store in the database. For the functional requirement, specify the function of the system or what the system will do. For the non-functional requirement, define how the system will perform its functions. Other requirements of this project will also be analysed by justifying the usage of the software and hardware that will be used in the project. This phase will be iterated for at least four times before the deployment of the product.

### 2.3.4 Design



One of the first activity for the design phase is to describe the high-level design of the system structure. For the high-level design, define the architecture of the system and describe the static and dynamic view of the system. Then, design the user interfaces including the navigation design, the input design, and the output design. For the database design, Entity Relationship Diagram (ERD) is designed along with the business rules. The data dictionary for the database to be created is also included in the database design. Finally, describe the low-level design of the system, that is the software design and the physical database design. The design phase is repeated for each module of the application.

### 2.3.5 Implementation

The implementation phase is the development phase of the Find Me Friends: GPS Tracker. First, set up the development environment for this application. At the same time, plan how to do software configuration management. Next, start developing the application, module by module. The first module is user authentication followed by friend authentication. Then user detection module and lastly the distance calculation. At the same time, keep track of the implementation status, that is the development status for each module. The application will be built module by module after the analysis and design of each module; thus, the phase will be repeated four times.

### 2.3.6 Testing



The testing phase is very crucial for this application to work. Therefore, testing will start halfway through the implementation project. First, develop the test plan, identify who will be involved in the testing process. Then, describe the test environment, plan the test schedule, and identify the test strategy to be used to test the Find Me Friends: GPS Tracker. When all of that is done, design the test by identifying the test cases and expected result for each module. After that, determine the test data for the testing. Test is done on each module. Then collect the test results and analyse the test results. Finally, summarized the testing process. Testing is done for each completed module before deployment of the final product. Therefore, the phase is repeated four times during the duration of the development of the project.



### 2.3.7 Deployment

In the final phase of the project, the application will be prepared for deployment. The application is stored into deployment package before being deployed as a final product. The application will be installed in the two mobile devices because the application requires at least two devices for it to demonstrate some functionality that the application is intended to do. For the deployment, all the documentation regarding the project will be completed and reviewed before final submission. The documentation is important for the traceability and maintainability of the system for future improvement or research.

## 2.4 Project Requirements

### 2.4.1 Software Requirement



The software requirement for this project is listed as follows:

1. Windows 8.1 (64-bit)
2. Android Studio version 2.3
3. StarUML version 2.8.0
4. Adobe Photoshop CS6
5. Mozilla Firefox (32-bit) version 52.0.1
6. FileZilla version 3.25.1
7. Notepad++ (32-bit) version 7.3.3
8. Microsoft Word 2016
9. Microsoft Excel 2016
10. MySQL

## 2.4.2 Hardware Requirement

The hardware requirement for this project is listed as follows:

1. Laptop
2. Android smartphone

## 2.5 Project Schedule and Milestones

The following table shows the detail tasks in each phase of the project, including the duration, start and end date of each activity. The total duration of each phase is calculated from the start to finish date, including the weekends.

**Table 2.2: High-level project schedule**

Activity	Duration	Start	Finish
Planning	7d	13/2/2017	17/2/2017
Topic discussion	1d	13/2/2017	13/2/2017
Proposal discussion	1d	13/2/2017	13/2/2017
Proposal submission	1d	17/2/2017	17/2/2017
Analysis I	4d	18/2/2017	21/2/2017
Problem identification	1d	18/2/2017	18/2/2017
Requirement gathering	1d	19/2/2017	19/2/2017
Requirement analysis	2d	20/2/2017	21/2/2017
Design I	4d	22/2/2017	24/2/2017
High-level design	2d	22/2/2017	23/2/2017
Low-level design	2d	23/2/2017	24/2/2017
Implementation I	8d	25/2/2017	4/3/2017

Setting up software development environment	1d	25/2/2017	25/2/2017
Modules development	7d	26/2/2017	4/3/2017
Testing I	1d	5/3/2017	5/3/2017
Module testing	1d	5/3/2017	5/3/2017
Analysis II	4d	6/3/2017	9/3/2017
Problem identification	1d	6/3/2017	6/3/2017
Requirement gathering	1d	7/3/2017	7/3/2017
Requirement analysis	2d	8/3/2017	9/3/2017
Design II	4d	10/3/2017	13/3/2017
High-level design	2d	10/3/2017	11/3/2017
Low-level design	2d	12/3/2017	13/3/2017
Implementation II	7d	14/3/2017	20/3/2017
Modules development	7d	14/3/2017	20/3/2017
Testing II	1d	15/3/2017	15/3/2017
Module testing	1d	15/3/2017	15/3/2017
Analysis III	4d	16/3/2017	19/3/2017
Problem identification	1d	16/3/2017	16/3/2017
Requirement gathering	1d	17/3/2017	17/3/2017
Requirement analysis	2d	18/3/2017	19/3/2017
Design III	4d	20/3/2017	23/3/2017
High-level design	2d	20/3/2017	21/3/2017
Low-level design	2d	22/3/2017	23/3/2017
Implementation III	7d	24/3/2017	30/3/2017
Modules development	7d	24/3/2017	30/3/2017
Testing III	1d	31/3/2017	31/3/2017
Module testing	1d	31/3/2017	31/3/2017
Analysis IV	4d	1/4/2017	4/4/2017
Problem identification	1d	1/4/2017	1/4/2017
Requirement gathering	1d	2/4/2017	2/4/2017
Requirement analysis	2d	3/4/2017	4/4/2017
Design IV	4d	5/4/2017	8/4/2017

High-level design	2d	5/4/2017	6/4/2017
Low-level design	2d	7/4/2017	8/4/2017
Implementation IV	14d	9/4/2017	23/4/2017
Modules development	7d	9/4/2017	15/4/2017
Modules integration	7d	16/4/2017	23/4/2017
Testing IV	3d	24/4/2017	26/4/2017
Application testing	3d	24/4/2017	26/4/2017
Deployment	27d	27/4/2017	24/5/2017
Compile final report	14d	27/4/2017	11/5/2017
Ready the application for deployment	1d	23/5/2017	23/5/2017
Application deployment	1d	24/5/2017	24/5/2017

The following table shows the milestone of this project and the expected documents for the respective milestones.

**Table 2.3: Project milestone**

Milestone	Expected Documents	Date
Project planning	1. Project proposal	6 January 2017
	2. Project milestone	23 February 2017
Problem analysis	1. Chapter 1	2 March 2017
	2. Chapter 2	
Conceptual design	1. Entity Relationship Diagram	16 March 2017
	2. Chapter 3	
	3. Chapter 4	
Project implementation	1. Interface demonstration	9 March 2017
	2. Chapter 5	11 May 2017

## 2.6 Conclusion

As a conclusion, tracking technology is a very useful technology in this era of technology. Based on the observation and testing of existing systems, this project is going to focus using the Global Positioning System on smartphone to track user location. The selected methodology for this project is Rapid Application Development Model methodology. The project will hopefully be able to run smoothly in accordance with the planned schedule and milestones.



## CHAPTER III

### ANALYSIS



#### 3.1 Introduction

This chapter is going to elaborate on the problem analysis and requirement analysis for this project to analyse the proposed system. In problem analysis section, how current system works and the problem statements in Chapter 1 will be further analyse. The requirement analysis is going to discuss on the data requirement, functional requirement, non-functional requirement and the others requirement for this Find Me Friends: GPS Tracker. The others requirement is going to focus on the justification of the usage of the software, hardware and other resource used for this project.

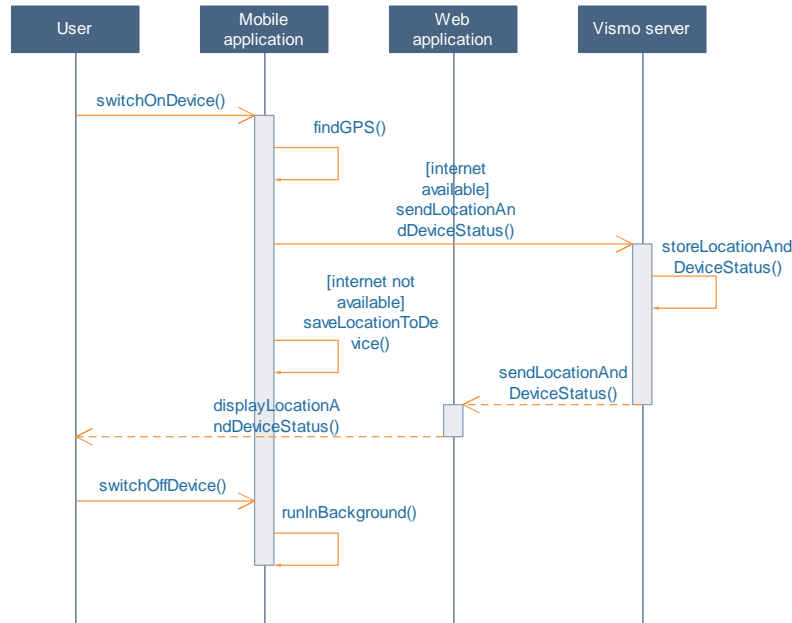
## 3.2 Problem Analysis

### 3.2.1 How Current System Works

There are many applications available today to track a person via their smartphone. Vismo is one of such application that uses GPS on the smartphone to track a person. It is developed and patented by Cellhire plc, a UK-based global service provider. The application, however, is not limited for smartphone use only. It can also be used on tablet, personal GPS tracker, and even satellite phone. Vismo also utilises other technologies besides GPS to provide the best GPS tracking updates even indoors. It works on most commonly used networks nowadays; GSM, 3G, WCDMA and CDMA. The key functions of this application are employee tracking, crisis management, panic button, privacy mode, geo-fencing, and multiplatform supported.

Vismo will immediately starts when the device is turned on and it will run as background process when the device is locked. It will retrieve the device location regularly and send them back over the internet. If the device is out of cell-tower coverage, the location will temporarily be stored into the device memory, and will be uploaded once back in coverage. Despite the regular update, Vismo will not impact the device battery life. A secure website-end of the application will display the real-time locations of all the mobile tracking devices. Other than the device location, Vismo will also display signal strength, battery level, charge indicator, and coverage levels.

Vismo involves the interaction between user, mobile application, web application and Vismo server. The following sequence diagram illustrate how Vismo works:



**Figure 3.1: How Vismo application works**

### 3.2.2 Analysis of Problem Statements

i. **Problem statement 1: Locating a person in a crowded place is difficult.**

The task of finding a person in a place full of people is like searching for a needle in a haystack. It will be even more challenging if the person one is looking for is small children as they can get easily lost in between people that are bigger than them. It is equally challenging to find a person who looks almost alike to people in the crowd in terms of height or any article of clothing such as hijabs or t-shirts. These are some factors that may cause looking for a person in crowded place difficult.

ii. **Problem statement 2: Locating a person in a massive expanse takes too much time and energy.**

The difficulty of searching for a person in a large area often costs a lot of time and energy. This is due to the need to increase the possibility of finding the person one is looking for. Hence, one tends to search in all possible places where the person might be found. However, that effort takes a lot of time and energy. This can lead to increase tiredness, agitation and level of stress.



iii. Problem statement 3: Exposing the exact location of a person can be very dangerous.

To find the person one is looking for, the other party needs to reveal their position to the searching party. While the data is useful to the person who is looking for the missing party, it can also be very dangerous if the information falls into the wrong hands. Location data is one of the most sensitive data to anyone. Many things can be exploited from this data alone. Therefore, exposing the location of a person has to be done very carefully.

### 3.3 Requirement Analysis

#### 3.3.1 Data Requirement



One of the result of the problem analysis in the previous section is the data requirement of the to-be application. The data requirement of this project is shown in the data dictionary as follows.

**Table 3.1: User table**

Attribute	Data Type	Null?	Auto-increment?	Primary key?	Foreign key?	Description
ID	VARCHAR	No	Yes	Yes	No	
Name	VARCHAR	No	No	No	No	
Phone number	INTEGER	No	No	No	No	
Email	VARCHAR	No	No	No	No	
Password	VARCHAR	No	No	No	No	

Table 3.2: Friend table

Attribute	Data Type	Null?	Auto-increment?	Primary key?	Foreign key?	Description
Request ID	VARCHAR	No	No	Yes	No	Use User ID from User Table. Refer to the person who makes the request.
Friend ID	VARCHAR	No	No	No	Yes	Use User ID from User Table. Refer to the person who is requested.
Request status	TINYINT	No	No	No	No	0 = Pending 1 = Accept

Table 3.3: Location table

Attribute	Data Type	Null?	Auto-increment?	Primary key?	Foreign key?	Description
Location ID	VARCHAR	No	No	Yes	No	Location of the user
Latitude	VARCHAR	No	No	No	No	Latitude of the user location
Longitude	VARCHAR	No	No	No	No	Longitude of user location

ID	VARCHAR	No	No	No	Yes	User ID from User Table
----	---------	----	----	----	-----	-------------------------------

### 3.3.2 Functional Requirement

The problem analysis in section 3.2 also produce functional requirement for the project. The functional requirement for this project is as follows:

- i. User shall have to register before using the system.
- ii. User shall be able to login to the system
- iii. User shall be able to add friends to their contact list.
- iv. User shall be able to see the friend's location.
- v. User shall be able to see the straight-line distance between the user's current location and their friend's.

### 3.3.3 Non-Functional Requirement

The non-functional requirement for this project is as follows:

- i. Usability  
Users of this application shall be able to understand how to use the application without detailed instruction.
- ii. Authentication  
Users shall be able to accept or reject location request from their friends.
- iii. Calculation time  
Users shall be able to see the distance between themselves and their friends in short time.

### 3.3.4 Others Requirement

The software, hardware and other requirements that will be used in this project is listed in the following tables.

**Table 3.4: List of software and its justification of usage**

Name	Version	Justification of Usage
Windows 8.1 (64-bit)	Single Language Edition	The default operating system of the laptop used for the project development.
Android Studio	2.3	The chosen IDE to develop the Find Me Friends: GPS Tracker.
StarUML	2.8.0	Software modelling tool to create UML diagrams related to this project.
Adobe Photoshop CS6	Creative Suite 6	Photo editor tool to edit, rescale, resize, and recompress images to be used inside the Find Me Friends: GPS Tracker.
Mozilla Firefox (32-bit)	52.0.1	Browser used to search for resources, such as article journal and online article, related to the project and other information needed for the project.
FileZilla	3.25.1	FTP client used to upload PHP files to the server.

Notepad++ (32-bit)	7.3.3	Notepad++ is used as source code editor for PHP files.
Microsoft Word 2016	2016 Edition	MS Word is used to type the proposal, reports and other documents for the project.
Microsoft Excel 2016	2016 Edition	MS Excel is used to create Gantt chart for this project.
MySQL		The selected relational database management system to store data on the server.
SQLite		An in-process library used to store data locally on the Android smartphone as a backup before being uploaded to the server.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**Table 3.5: List of hardware and its justification of usage**

Name	Details	Justification of Usage
Laptop	<ul style="list-style-type: none"> <li>- OS: Windows 64-bit</li> <li>- Processor: Intel® Core™ i5-4210U CPU @ 1.70 GHz 2.40 GHz</li> <li>- RAM: 8.00 GB</li> <li>- GPU: NVidia GeForce 830M</li> </ul>	The laptop used for the development of this project.
Android Smartphone	<ul style="list-style-type: none"> <li>- Android version: 5.1.1 (API 22)</li> </ul>	Find Me Friends: GPS Tracker is a native

	<ul style="list-style-type: none"> <li>- CPU: Quad-core Max 1.2GHz</li> <li>- RAM: 2.00GB</li> </ul>	<p>Android application. Therefore, the development requires an Android smartphone.</p>
--	--	--

### 3.4 Conclusion

To conclude, how current system works is sufficient to achieve the goals of this project. However, some improvisation can still be put in place. The problems stated in Chapter 1 is analysed in section 3.2.2. The data requirement of the project is shown in Table 3.1 and 3.2. These data are the input data that is going to be stored inside the database of this application. The project is going to focus on five functional requirements, three non-functional requirements and the justification of usage for each software and hardware used for this project is defined in detail in section 3.3.4.

## CHAPTER IV

### DESIGN



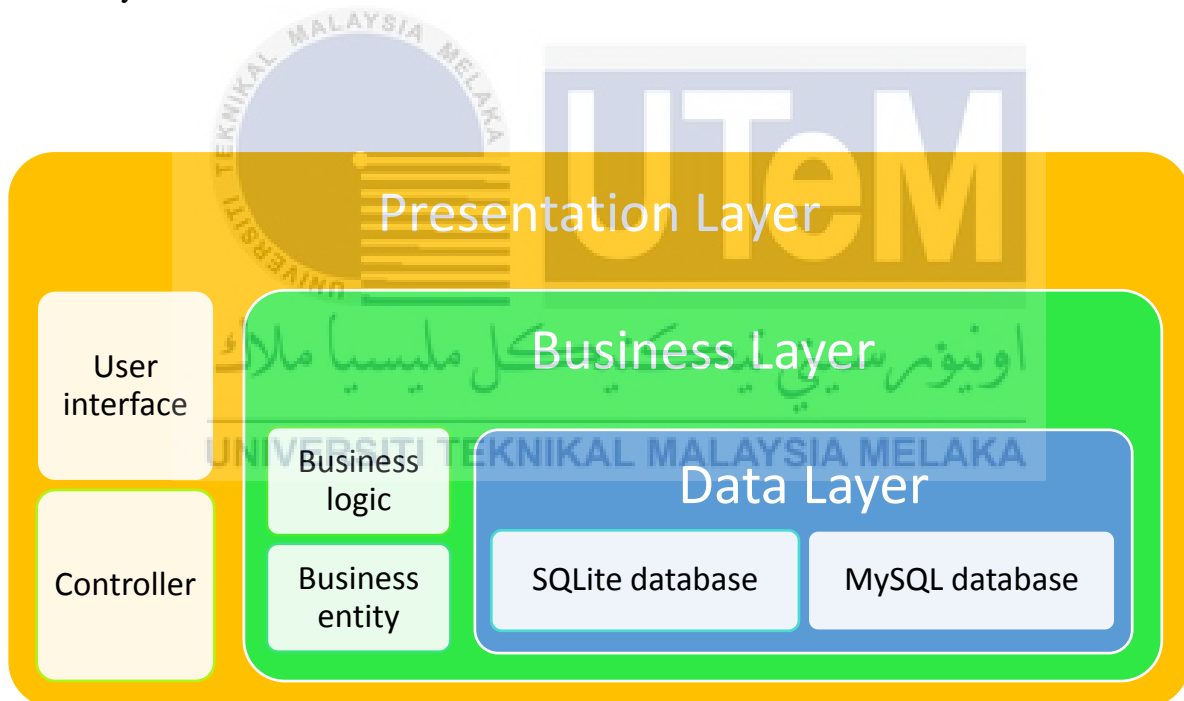
#### 4.1 Introduction

This chapter is going to explain about the high-level design and detailed design of Find Me Friends: GPS Tracker. The high-level design will elaborate more on the system architecture, user interface design, and the database design. The user interface design will consist of the navigation design, the input design and the output design. On the other hand, the detailed design is going to explain on the software design and the physical database design.

## 4.2 High-Level Design

The Find Me Friends: GPS Tracker will provide a medium to locate a friend using the application installed on the friend's device. The Find Me Friends: GPS Tracker will be built on Android platform. The application relies on the GPS functionality on the user's device to track the user's location. The user's location will be stored both locally, on the user's device, and on web server. The user's location will be sent to the server per user's request. To ensure only the authorized person can track an individual, the user must be on each other's friend list.

### 4.2.1 System Architecture

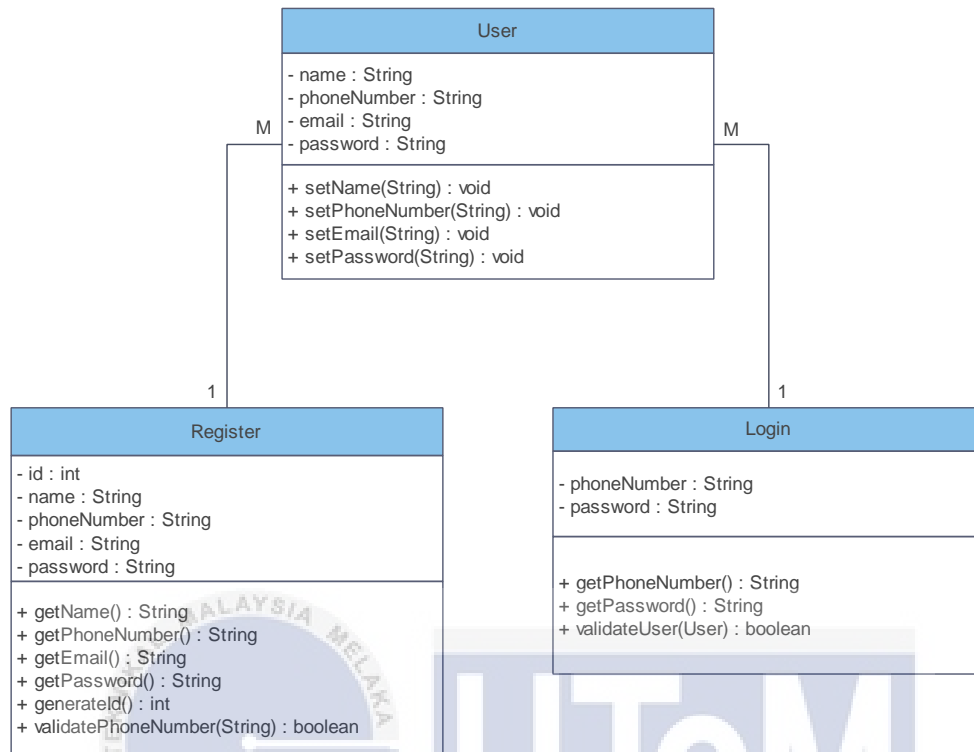


**Figure 4.1: Find Me Friends: GPS Tracker architecture**

The Find Me Friends: GPS Tracker is built upon a three-tier architecture. The first layer is the presentation layer. This layer will deal with the user's request and display the result of any business process. The second layer, the business layer, will process the user's request and send them to the data layer to be stored. The third layer, the data layer is where the all data will be stored. Figure 4.1 will illustrate the application's architecture.

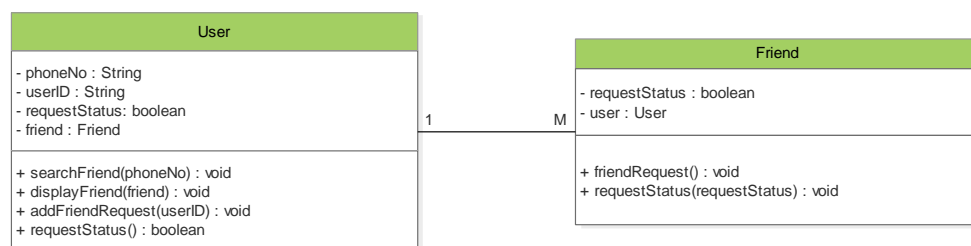


The static view of the application will be illustrated in Figure 4.2 through Figure 4.5 using class diagram.



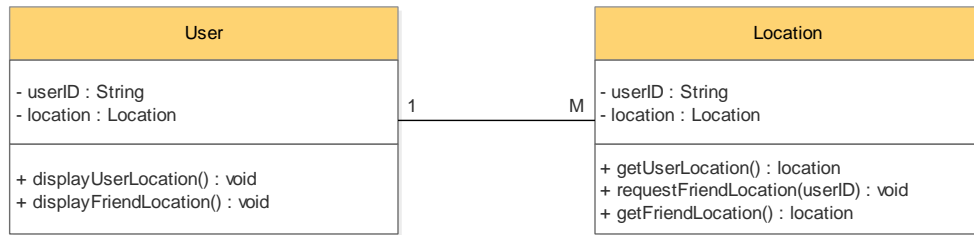
**Figure 4.2: Class diagram for user authentication**

Figure 4.2 shows that there are three classes that will interact in user authentication module. Login class and Register class inherit some of the attributes from User class.



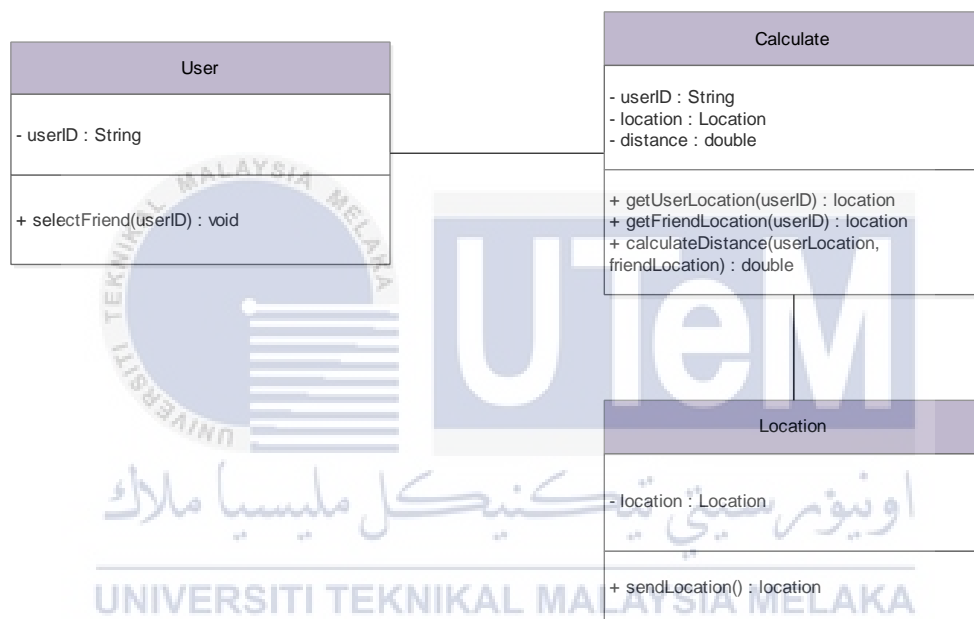
**Figure 4.3: Class diagram for friend authentication**

The above figure shows there are two classes that will interact in friend authentication module. User class's attribute is important to be used by Friend class to authenticate the relationship between two users.



**Figure 4.4: Class diagram for user detection**

In user detection module, the classes involved are User and Location class (Figure 4.4). The User class is mostly used to display the users' location on the map.



**Figure 4.5: Class diagram for distance calculation**

Distance calculation module used three classes as shown in Figure 4.5. User class and location attribute are important for calculate class to compute the distance between the users.

The dynamic view of the Find Me Friends: GPS Tracker will be illustrated in Figure 4.6 to Figure 4.9.

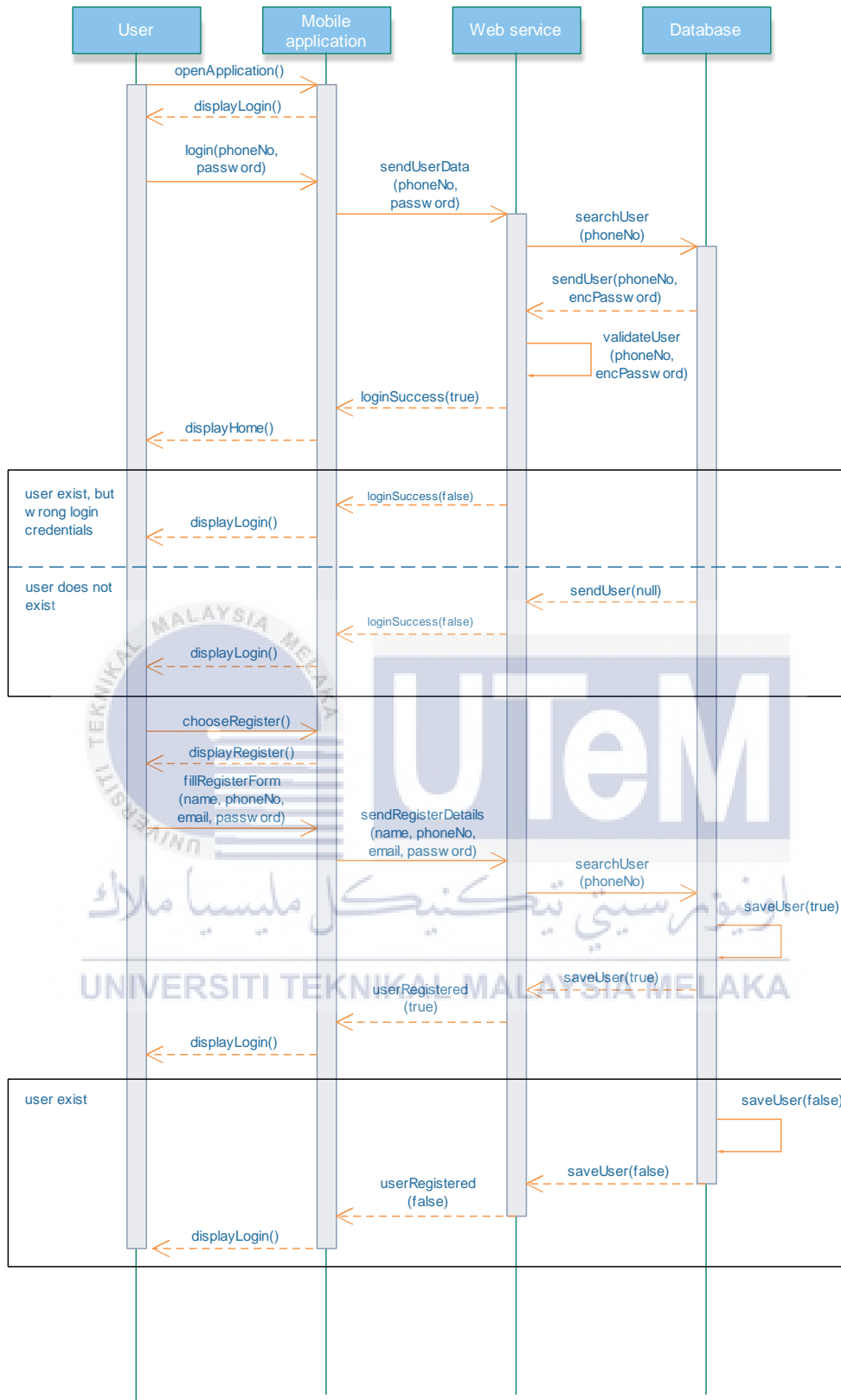
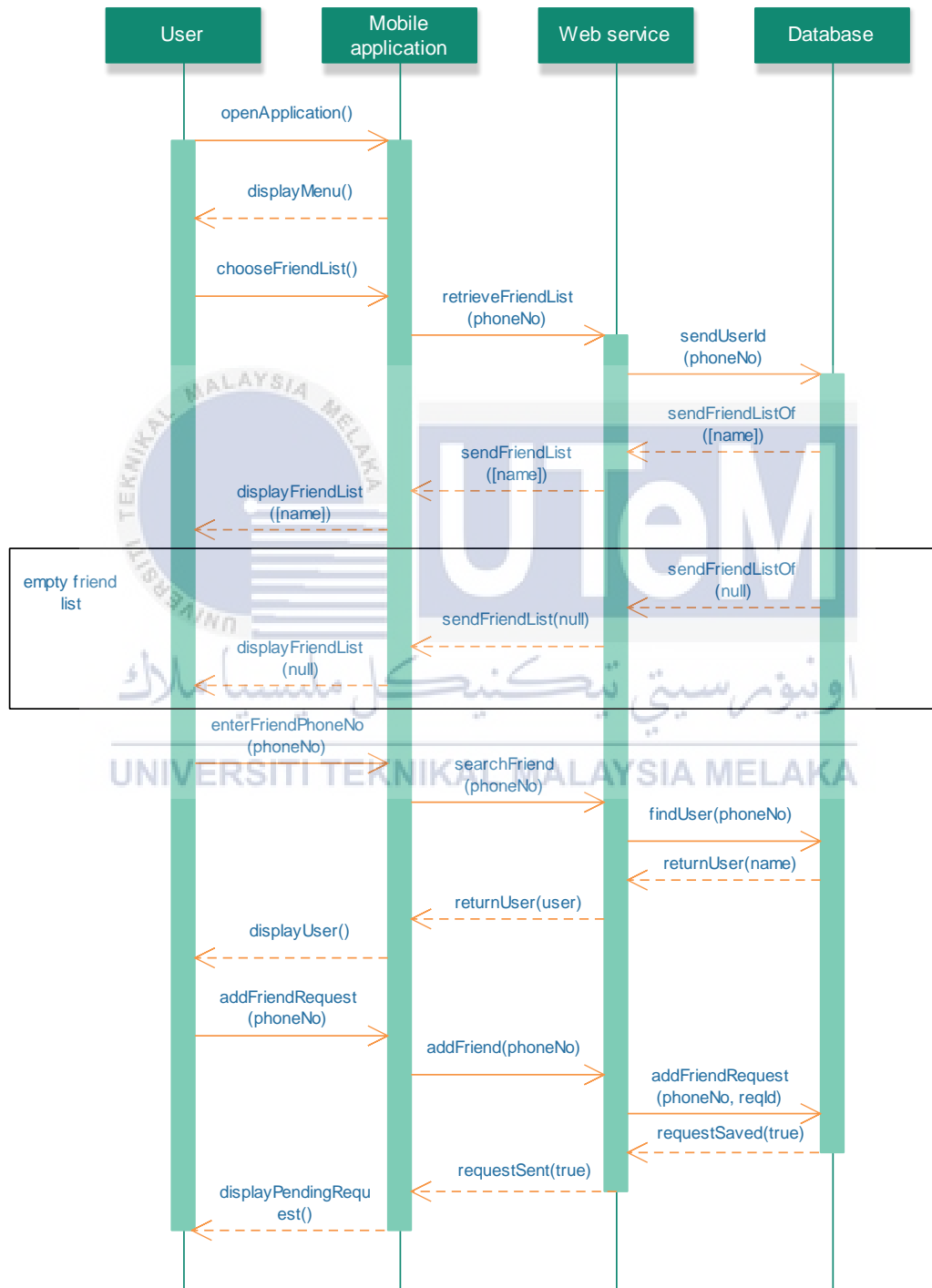


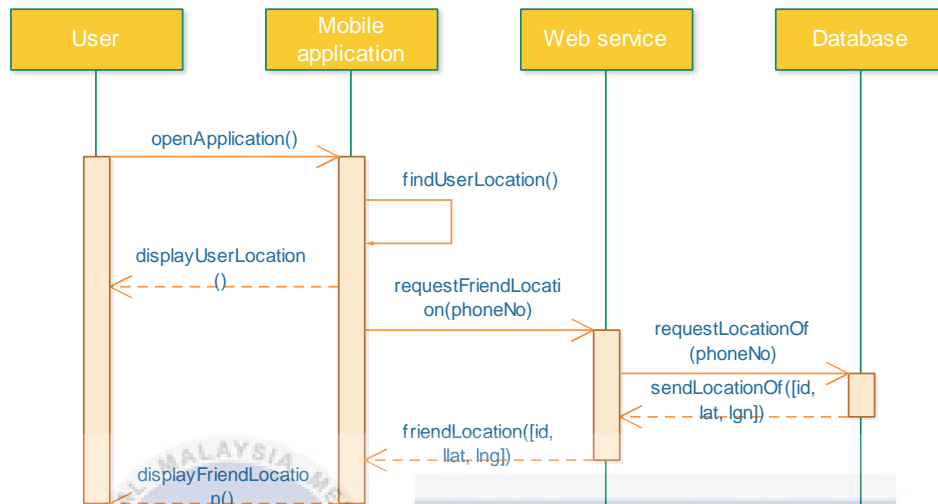
Figure 4.6: Sequence diagram for user authentication

Figure 4.6 shows five conditions that might occur in this module. The first flow shows successful login. The second flow shows the condition where the user enter either wrong phone number or password. The third flow shows when user enter unregistered account phone number. The fourth flow demonstrates successful registration and the last flow is when user is trying to register a registered phone number.



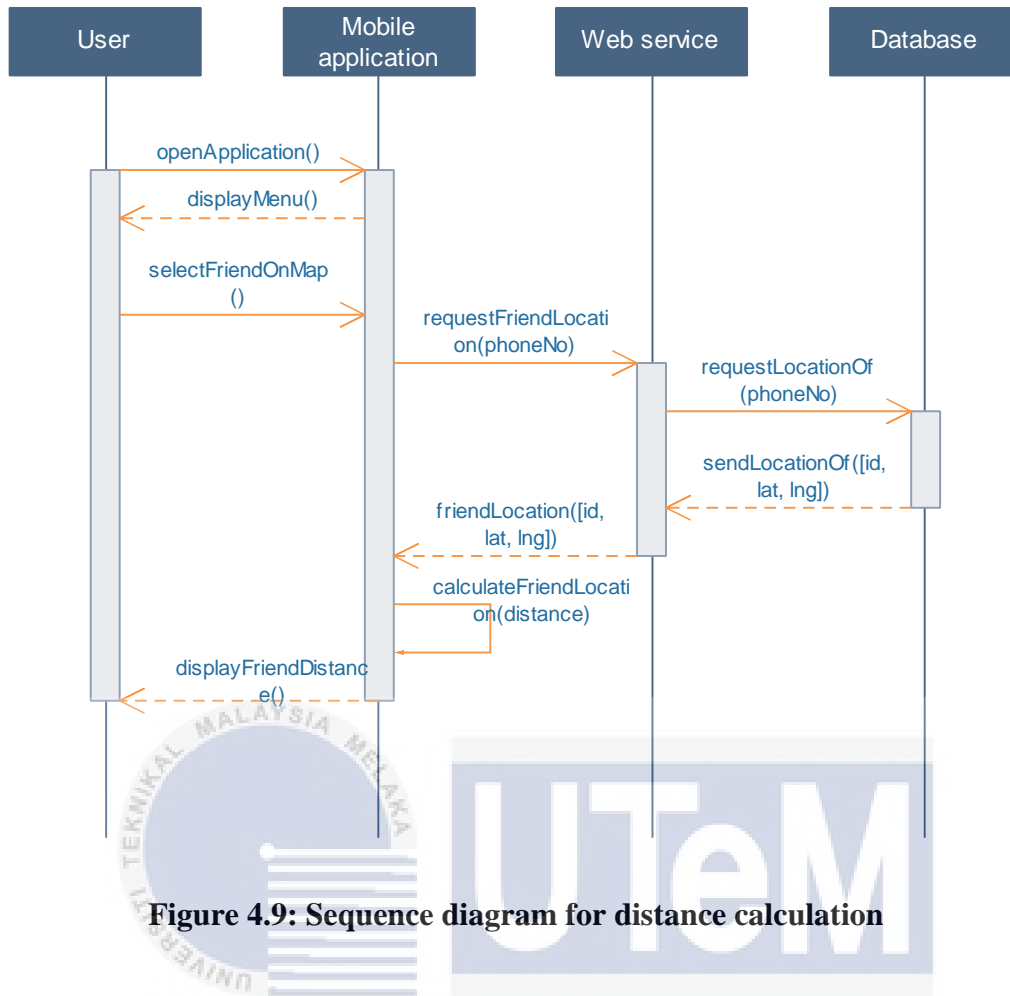
**Figure 4.7: Sequence diagram for friend authentication**

Figure 4.7 shows the three conditions that are possible for friend authentication module. The first condition shows successful retrieval of friend list. The second condition shows what will happen when the friend list is empty. The last condition is when user send friend request to another user.



**Figure 4.8: Sequence diagram for user detection**

Figure 4.8 shows how the objects in the application will interact with each other during location retrieval. This user detection module will display all users' location on the map. These users must first be friends with each other before they are able to see another user location.



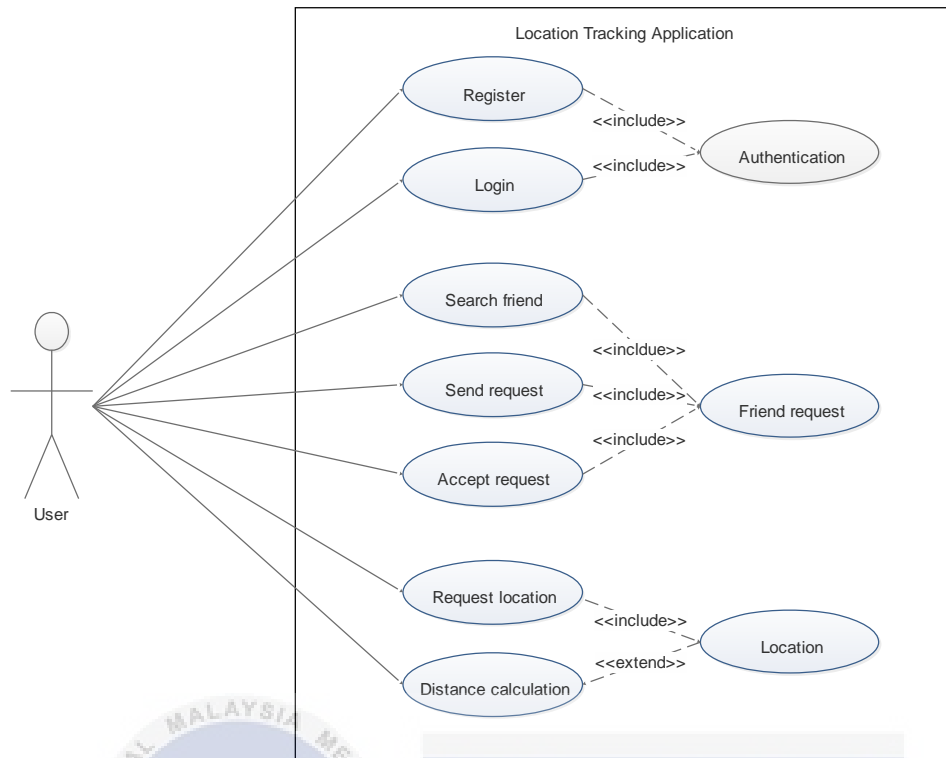
**Figure 4.9: Sequence diagram for distance calculation**

Figure 4.9 demonstrate how the distance between user will be calculated in the distance calculation module. The friend must first be selected before the distance between the user are computed.

## 4.2.2 User Interface Design

### 4.2.2.1 Navigation Design

The navigation flow of the system is illustrated in Figure 4.10. The figure shows how the user will interact with the system.



**Figure 4.10: Use case for Find Me Friends: GPS Tracker**

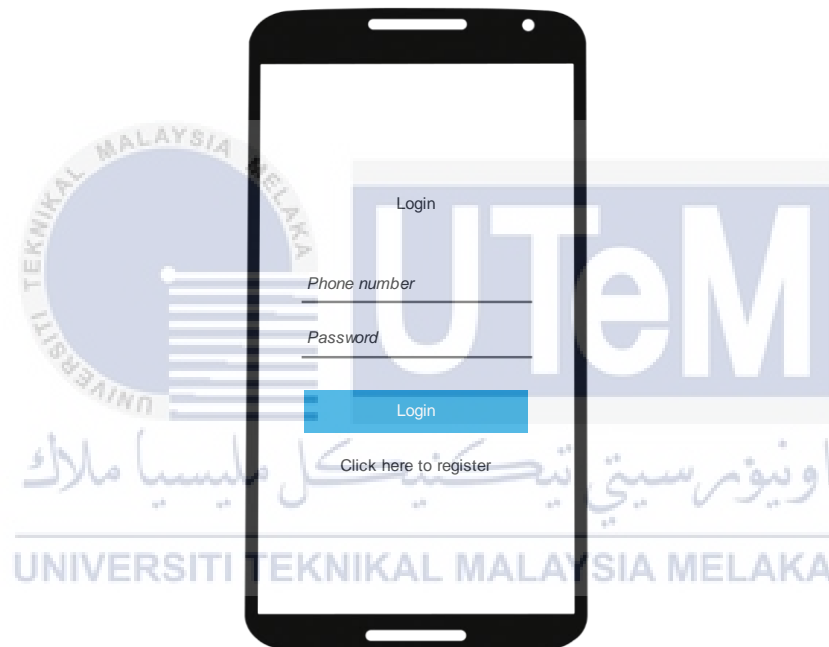
The type of navigation control that the application will implement are linear and hierarchical. The linear navigation control will only enable the user to explore through the interface in a one-way manner. This navigation control will be used in the search friend and sending and accepting request. The hierarchical navigation control is where there is only one path leading to each screen and every screen has a parent screen. The navigation control will be used the register and login screen.

#### 4.2.2.2 Input Design

Find Me Friends: GPS Tracker has a few screens where the user will have to enter their input for processing. The screens are the login screen, register screen and search screen. The type of input for this application are text, number, and password. There are two forms on which user is required to enter their information, that is login form and register form. There is also a search box to search for friend where the user is required to enter their friend's phone number.

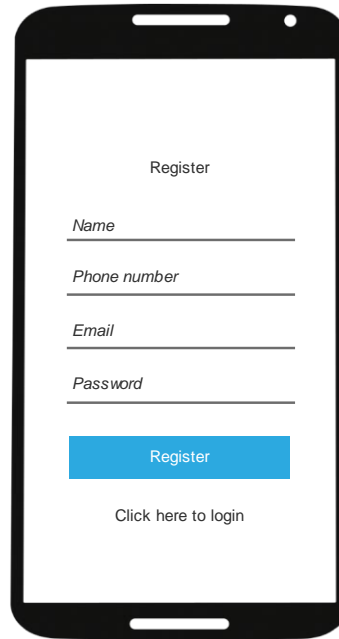
The validation rule that apply to the login form is the phone number entered must already registered to a user. The phone number and password also must be matching before the user can login. The validation rule that apply to the login form is the phone number entered must not belong to any existing user. The form also must be completed in for the user to be registered. The validation rule that apply to the search friend is the phone number must exist in the database or the result will return empty.

There are three screens that will require user input. The screens are login screen (Figure 4.11), registration screen (Figure 4.12) and search friend screen (Figure 4.13).



**Figure 4.11: Login screen**





Register

Name \_\_\_\_\_

Phone number \_\_\_\_\_

Email \_\_\_\_\_

Password \_\_\_\_\_

Register

[Click here to login](#)

Figure 4.12: Register screen



Friends

Search friend

Search

Figure 4.13: Search friend screen

### 4.2.2.3 Output Design

The type of output of the Find Me Friends: GPS Tracker is mainly the markers to show the user location on map, and list view to show friend list. The result of distance calculation will be displayed in a dialog box. The user location screen is updated every 30 seconds. While the list view and the distance calculation result are displayed on ad-hoc basis where it will only be displayed when prompted by user.

There are three main screens that will display the output to the user. The screens are user location screen (Figure 4.14), friend list screen (Figure 4.15) and distance calculation screen (Figure 4.16).



**Figure 4.14: User location screen**

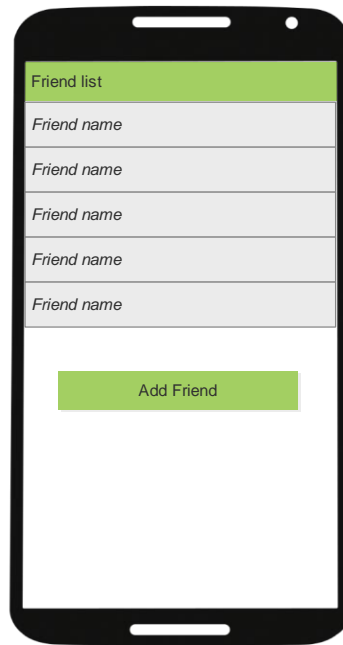


Figure 4.15: Friend list screen

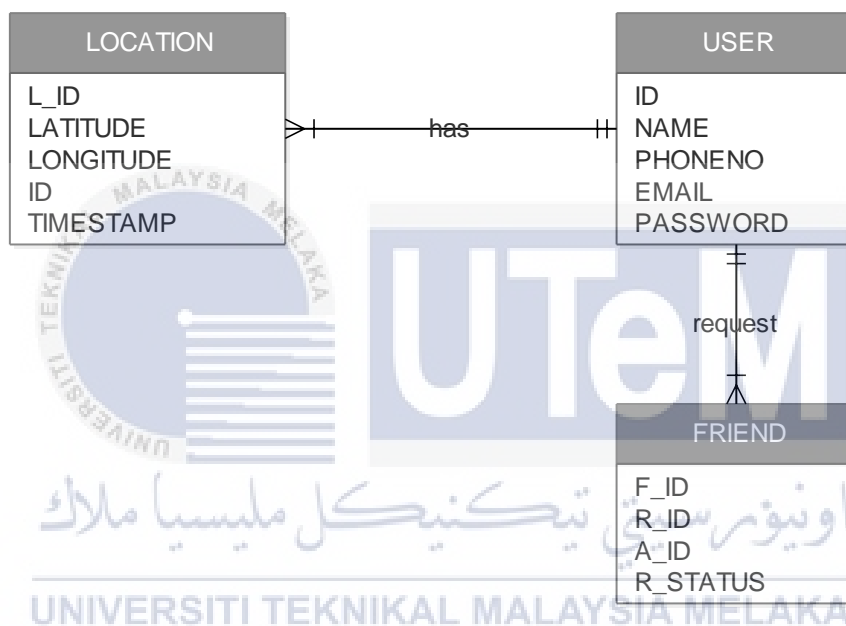


Figure 4.16: Distance calculation result

### 4.2.3 Database Design

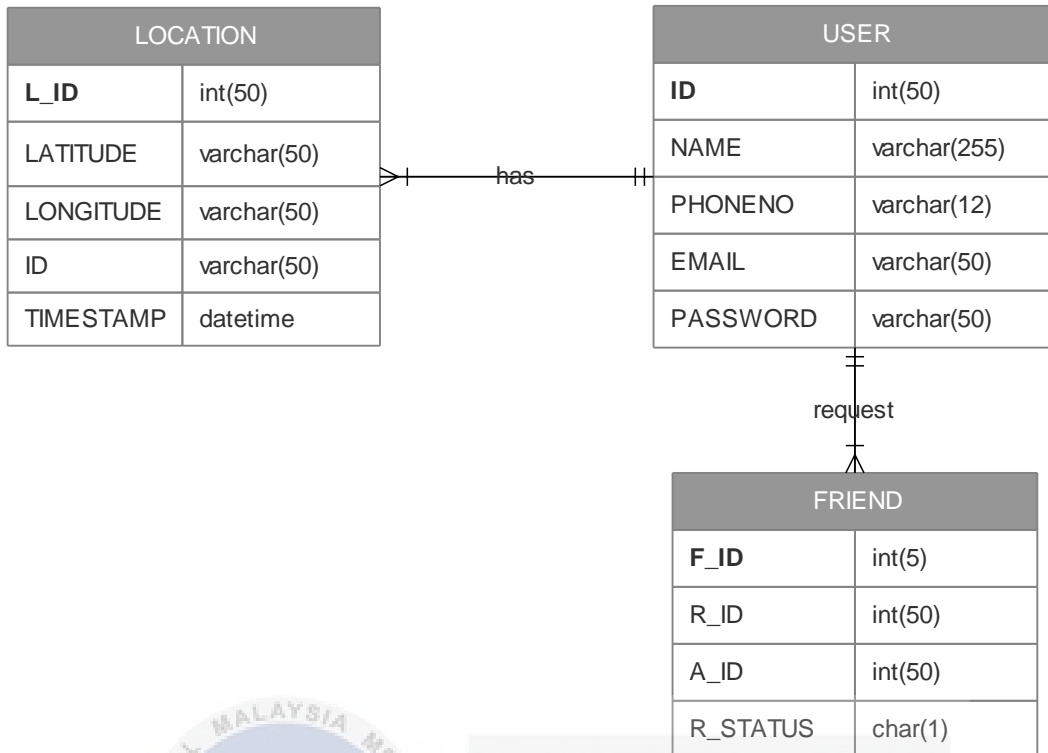
#### 4.2.3.1 Conceptual and Logical Database Design

The data requirement from section 3.3.1 is used to create a conceptual data model which is illustrated in Figure 4.17. The entity relationship diagram illustrated how different entities in the application will interact with each other.



**Figure 4.17: Conceptual database design**

A logical database design is created based on the conceptual database design previously. The logical database design is how the entity relationship diagram is translated into relation schema. In this phase, the key constraint and data type of each attribute of the entity is defined.



**Figure 4.18: Logical database design**

The relationship of the entity in both Figure 4.17 and Figure 4.18 is only illustrated by the relationship name and cardinality. Business rule will define some aspect of business into the relationship of the entities.

- i. Each user can send one or more friend request. Each friend request is submitted by only one user.
- ii. Each user has one or more location stored in latitude and longitude. One location belongs to one user (identified by user ID).

The details of the attribute in the entity relationship diagram is in Table 4.1 to Table 4.3.

**Table 4.1: User table**

TABLE NAME	ATTRIBUTE NAME	CONTENT	TYPE	FORMA T	SIZE	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
USER	ID	User ID	VARCHAR	###	50	N/A	YES	PK	
	NAME	Name	VARCHAR	Xxxxxx Xxxxx	255	N/A	YES		
	PHONENO	Phone number	VARCHAR	xxxxxxx xxxx	12	N/A	YES		
	EMAIL	Email	VARCHAR	xxxxxx @xxxx.c om	50	N/A	YES		
	PASSWORD	Password	VARCHAR	xxxxxxx xxx	50	N/A	YES		

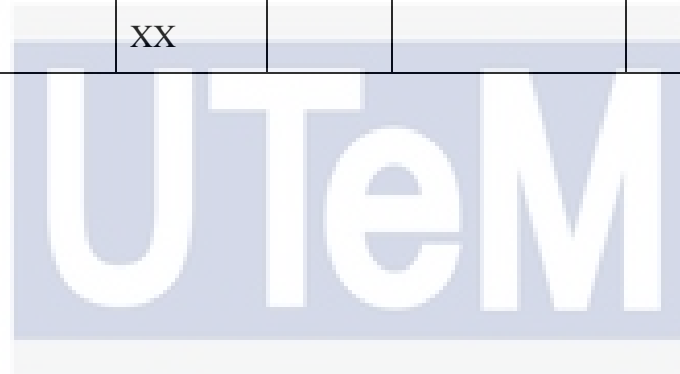
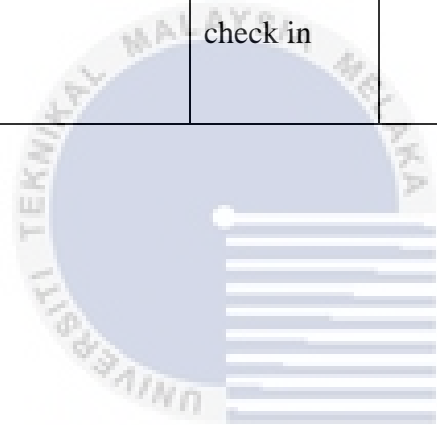
Table 4.2: Friend table

TABLE NAME	ATTRIBUTE NAME	CONTENT	TYPE	FORMAT	SIZE	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
<b>FRIEND</b>	F_ID	Request ID	INT	###	5	N/A	YES	PK	
	R_ID	Friend's ID	VARCHAR	###	50	N/A	YES	FK	USER
	A_ID	User ID	VARCHAR	###	50	N/A	YES	FK	USER
	R_STATUS	Request status	TINYINT	Xxxxxxx Xxxxxx	1	0/1	YES		

Table 4.3: Location table

TABLE NAME	ATTRIBUTE NAME	CONTENT	TYPE	FORMAT	SIZE	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
<b>LOCATION</b>	L_ID	Location ID	VARCHAR	xxxxxxx xxx	255	N/A	YES	PK	
	LATITUDE	Location latitude	VARCHAR	xxxxxxx xxx	50	N/A	YES		

	LONGITUDE	Location longitude	VARCHAR	xxxxxxx xxx	50	N/A	YES		
	ID	User ID	VARCHAR	###	50	N/A	YES	FK	USER
	TIMESTAMP	Timestamp of user check in	DATETIME	XXXX- XX-XX XX:XX: XX	N/A	N/A	YES		



اونیورسیتی تکنیکل ملیسیا ملاک

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



### 4.3 Detailed Design

#### 4.3.1 Software Design

This section will further elaborate on the class diagrams in Figure 4.2. to Figure 4.5. The elaboration will include the class responsibility, attributes, methods, input and output parameters, and pre-condition and post-condition of the class.

##### 4.3.1.1 User authentication

**Table 4.4: User class**

Class name	User
Responsibility	This class will store the user description for use of other class
Attributes	<ul style="list-style-type: none"> <li>i. name:String</li> <li>ii. phoneNumber:String</li> <li>iii. email:String</li> <li>iv. password:String</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. setName(String):void</li> <li>ii. setPhoneNumber(String):void</li> <li>iii. setEmail(String):void</li> <li>iv. setPassword(String):void</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. setName(String):void               <ul style="list-style-type: none"> <li>a. Input parameter: String name</li> <li>b. Output parameter: None</li> </ul> </li> <li>ii. setPhoneNumber(String):void               <ul style="list-style-type: none"> <li>a. Input parameter: String phoneNumber</li> <li>b. Output parameter: None</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>iii. setEmail(String):void <ul style="list-style-type: none"> <li>a. Input parameter: String email</li> <li>b. Output parameter: None</li> </ul> </li> <li>iv. setPassword(String):void <ul style="list-style-type: none"> <li>a. Input parameter: String password</li> <li>b. Output parameter: None</li> </ul> </li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. setName(String):void <ul style="list-style-type: none"> <li>a. Pre-condition: User enter user name</li> <li>b. Post-condition: Name is stored in variable 'name'</li> </ul> </li> <li>ii. setPhoneNumber(String):void <ul style="list-style-type: none"> <li>a. Pre-condition: User enter phone number</li> <li>b. Post-condition: Phone number is stored in variable 'phoneNumber'</li> </ul> </li> <li>iii. setEmail(String):void <ul style="list-style-type: none"> <li>a. Pre-condition: User enter email</li> <li>b. Post-condition: Email is stored in variable 'email'</li> </ul> </li> <li>iv. setPassword(String):void <ul style="list-style-type: none"> <li>a. Pre-condition: User enter password</li> <li>b. Post-condition: Password is stored in variable 'password'</li> </ul> </li> </ul>

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

**Table 4.5: Register class**

Class name	Register
Responsibility	This class will send the user data to the database for user registration
Attributes	<ul style="list-style-type: none"> <li>i. id:int</li> <li>ii. name:String</li> <li>iii. phoneNumber:String</li> <li>iv. email:String</li> <li>v. password:String</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. getName():String</li> <li>ii. getPhoneNumber():String</li> <li>iii. getEmail():String</li> </ul>

	<ul style="list-style-type: none"> <li>iv. getPassword():String</li> <li>v. generateId():int</li> <li>vi. validatePhoneNumber(String):boolean</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. getName():String <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return name</li> </ul> </li> <li>ii. getPhoneNumber():String <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return phone number</li> </ul> </li> <li>iii. getEmail():String <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return email</li> </ul> </li> <li>iv. getPassword():String <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return password</li> </ul> </li> <li>v. generateId():int <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return generated ID</li> </ul> </li> <li>vi. validatePhoneNumber(String):Boolean <ul style="list-style-type: none"> <li>a. Input parameter: User phone number in String</li> <li>b. Output parameter: Return true or false</li> </ul> </li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. getName():String <ul style="list-style-type: none"> <li>a. Pre-condition: Variable 'name' is not empty</li> <li>b. Post-condition: Name is sent to database</li> </ul> </li> <li>ii. getPhoneNumber():String <ul style="list-style-type: none"> <li>a. Pre-condition: Variable 'phoneNumber' is not empty</li> <li>b. Post-condition: Phone number is sent to database</li> </ul> </li> <li>iii. getEmail():String <ul style="list-style-type: none"> <li>a. Pre-condition: Variable 'email' is not empty</li> <li>b. Post-condition: Email is sent to database</li> </ul> </li> <li>iv. getPassword():String <ul style="list-style-type: none"> <li>a. Pre-condition: Variable 'password' is not empty</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>b. Post-condition: Password is sent to database</li> <li>v. generateId():int <ul style="list-style-type: none"> <li>a. Pre-condition: Request is made to generate user ID</li> <li>b. Post-condition: User ID is generated and sent to database</li> </ul> </li> <li>vi. validatePhoneNumber(String):Boolean <ul style="list-style-type: none"> <li>a. Pre-condition: 'phoneNumber' variable is not empty</li> <li>b. Post-condition: User is allowed or not to be registered</li> </ul> </li> </ul>
--	---

**Table 4.6: Login class**

Class name	Login
Responsibility	This class will retrieve data from database for user login
Attributes	<ul style="list-style-type: none"> <li>i. phoneNumber:String</li> <li>ii. password:String</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. getPhoneNumber():void</li> <li>ii. getPassword():void</li> <li>iii. validateUser():boolean</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. getPhoneNumber():void <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return phone number</li> </ul> </li> <li>ii. getPassword():void <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return password</li> </ul> </li> <li>iii. validateUser(User):Boolean <ul style="list-style-type: none"> <li>a. Input parameter: User Object</li> <li>b. Output parameter: Return true or false</li> </ul> </li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. getPhoneNumber():void <ul style="list-style-type: none"> <li>a. Pre-condition: Phone number in database is not empty</li> <li>b. Post-condition: Phone number from database is stored in 'phoneNumber'</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>ii. getPassword():void <ul style="list-style-type: none"> <li>a. Pre-condition: Password in database is not</li> <li>b. Post-condition: Password from database is stored in 'password'</li> </ul> </li> <li>iii. validateUser(User):Boolean <ul style="list-style-type: none"> <li>a. Pre-condition: 'User' object is not empty</li> <li>b. Post-condition: User is allowed or not to be login</li> </ul> </li> </ul>
--	--

#### 4.3.1.2 Friend authentication

**Table 4.7: User class**

Class name	User
Responsibility	This class will help display friend list,
Attributes	<ul style="list-style-type: none"> <li>i. phoneNumber:String</li> <li>ii. userID:String</li> <li>iii. requestStatus:Boolean</li> <li>iv. friend:Friend</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. searchFriend(phoneNumber):void</li> <li>ii. displayFriend(friend):void</li> <li>iii. addFriendRequest(userID):void</li> <li>iv. requestStatus():boolean</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. searchFriend(phoneNumber):void <ul style="list-style-type: none"> <li>a. Input parameter: String phoneNumber</li> <li>b. Output parameter: None</li> </ul> </li> <li>ii. displayFriend(friend):void <ul style="list-style-type: none"> <li>a. Input parameter: Friend friend</li> <li>b. Output parameter: None</li> </ul> </li> <li>iii. addFriendRequest(userID):void <ul style="list-style-type: none"> <li>a. Input parameter: String userID</li> <li>b. Output parameter: None</li> </ul> </li> <li>iv. requestStatus():Boolean</li> </ul>

	<ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Return true or false</li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. searchFriend(phoneNumber):void <ul style="list-style-type: none"> <li>a. Pre-condition: User enter friend's phone number</li> <li>b. Post-condition: Search the phone' number in database</li> </ul> </li> <li>ii. displayFriend(friend):void <ul style="list-style-type: none"> <li>a. Pre-condition: Phone number entered is available in database</li> <li>b. Post-condition: Friend is displayed</li> </ul> </li> <li>iii. addFriendRequest(userID):void <ul style="list-style-type: none"> <li>a. Pre-condition: User select the friend they want to add</li> <li>b. Post-condition: Friend request is made and sent to database</li> </ul> </li> <li>iv. requestStatus():Boolean <ul style="list-style-type: none"> <li>a. Pre-condition: User sent friend request</li> <li>b. Post-condition: Friend request is accepted or rejected</li> </ul> </li> </ul>

**Table 4.8: Friend class**

Class name	Friend
Responsibility	This class will handle friend request
Attributes	<ul style="list-style-type: none"> <li>i. requestStatus:Boolean</li> <li>ii. user:User</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. friendRequest():void</li> <li>ii. requestStatus(requestStatus):void</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. friendRequest():void <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: None</li> </ul> </li> <li>ii. requestStatus(requestStatus):void <ul style="list-style-type: none"> <li>a. Input parameter: Boolean requestStatus</li> <li>b. Output parameter: None</li> </ul> </li> </ul>

Pre/Post condition	<ul style="list-style-type: none"> <li>i. friendRequest():void <ul style="list-style-type: none"> <li>a. Pre-condition: There is friend request from another user</li> <li>b. Post-condition: Display friend request</li> </ul> </li> <li>ii. requestStatus(requestStatus):void <ul style="list-style-type: none"> <li>a. Pre-condition: Friend request status is false</li> <li>b. Post-condition: Friend request is rejected (false) or return true</li> </ul> </li> </ul>
--------------------	--

#### 4.3.1.3 User detection

**Table 4.9: User class**

Class name	User
Responsibility	This class will display user and friend location
Attributes	<ul style="list-style-type: none"> <li>i. userID:String</li> <li>ii. location:Location</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. displayUserLocation:void</li> <li>ii. displayFriendLocation:void</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. displayUserLocation():void <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: None</li> </ul> </li> <li>ii. displayFriendLocation():void <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: None</li> </ul> </li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. displayUserLocation():void <ul style="list-style-type: none"> <li>a. Pre-condition: User has checked in to a location</li> <li>b. Post-condition: User location is displayed on map</li> </ul> </li> <li>ii. displayFriendLocation():void <ul style="list-style-type: none"> <li>a. Pre-condition: Friend has checked in to a location</li> <li>b. Post-condition: Friend location is displayed on map</li> </ul> </li> </ul>

**Table 4.10: Location class**

Class name	Location
Responsibility	This class will get user location and retrieve friend's location
Attributes	<ul style="list-style-type: none"> <li>i. userID:String</li> <li>ii. location:Location</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. getUserLocation():location</li> <li>ii. requestFriendLocation(userID):void</li> <li>iii. getFriendLocation():location</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. getUserLocation():location <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Location location</li> </ul> </li> <li>ii. requestFriendLocation(userID):void <ul style="list-style-type: none"> <li>a. Input parameter: String userID</li> <li>b. Output parameter: None</li> </ul> </li> <li>iii. getFriendLocation():location <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Location location</li> </ul> </li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. getUserLocation():location <ul style="list-style-type: none"> <li>a. Pre-condition: User location is available in the database</li> <li>b. Post-condition: User location is stored into 'location' object</li> </ul> </li> <li>ii. requestFriendLocation(userID):void <ul style="list-style-type: none"> <li>a. Pre-condition: Friend request is approved in the database</li> <li>b. Post-condition: Friend location request is approved or rejected</li> </ul> </li> <li>iii. getFriendLocation():location <ul style="list-style-type: none"> <li>a. Pre-condition: Friend location is available in the database</li> <li>b. Post-condition: Friend location is stored into 'location' object</li> </ul> </li> </ul>



## 4.3.1.4 Distance calculation

**Table 4.11: User class**

Class name	User
Responsibility	This class will select the friend location to calculate distance
Attributes	userID:String
Methods	selectFriend(userID):void
Input/Output parameter	<ul style="list-style-type: none"> <li>i. selectFriend(userID):void <ul style="list-style-type: none"> <li>a. Input parameter: String userID</li> <li>b. Output parameter: None</li> </ul> </li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. selectFriend(userID):void <ul style="list-style-type: none"> <li>a. Pre-condition: User is available in the database</li> <li>b. Post-condition: Friend is selected</li> </ul> </li> </ul>

**Table 4.12: Calculate class**

Class name	Calculate
Responsibility	This class will calculate distance between two users
Attributes	<ul style="list-style-type: none"> <li>i. userID:String</li> <li>ii. location:Location</li> <li>iii. distance:double</li> </ul>
Methods	<ul style="list-style-type: none"> <li>i. getUserLocation(userID):location</li> <li>ii. getFriendLocation(userID):location</li> <li>iii. calculateDistance(userLocation, friendLocation):double</li> </ul>
Input/Output parameter	<ul style="list-style-type: none"> <li>i. getUserLocation(userID):location <ul style="list-style-type: none"> <li>a. Input parameter: String userID</li> <li>b. Output parameter: Location location</li> </ul> </li> <li>ii. getFriendLocation(userID):location <ul style="list-style-type: none"> <li>a. Input parameter: String userID</li> <li>b. Output parameter: Location location</li> </ul> </li> <li>iii. calculateDistance(userLocation, friendLocation):double <ul style="list-style-type: none"> <li>a. Input parameter: userLocation, friendLocation</li> <li>b. Output parameter: double distance</li> </ul> </li> </ul>

Pre/Post condition	<ul style="list-style-type: none"> <li>i. getUserLocation(userID):location <ul style="list-style-type: none"> <li>a. Pre-condition: User location is available in the database</li> <li>b. Post-condition: User location is stored into 'location' object</li> </ul> </li> <li>ii. getFriendLocation():location <ul style="list-style-type: none"> <li>a. Pre-condition: Friend location is available in the database</li> <li>b. Post-condition: Friend location is stored into 'location' object</li> </ul> </li> <li>iii. calculateDistance(userLocation, friendLocation):double <ul style="list-style-type: none"> <li>a. Pre-condition: Another user is selected and the selected user has approved friend request</li> <li>b. Post-condition: Distance between two users is calculated</li> </ul> </li> </ul>
--------------------	---

**Table 4.13: Location class**

Class name	Location
Responsibility	This class will store the user description for use of other class
Attributes	location:Location
Methods	sendLocation():location
Input/Output parameter	<ul style="list-style-type: none"> <li>i. sendLocation():location <ul style="list-style-type: none"> <li>a. Input parameter: None</li> <li>b. Output parameter: Location location</li> </ul> </li> </ul>
Pre/Post condition	<ul style="list-style-type: none"> <li>i. sendLocation():location <ul style="list-style-type: none"> <li>a. Pre-condition: User has checked in</li> <li>b. Post-condition: User location is stored into 'location' object</li> </ul> </li> </ul>

## 4.3.2 Physical Database Design

### 4.3.2.1 User table

The create table command for user table is as follow:

```
CREATE TABLE `user` (
  `id` int(50) NOT NULL,
  `name` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `phoneno` varchar(12) COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `password` varchar(100) COLLATE utf8_unicode_ci NOT NULL
)
```

Column 'id' is set as primary key and 'phoneno' is set as unique key. The command is as follow:

```
ALTER TABLE `user`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `phoneno` (`phoneno`);
```

Since column 'id' is set as the primary key, the column is set to autoincrement. The command is as follow:

```
ALTER TABLE `user`
  MODIFY `id` int(50) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;
```

#### 4.3.2.1 Friend table

The create table command for friend table is as follow:

```
CREATE TABLE `friend` (
  `f_id` int(5) NOT NULL,
  `r_id` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `a_id` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `f_status` char(1) COLLATE utf8_unicode_ci NOT NULL DEFAULT 'P'
)
```

Column 'f\_id' is the primary key for table friend, and columns 'r\_id' and 'a\_id' are the foreign key in table friend. The DDL command is as follow:

```
ALTER TABLE `friend`
```

```
ADD PRIMARY KEY (`f_id`),
```

```
ADD KEY `r_id` (`r_id`),
```

```
ADD KEY `a_id` (`a_id`);
```

The following command is to set column 'f\_id' to autoincrement:

```
ALTER TABLE `friend`
```

```
MODIFY `f_id` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
```

#### 4.3.2.2 Location table

The create table command for table location is as follow:

```
CREATE TABLE `location` (
  `l_id` int(255) NOT NULL,
  `latitude` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `longitude` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `id` int(50) DEFAULT NULL,
  `timestamp` datetime NOT NULL
)
```

Column 'l\_id' is the primary key for location table and column 'id' is the foreign key in this table. The DDL command is as follow:

```
ALTER TABLE `location`
  ADD PRIMARY KEY (`l_id`),
  ADD KEY `id` (`id`);
```

Column 'l\_id' is set as autoincrement with limit of three digits and the command is as follow:

```
ALTER TABLE `location`
  MODIFY `l_id` int(255) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
```

#### 4.4 Conclusion

To conclude, the Find Me Friends: GPS Tracker will be using an architecture. The architecture consists of many layers. The layers are meant to help the other layers. In the navigation design, how the user will navigate the system is explained. The input design showed the screens that the user will interact with. The output design explained the type of outputs the user will receive. The software design described in detail the classes in every module of the Find Me Friends:

GPS Tracker. The physical database design explains the logical design to physical design of the database.



## CHAPTER V

### IMPLEMENTATION



#### 5.1 Introduction

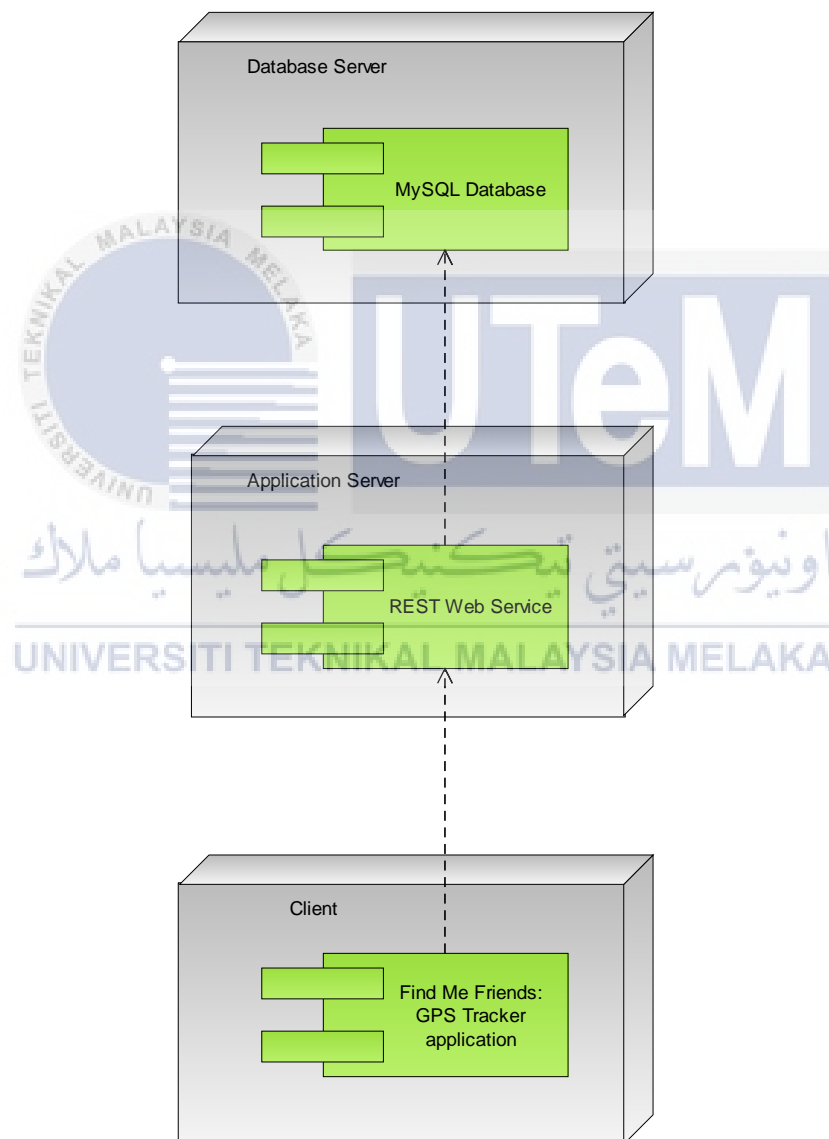
اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

This chapter is going to explain about the implementation phase of the development of Find Me Friends: GPS Tracker application. The activity involved in this phase is the development of every modules of the application that are user authentication, friend authentication, user detection, and distance calculation. The expected output of this phase is a fully functional Find Me Friends: GPS Tracker application. Section 5.2 will explain about the software development setup of the project. Section 5.3 will be about software configuration management and section 5.4 is the implementation status of the project.

## 5.2 Software Development Environment Setup

Find Me Friends: GPS Tracker used a three-tier architecture, where all the user interface, process logic and data storage are developed and maintained separately. This makes the whole system easy to maintain and easy to find and correct any errors. The following diagram is the deployment diagram of the Find Me Friends: GPS Tracker.



**Figure 5.1: Deployment diagram**



Find Me Friends: GPS Tracker is developed primarily using Java. The visual structure of the user interface of the application is developed using Android's XML. The layout elements will later be instantiated in the activity class before being used. This application relies on REST web service to connect to the remote database. REST web service is a web application that has no graphical user interface (GUI) and it handles the communication between the mobile application and the server using HTTP method either POST or GET. The respond is sent back to the application in JSON. The web service is written in PHP.

The process of designing and coding the mobile application is done in Android Studio. Android Studio is the Android's official IDE and it is very convenient to use. It provides advanced code completion, specifically catered to Android application development. The user interface is easy to understand where the project structure, debugger and other functions included all in the main interface. It also provides drag-and-drop function for the mobile application GUI. On the other hand, the REST API is written using Notepad++ because the interface is very straightforward with code-completion capability.

### 5.3 Software Configuration Management

#### 5.3.1 Configuration Environment Setup

Software configuration management is very important to software development because it helps to keep track of the version changes and track error, if any. The project's software configuration management used the built-in version control system (VCS) in the Android Studio. Some of the VCS integrated in the Android Studio are Git, Mercurial, Github and more.

For the Find Me Friends: GPS Tracker application development, Github is used because it is easy to understand and use.

To start using Github from Android Studio, first create an account in Github. In Android Studio, select Github as the chosen VCS for the project developed. Then, log into the Github account from Android Studio. To start storing the project's coding and other necessary data for the project's development, create a repository to store the project on Github and then select all files to be uploaded into the Github repository for initial commit. Each successful stage of the project development is pushed to the Github.

### 5.3.2 Version Control Procedure

The version control procedure of the implementation phase of this application is shown and described in the following table.

**Table 5.1: Version control procedure**

Version	Description
FMF: GT v 1.0	This is the initial version of the application where all the GUI are completed with no function included yet. This version is important to understand the navigation flow of the application.
FMF: GT v 1.1	This version includes the first module of the application, user authentication.
FMF: GT v 1.2	In this version, the application has the friend authentication module added.
FMF: GT v 1.3	This third version of the application has three modules completed, with user detection module included.
FMF: GT v 1.4	This version has all the module completed, combining the distance calculation module.

## 5.4 Implementation Status

The following table contains the details of the implementation status of Find Me Friends: GPS Tracker application.

**Table 5.2: Find Me Friends: GPS Tracker implementation status**

Module name	Description	Duration to Complete	Date Completed
User authentication	This module includes the login and register function of the application.	20d	29 March 2017
Friend authentication	The friend authentication includes viewing friend list, searching and adding friend, and viewing and reacting to friend request.	15d	13 April 2017
User detection	This is where user can check in and other user can view the checked-in location on the map.	23d	6 May 2017
Distance calculation	Calculate the distance between users and display it on the map.	14d	20 May 2017

## 5.5 Conclusion

In a nutshell, this chapter explained the implementation phase of the development of Find Me Friends: GPS Tracker application. The first step in the implementation module is the software development setup of the project, which involves software configuration setup and the version

control procedure to manage the code. Later, during development, the application is developed module by module, starting from the user registration module, followed by access authentication module, user mapping module and the last module developed is distance between user calculation. The output of this implementation phase is a fully functional Find Me Friends: GPS Tracker application. The next activity to be developed is the testing phase of the project.



## CHAPTER VI

### TESTING



#### 6.1 Introduction

This chapter is going to explain about the testing phase of the development of Find Me Friends: GPS Tracker application. Various types of testing will be conducted in this phase to ensure that the application is working properly. The testing process will cover all the modules of the application, and the objective of the testing is to ensure that the application meet all the functional requirements stated in Section 3.2.2 in Chapter 3. Three types of testing will be conducted that is unit testing, integration testing, and user acceptance testing. Section 6.2 will describe the test plan for the testing phase of this project. Section 6.3 will explain in detail the test strategy, followed by test design in Section 6.4. The result and analysis of the testing phase will be in Section 6.5.

## 6.2 Test Plan

### 6.2.1 Test Organization

This section will describe the personnel involved in the testing process. The tester selected for this project is chosen based on prior knowledge in testing and availability to perform the testing.

The test organisation for this application testing is as follow:

Test Lead: Syafida binti Mohammad Hasbullah

Unit testing: Syafida binti Mohammad Hasbullah

Integration testing: Syafida binti Mohammad Hasbullah, Nurul Syafika Khairina binti Hasbollah

User Acceptance testing: Izzlyn Izzaty binti Zulkefli, Nurzafirah binti Zamri

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

### 6.2.2 Test Environment

The testing is carried out at Universiti Teknikal Malaysia Melaka. The application is installed into the testers' personal smartphone and they will start to conduct the test according to the test requirement of this project.

The hardware required for the testing is smartphone running on Android OS with GPS capability and internet connectivity. The details of the hardware used for the testing is stated in the following table.

**Table 6.1: Testing device configuration**

	<b>Tester 1</b>	<b>Tester 2</b>	<b>Tester 3</b>	<b>Tester 4</b>
<b>Model name</b>	Xiaomi Redmi 2 Enhanced	Vivo Y51	Xiaomi Redmi 4a	ZTE Blade A310
<b>OS</b>	Android 5.1.1	Android 5.0.2	Android 6.0.1	Android 6.0.1
<b>CPU</b>	Quad-core 1.2GHz Cortex-A53	Quad-core 1.2GHz Cortex-A53	Quad-core 1.4GHz Cortex-A53	Quad-core 1260MHz ARM Cortex A7
<b>RAM</b>	2.00 GB	2.00 GB	2.00 GB	1.00 GB
<b>Internet</b>	GSM/HSPA/LTE	GSM/HSPA/LTE	GSM/HSPA/LTE	GSM/UMTS/LTE
<b>GPS</b>	Available	Available	Available	Available

In the above table, Tester 1 is Syafida binti Mohammad Hasbullah, Tester 2 is Nurul Syafika Khairina binti Hasbollah, Tester 3 is Izzlyn Izzaty binti Zulkefli and Tester 4 is Nurzafirah binti Zamri. These identification will be used in Section 6.5.

There is no training required prior to testing. However, prior to testing, the testers will be briefed on the background of the project and the objectives of the testing. Then, they will be briefed on how the application works as stated in user manual in Appendix B.

### 6.2.3 Test Schedule

Find Me Friends: GPS Tracker development used Rapid Application Development methodology where several phases are repeated before deployment. The testing is repeated a few times before the application is finished. The duration and cycle of the testing is described in the table below.

**Table 6.2: Test schedule**

Activity	Component Tested	Duration	No. of Cycle	Start Date	End Date
Unit testing	Module 1	1d	6	29 March 2017	29 March 2017
	Module 2	1d	4	11 April 2017	11 April 2017
	Module 3	1d	8	4 May 2017	4 May 2017
	Module 4	1d	3	18 May 2017	18 May 2017
Integration testing	Find Me Friends: GPS Tracker	3d	8	18 May 2017	20 May 2017
User Acceptance Testing	Find Me Friends: GPS Tracker	2d	3	14 August 2017	15 August 2017

In the table above, Module 1 refers to user authentication module, Module 2 refers to friend authentication module, Module 3 refers to user detection module, and Module 4 refers to distance calculation module.

### 6.3 Test Strategy

The test strategy selected for the testing of Find Me Friends: GPS Tracker application is bottom-up approach, where the smallest components of the application is tested first before the



higher level component is tested. The modules are tested right after they are finished. The finished module will be integrated with the previous module and tested again. This process is repeated until the implementation is finished.

Black box and white box testing are both chosen for the development of this application. Black box testing, where the testing is conducted without any regard to the internal structure of the application, is chosen for higher level of testing that is User Acceptance testing. This is important to ensure that the application satisfies the functional requirement stated previously in Section 3.2.2.

On the other hand, white box testing is where testing is conducted concerning the internal structure of the application. Unit testing and integration testing are part of the white box testing. This testing is important to pinpoint any errors in the coding of the application, and to ensure that the development of the other module will not be affected.

### 6.3.1 Classes of Tests

There are three classes of test involved in the testing phase of this project. The classes of test are described in detail in the following table.

**Table 6.3: Test classes**

<b>Class of Test</b>	<b>Description</b>
Unit testing	The test is conducted on individual module independent of each other. The objective of this test is to make sure that the code meets its design and requirements, and behaves correctly.
Integration testing	The test is focused on testing two or more modules after integrating them. The objective of conducting this test is to ensure that the modules can interact properly with each other.

User Acceptance testing	The test is done on the finished application with actual user. The objective of performing this test is to make sure that the application can function as intended in real-world scenario.
-------------------------	---

## 6.4. Test Design

### 6.4.1 Test Description

The test case identification, test cases and expected result for each module are designed and documented as shown in the table below. In the following table, user authentication module is referred to as Module 1, friend authentication module as Module 2, user detection module as Module 3, and distance calculation module as Module 4.

**Table 6.4: Test case**

Test Case ID	Test Requirement	Description	Expected Result
<b>Module 1</b>			
FMF_GT_1	Validate that new user can register if they insert data to all required fields	Check that user can create an account	User can register
FMF_GT_2	Validate that new user cannot register if they did not insert data to all required fields	Check that user cannot create an account due to insufficient information	User cannot register
FMF_GT_3	Validate that new user cannot register	Check that user cannot create an	User cannot register

	if they did not fill the registration form	account when register form is empty	
FMF_GT_4	Validate that user can login if they enter the correct phone number and password combination	Check that user can login with the correct login credentials	User can login
FMF_GT_5	Validate that user cannot login if they enter incorrect phone number or password	Check that user cannot login with due to incorrect login credentials	User cannot login
FMF_GT_6	Validate that user cannot login if they did not fill in the login form	Check that user cannot login when login forms is empty	User cannot login
<b>Module 2</b>			
FMF_GT_7	Validate that user can add friends that have register into the system	Check that user can add friends into their friend list	User can add friends
FMF_GT_8	Validate that user cannot add friends that have not register into the system	Check that user cannot find friends that have not register into the system	The search result is empty
FMF_GT_9	Validate that search friend result return empty when no keyword is entered	Check that no result is displayed when there is no keyword entered	The search result is empty
<b>Module 3</b>			

FMF_GT_10	Validate that user can see their own and friend's location only	Check that user can see their own location and their friend's location	User can see their location and their friend's location
FMF_GT_11	Validate that user cannot see anyone's location when their friends have not checked in	Check that user cannot see anyone else's location when their friends have not checked in	User can only see their own location on the map and get a message that their friends have not check in
FMF_GT_12	Validate that the user will not be able to see anyone's location when they have no friends	Check that user will not be able to see anyone else's location when they have no friends	User can only see their own location on the map and get a message that their friends have not check in
<b>Module 4</b>			
FMF_GT_13	Validate that user can see the distance between user and friend	Check that user can see the result of distance calculation	User can see the distance between their location and their friend's location
FMF_GT_14	Validate that user cannot see the distance between user and friend when the friend has not checked in	Check that user cannot see result of distance calculation when their friends have not checked in	User can only see their own location on the map and get a message that their friends have not check in
FMF_GT_15	Validate that user cannot see any distance result when there is no friend in friend list	Check that user cannot see result of distance calculation when they have no friends	User can only see their own location on the map and get a message that their

			friends have not check in
--	--	--	------------------------------

#### 6.4.2 Test Data

To conduct the testing that has been designed, test data is crucial. The data used for unit testing and integration testing are experimental data, where the phone number entered may or may not exist and may or may not belong to the name registered in the system. The data collection technique used is questionnaire where the testers are required to answer according to the set of answer provided.

Based on the test cases stated in Section 6.4.1, the test data for FMF\_GT\_1 to FMF\_GT\_9 are detailed in the following table.

**Table 6.5: Test data**

Test Case ID	Field	Test Data
FMF_GT_1	Name	Han
	Phone number	0123456789
	Email	<a href="mailto:han@gmail.com">han@gmail.com</a>
	Password	han123
FMF_GT_2	Name	Jack
	Phone number	0123456780
	Email	<a href="mailto:jack@gmail.com">jack@gmail.com</a>
	Password	<i>Null</i>
FMF_GT_3	Name	<i>Null</i>
	Phone number	<i>Null</i>
	Email	<i>Null</i>
	Password	<i>Null</i>

FMF_GT_4	Phone number	0123456789
	Password	han123
FMF_GT_5	Phone number	0123456789
	Password	123456
FMF_GT_6	Phone number	<i>Null</i>
	Password	<i>Null</i>
FMF_GT_7	Search	Han
FMF_GT_8	Search	Zafirah
FMF_GT_9	Search	<i>Null</i>

For FMF\_GT\_10 to FMF\_GT\_15, the tester is required to follow certain step to conduct the test. All users that are used in this step are already registered into the system. Those steps are detailed in the following table.

**Table 6.6: Test Step**

Test Case ID	Test Step
FMF_GT_10	<ol style="list-style-type: none"> <li>i. Login using '0123456789' with password 'han123'.</li> <li>ii. Add 'Jack' as friend.</li> <li>iii. Return to home screen.</li> <li>iv. Press 'My Location' button.</li> <li>v. Click 'Check in'</li> <li>vi. Logout.</li> <li>vii. Login using '0123456780' with password 'jack123'.</li> <li>viii. Press 'Friend Request' button.</li> <li>ix. Click '√' to accept Han's friend request.</li> <li>x. Return to home screen.</li> <li>xi. Press 'My Location' button.</li> </ol>
FMF_GT_11	<ol style="list-style-type: none"> <li>i. Login using '0123456789' with password 'han123'.</li> <li>ii. Add 'Jill' as friend.</li> <li>iii. Logout.</li> <li>iv. Login using '0123456781' with password 'jill123'.</li> <li>v. Press 'Friend Request' button.</li> <li>vi. Click '√' to accept Han's friend request.</li> </ol>

	<ul style="list-style-type: none"> <li>vii. Return to home screen.</li> <li>viii. Logout.</li> <li>ix. Login using '0123456789' with password 'han123'.</li> <li>x. Click 'My Location' button.</li> </ul>
FMF_GT_12	<ul style="list-style-type: none"> <li>i. Login using '0123456789' with password 'han123'.</li> <li>ii. Remove all friends from friend list.</li> <li>iii. Return to home screen.</li> <li>iv. Click 'My Location' button.</li> </ul>
FMF_GT_13	<ul style="list-style-type: none"> <li>i. Login using '0123456789' with password 'han123'.</li> <li>ii. Add 'Jack' as friend.</li> <li>iii. Return to home screen.</li> <li>iv. Press 'My Location' button.</li> <li>v. Click 'Check in'.</li> <li>vi. Logout.</li> <li>vii. Login using '0123456780' with password 'jack123'.</li> <li>viii. Press 'Friend Request' button.</li> <li>ix. Click '√' to accept Han's friend request.</li> <li>x. Return to home screen.</li> <li>xi. Press 'My Location' button.</li> </ul>
FMF_GT_14	<ul style="list-style-type: none"> <li>i. Login using '0123456789' with password 'han123'.</li> <li>ii. Add 'Jill' as friend.</li> <li>iii. Logout.</li> <li>iv. Login using '0123456781' with password 'jill123'.</li> <li>v. Press 'Friend Request' button.</li> <li>vi. Click '√' to accept Han's friend request.</li> <li>vii. Return to home screen.</li> <li>viii. Logout.</li> <li>ix. Login using '0123456789' with password 'han123'.</li> <li>x. Click 'My Location' button.</li> </ul>
FMF_GT_15	<ul style="list-style-type: none"> <li>i. Login using '0123456789' with password 'han123'.</li> <li>ii. Remove all friends from friend list.</li> <li>iii. Return to home screen.</li> <li>iv. Click 'My Location' button.</li> </ul>

## 6.5 Test Result and Analysis

The test result of the testing is shown in the table below.

**Table 6.7: Test Result**

Test Case ID	Expected Result	Actual Result	Test Result (P/F)	Tester ID
FMF_GT_1	User can register	User can register	P	Tester 1,
FMF_GT_2	User cannot register	User cannot register	P	Tester 1
FMF_GT_3	User cannot register	User cannot register	P	Tester 1
FMF_GT_4	User can login	User can login	P	Tester 1
FMF_GT_5	User cannot login	User cannot login	P	Tester 1
FMF_GT_6	User cannot login	User cannot login	P	Tester 1
FMF_GT_7	User can add friends	User can add friends	P	Tester 1
FMF_GT_8	The search result is empty	The search result is empty	P	Tester 1
FMF_GT_9	The search result is empty	The search result is empty	P	Tester 1
FMF_GT_10	User can see their location and their friend's location	User can see their location and their friend's location	P	Tester 1
FMF_GT_11	User can only see their own location on the map and get a message that	User can only see their own location on the map and get a message that	P	Tester 1



	their friends have not check in	their friends have not check in		
FMF_GT_12	User can only see their own location on the map and get a message that their friends have not check in	User can only see their own location on the map and get a message that their friends have not check in	P	Tester 1
FMF_GT_13	User can see the distance between their location and their friend's location	User can see the distance between their location and their friend's location	P	Tester 1
FMF_GT_14	User can only see their own location on the map and get a message that their friends have not check in	User can only see their own location on the map and get a message that their friends have not check in	P	Tester 1
FMF_GT_15	User can only see their own location on the map and get a message that their friends have not check in	User can only see their own location on the map and get a message that their friends have not check in	P	Tester 1

## 6.6 Conclusion

In a nutshell, this chapter focuses on details of the testing stage of the development of Find Me Friends: GPS Tracker application. The first step in this phase is the test plan which describe all the personnel involved in the testing process, the test environment where the testing will be conducted, and the test schedule to ensure that the testing process is on schedule with the whole project schedule. Next, the classes of test are described in the test strategy. This step is followed by the test design which includes the test description and test data. The last step in testing phase is the analysis of test result which is explained in Section 6.5. The next activity will be the conclusion of this project.



## CHAPTER VII

### CONCLUSION



#### 7.1 Observation on Weaknesses and Strengths

While the project execution was a success, there are some weaknesses and strengths on the project that can be observed. One of the weakness of Find Me Friends: GPS Tracker is the application rely heavily on the Internet. Without the internet, the application will not work at all. Therefore, the ability to find the other friend is limited to the internet availability itself. This can hinder the objective of to find friends or family in a crowd. Another weakness of the application is it relies on user input for location tracking. The user is required to 'check in' to their location before their location can be shared and accessed by their family or friend. This limitation can also be a hindrance to the achievement of the main objective of the project.

Despite its weaknesses, there are also some strengths which can be observed on the application. True to its main objective, the application can help a person to find their friends and relative using the application on their smartphone as the medium. They simply have to install the

application on their smartphone and remember to 'check in' to their locations. Besides that, the application is also very easy to use. It has a very simple interface and a very straight-forward way of how to use it. Therefore, anyone can understand how to use it. It will not take a long time to grasp how to use the application, and exploit its function to its best and the benefit of the user themselves.

## 7.2 Proposition for Improvement

To overcome some of the weaknesses of the application, there are some suggestions which can be implemented to improve the application.

The first suggestion to improve the application is to add the reminder notification to the user to 'check in' to their location after a period of time. This can help improve the location tracking function of the application. This reminder can be set automatically by the other user for some specified period of time to ensure that the other user is traceable.

The next proposition for improvement of the application is the application can send SOS message with their exact location to some assigned emergency contacts. This feature can help the user to send distress message immediately in the face of an emergency. The SOS button should be easy to access by the user such as via a widget on the home screen.

Another recommendation on how the application can be improved better is the application can use extra features for location tracking such as cell tracking. Cell tracking can improve the ability to track user even without the availability of the internet because it relies on the signal from the smartphone to the cell tower.

### 7.3 Project Contribution

Find Me Friends: GPS Tracker has been able to contribute to some UTeM's students to track their friends. The application helps the user to find their friend without the hassle of having to call and find the relative location of the other user. The application made it easy to track friends because it can show the exact location of the other user via the GPS feedback from their smartphone.

### 7.3 Conclusion

Finding a person in a place bustling with people is very difficult. This can lead to the danger of being lost, or pose threat to the safety of the other person such as elderly and small children. Therefore, this project has been set to achieve one main objective; to locate the location of relative or friend in a crowded area. With that objective in mind, the project has been going through a lot of processes; refining the requirements, analysing the problems, designing the application, implementing the design, and testing of the project. The journey has led to the achievement of the project objective. The road to the end has been difficult but the project finally comes to an end.

## REFERENCES

V. C. Ta, D. Vaufreydaz, T. K. Dao and E. Castelli, "Smartphone-based user location tracking in indoor environment," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Alcalá de Henares, 2016, pp. 1-8.

doi: 10.1109/IPIN.2016.7743704

Ireland, K. (2016, October 26). The Advantages of Using Technology to Track People. Retrieved April 4, 2017, from <http://smallbusiness.chron.com/advantages-using-technology-track-people-30720.html>

What are the advantages and disadvantages of mobile phone tracking systems? (2012, October 10). Retrieved April 4, 2017, from <http://www.diamondsolutionsuk.com/mobile-phones/what-are-the-advantages-and-disadvantages-of-mobile-phone-tracking-systems/>

H. Liu, H. Darabi, P. Banerjee and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067-1080, Nov. 2007.

doi: 10.1109/TSMCC.2007.905750

H. Tang, J. Shi and K. Lei, "A smart low-consumption IoT framework for location tracking and its real application," *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Beijing, 2016, pp. 306-309.

doi: 10.1109/ICEIEC.2016.7589744

Department of Social Welfare. (n.d.). Official Portal Department of Social Welfare. Retrieved April 5, 2017, from

<http://www.jkm.gov.my/jkm/index.php?r=portal%2Fleft&id=UnN2U3dtUHhacVN4aHNPbUIPayt2QT09>

World Health Organization, The World Bank. (2011). *World Report on Disability 2011*. Geneva, Switzerland: World Health Organization.

K. Salhi, A. M. Alimi, P. Gorce and M. M. Ben Khelifa, "Navigation assistance to disabled persons with powered wheelchairs using tracking system and cloud computing technologies," *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, Grenoble, 2016, pp. 1-6.  
doi: 10.1109/RCIS.2016.7549316

Hubstaff. (n.d.). Everything You Need to Know About Employee Monitoring. Retrieved April 21, 2017, from [https://hubstaff.com/employee\\_monitoring](https://hubstaff.com/employee_monitoring)

S. Chandrasekaran and N. U. Maheswari, "Overview on location tracking for authentication using smartphones," *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, 2016, pp. 1-6.  
doi: 10.1109/ISCO.2016.7726919

Rouse, M. (2014, November). What is remote patient monitoring (RPM)? - Definition from WhatIs.com. Retrieved April 21, 2017, from <http://searchhealthit.techtarget.com/definition/remote-patient-monitoring-RPM>

P. C. Liang and P. Krause, "Smartphone-Based Real-Time Indoor Location Tracking With 1-m Precision," in *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 3, pp. 756-762, May 2016.  
doi: 10.1109/JBHI.2015.2500439

Fuller, J. (2008, July 28). How Bluetooth Surveillance Works. Retrieved April 21, 2017, from <http://electronics.howstuffworks.com/bluetooth-surveillance2.htm>

L. R. Vijayasarathy and C. W. Butler, "Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?" in *IEEE Software*, vol. 33, no. 5, pp. 86-94, Sept.-Oct. 2016.

doi: 10.1109/MS.2015.26

Cellhire. (n.d.). How does GPS Tracking Work on Smartphones? Retrieved April 24, 2017, from <https://www.vismo.com/how-it-works>





## BIBLIOGRAPHY

History of GPS – Mio Technology. (n.d.). Retrieved January 05, 2017, from

<http://www.mio.com/technology-history-of-gps.htm>

Lee, D. (n.d.). What is GPS? Retrieved January 05, 2017, from

<http://www8.garmin.com/aboutGPS/>





## APPENDIX A: PROPOSAL FORM

UTeM/FTMK/PSM/1

Kod Projek : BITU 3973

 <p>اوتيمر سیتی تیکنیکل ملیسيا ملاک UNIVERSITI TEKNIKAL MALAYSIA MELAKA</p>	<p><b>UNIVERSITI TEKNIKAL MALAYSIA MELAKA</b> FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY</p>
<p><b>PROJEK SARJANA MUDA 1</b> <b>PROPOSAL FORM</b> [Incomplete form will be rejected]</p>	
<b>A</b>	<p><b>TITLE OF PROPOSED PROJECT:</b> <i>Tajuk projek yang dicadangkan:</i></p> <p><b>Location Tracking Application</b></p>
<b>B DETAILS OF STUDENT / MAKLUMAT PELAJAR</b>	
<b>B(i)</b>	<p><b>Name of Student:</b> <i>Nama Pelajar:</i> SYAFIDA BINTI MOHAMMAD HASBULLAH</p> <p><b>Identity card no.:</b> <i>No. Kad Pengenalan:</i> 950510-13-5730</p> <p><b>Student card no.:</b> <i>No. Kad Pelajar:</i> B031410238</p>
<b>B(ii)</b>	<p><b>Correspondence Address:</b> <i>Alamat Surat Menyurat:</i> Lot 873, Kampung Petanak Daya, 96400 Mukah, Sarawak</p>
<b>B(iii)</b>	<p><b>Program Pengajian:</b> <i>Study Program:</i></p> <p><input type="checkbox"/> BITC    <input type="checkbox"/> BITD    <input type="checkbox"/> BITI    <input type="checkbox"/> BITM    <input checked="" type="checkbox"/> BITS</p>
<b>B(iv)</b>	<p><b>Home Telephone No.:</b> <i>No. Telefon Rumah:</i> 084 871 477</p> <p><b>Handphone No.:</b> <i>No. Telefon Bimbit:</i> 019 889 3310</p>

B(v)	<b>E-mail Address:</b> Alamat e-mel: syfd1005@gmail.com
<b>C PROJECT INFORMATION / MAKLUMAT PROJEK</b>	
C(i)	<b>Project Area (Please tick):</b> <i>Bidang Projek (Sila tanda (√)):</i> <ul style="list-style-type: none"> <li><input type="checkbox"/> A. <b>Intelligent Information Systems</b> <i>Sistem Informasi Pintar</i></li> <li><input checked="" type="checkbox"/> B. <b>Software Technology</b> <i>Teknologi Perisian</i></li> <li><input type="checkbox"/> C. <b>Database Technology</b> <i>Teknologi Pangkalan Data</i></li> <li><input type="checkbox"/> D. <b>Computer System Technology</b> <i>Teknologi Sistem Komputer</i></li> <li><input type="checkbox"/> E. <b>Computer and Network Security</b> <i>Komputer dan Keselamatan Rangkaian</i></li> <li><input type="checkbox"/> F. <b>Networking and Distributed Computing</b> <i>Rangkaian dan Pengkomputeran Teragih</i></li> <li><input type="checkbox"/> G. <b>Immersive Technology</b> <i>Teknologi Imersif</i></li> </ul>
C(ii)	<b>Duration of this project (Maximum 12 months):</b> <i>Tempoh masa projek ini (Maksimum 12 bulan):</i> <p>Duration: <u>4 months</u>          Tempoh : <u>4 months</u></p> <p>From : <u>13 February 2017</u>          Dari : <u>13 Februari 2017</u></p> <p>To : <u>8 May 2017</u>          Hingga : <u>8 Mei 2017</u></p>
C(v)	<b>Executive Summary of Project Proposal (maximum 300 words)</b> <b>(Please include the background of project, literature reviews, objectives, project methodology and expected outcomes from the project)</b> <i>Ringkasan Cadangan Eksekutif Projek (maksima 300 patah perkataan)</i> <i>(Meliputi latar belakang projek, kajian literatur, kaedah projek, objektif dan jangkaan hasil projek)</i> <p>Location Tracking Application is a GPS-based application used to pinpoint the location of other users. The users can be their family members or friends. There are various other ways of tracking technologies available today such as RFID, cellular-based, and Bluetooth. Global Positioning System (GPS) is used because it is the most</p>

	<p>straightforward way of tracking a smartphone. The location of other users can be used to locate and keep track of their whereabouts in a given environment.</p> <p>The project will be developed using Rapid Application Development methodology. The 4-months duration is the main constraint in the development of Location Tracking Application and there are changes expected to be made during the development of this application. The project is expected to track and find users without compromising the security of the other users.</p>
<p><b>C(vi)</b></p>	<p><b>Detailed proposal of project:</b> <i>Cadangan maklumat projek secara terperinci:</i></p> <p><b>(a) Project background including Introduction / Problem Statements and Literature Reviews.</b> <i>Keterangan latar belakang projek termasuk pengenalan / pernyataan masalah dan kajian literatur.</i></p> <p><b>1. Introduction</b> Global Positioning System (GPS) has made location tracking relatively easy. GPS has been used for determining a position, navigating from one point to another, monitoring the movement of an object, creating maps, and providing precise timing. The versatility of GPS and availability of GPS-equipped devices has expanded the use of GPS into our everyday lives.</p> <p>Location Tracking Application is an Android application that use the GPS function to locate other user's location and display it on a map for tracking. The application can be used to find the location of a friend in a big, crowded place by locating their smartphone with this application on. This project is made up of four modules that is access authentication, user registration, distance between user calculation, and user mapping.</p> <p><b>2. Problem statements</b></p> <ol style="list-style-type: none"> <li>Finding the location off a person in a big and crowded place is difficult.</li> <li>Keeping track of a person under the one's care is difficult in a busy environment.</li> <li>Exposing the exact location of a person can be very dangerous.</li> </ol> <p><b>3. Literature review</b> Global Positioning System (GPS) is a positioning system that uses the information from satellites to determine the position of a person or object on Earth. GPS is a project intially developed for US Military before being released for public use in 1980s. GPS works by using the information from a transmitted signal from a GPS satellite. A GPS device will capture the signal and calculate the period between the time when information was sent and when it is recieved on Earth.</p> <p>The ever growing technology has made increase GPS availability to just about anyone. GPS is now a very common features in smartphone. One of the most common use of GPS is for navigation. Application like Google Maps and Waze use GPS data to navigate a user from one point to another. GPS is also been used in social networking such as Facebook and Wechat to find nearby friends.</p> <p><b>(b) Objective(s) of the Project.</b> <i>Objektif Projek.</i></p> <p>This project aims to achieve the following objectives:</p> <ol style="list-style-type: none"> <li>To develop an application for user locating and tracking.</li> <li>To explore the GPS function on smartphone</li> <li>To evaluate the tracking capabilities of GPS on smartphone.</li> </ol>

**(c) Project Methodology**  
*Kaedah projek*

1. Description of methodology

This project will be developed using Rapid Application Development (RAD) methodology. This methodology has five phases; business modelling, data modelling, process modelling, application generation, and testing and turnover. This methodology is chosen because this project will be developed under a very tight time constraint and RAD enables rapid delivery of final product.

In business modelling, the business requirement for Location Tracking Application will be identified. In the next phase, the information gathered in the previous phase is used to define the data objects for the project development. The data objects is later used in the process modelling process to define the flow of the application according to the business requirement identified previously. In the application generation phase, the programming and actual development of application is done. In the final phase, the application results from the previous phase is tested. This process will be repeated until all modules of this application is completed.

2. Flow chart of project activities (Enclosed in the Appendix)

3. Gantt Chart of project activities (Enclosed in the Appendix)

4. Milestones and dates


Activity	Start	Finish
Proposal discussion and submission		
Proposal assessment and verification		
Chapter 1 submission		
System development		
Chapter 2 submission		
Chapter 3 submission		
Project demonstration		
Chapter 4 submission		
Project demonstration		
PSM 1 report draft		
Final presentation and project demonstration		
PSM 1 report draft correction		

**(d) Expected Results/Benefit**  
*Jangkaan Hasil Projek*

This project will hopefully be able to help better understand how GPS on smartphone works both indoors and outdoors. This project is also hopefully will be able to help people to keep track of their family or friends and increase people's awareness to their surroundings. This application has the potential to be used to during pilgrimage in Mecca where a Tabung Haji employee can keep track of the pilgrims under their supervision. This application also has the potential to be used by parents to keep track of their children in a packed environment such as shopping mall or a festival.

UTeM/FTMK/PSM/1

D REFERENCES	
<p>History of GPS – Mio Technology. (n.d.). Retrieved January 05, 2017, from <a href="http://www.mio.com/technology-history-of-gps.htm">http://www.mio.com/technology-history-of-gps.htm</a></p> <p>Lee, D. (n.d.). What is GPS? Retrieved January 05, 2017, from <a href="http://www8.garmin.com/aboutGPS/">http://www8.garmin.com/aboutGPS/</a></p> <p>Ta, V., Vaufreydaz D., Dao T., &amp; Castelli E. (2016). Smartphone-based User Location Tracking in Indoor Environment. <i>2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)</i>. doi:10.1109/ipin.2016.7743704</p>	
E ACCESS TO EQUIPMENT AND MATERIAL (PLEASE LIST IN DETAIL) / KEMUDAHAN SEDIA ADA UNTUK KEGUNAAN BAGI PROJEK INI (SILA SENARAIKAN DENGAN TERPERINCI)	
University Universiti	Other Sources or Places Lain-lain tempat/sumber
	1. Hardware <ul style="list-style-type: none"> <li>a. PC/Laptop</li> <li>b. Android smartphone</li> </ul> 2. Software <ul style="list-style-type: none"> <li>a. Android Studio 2.2.3</li> <li>b. StarUML version 2.7.0</li> <li>c. Smartsheet</li> <li>d. Adobe Photoshop CS6</li> </ul>

F (i) Declaration by applicant / Akuan Pemohon	
Date : 6/8/2017 Tarikh :	Applicant's Signature : Tandatangan Pemohon : 

F (ii) Recommended by the Supervisor Perakuan oleh Penyelia		Recommendation by the Evaluator Perakuan oleh Penilai	
Please tick (✓) Sila tandakan (✓)	Recommended: Diperakukan:	Please tick (✓) Sila tandakan (✓)	Recommended: Diperakukan:
<input checked="" type="checkbox"/> A. Highly Recommended Sangat Disokong	<input type="checkbox"/> B. Recommended Disokong	<input type="checkbox"/> A. Highly Recommended Sangat Disokong	<input type="checkbox"/> B. Recommended Disokong
<input type="checkbox"/> C. Not Recommended (Please specify reason) Tidak Disokong (Sila Nyatakan Sebab)		<input type="checkbox"/> C. Not Recommended (Please specify reason) Tidak Disokong (Sila Nyatakan Sebab)	

UTeM/FTMK/PSM/1

<p><b>General Comments:</b> Ulasan umum:</p> <p><i>Penyediaan tahap melalui penerbitan.</i></p> <p><b>Supervisor's Name:</b> Muhammad Suhaizan bin Sulong <i>Nama penyelia:</i></p> <p><b>Signature:</b> <i>Tandatangan:</i></p> <p><b>Date:</b> <i>Tarikh:</i> 6/1/2017</p>	<p><b>General Comments:</b> Ulasan umum:</p> <p><b>Evaluator's Name:</b> Mohd Hariz bin Naim @ Mohayat <i>Nama penilai:</i></p> <p><b>Signature:</b> <i>Tandatangan:</i></p> <p><b>Date:</b> <i>Tarikh:</i> 6/1/2017</p>
--	--

**MUHAMMAD SUHAIZAN BIN SULONG**  
Pensyarah  
Jabatan Kejuruteraan Perisian  
Fakulti Teknologi Maklumat dan Komunikasi  
Universiti Teknikal Malaysia Melaka (UTeM)

PSM & PD COMMITTEE Comments

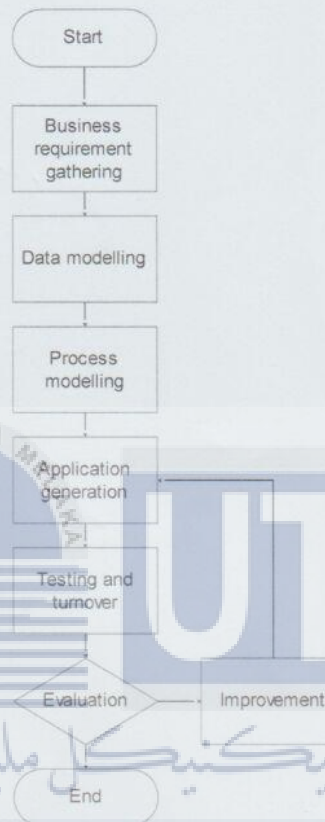


اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



## APPENDIX A



اونيورسيتي تيكنيكل مليسيا ملاك  
UNIVERSITI TEKNIKAL MALAYSIA MELAKA








UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX B

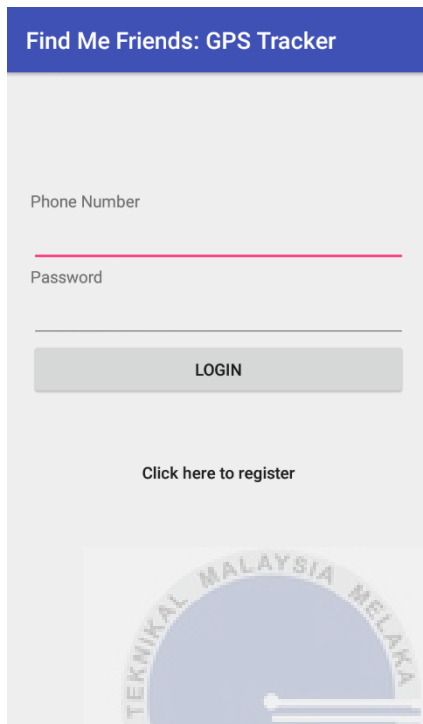
Activity	1	2	3	4	5	6	7	8	9	10	11	12	
	Week												
Proposal discussion and submission													
Proposal assessment and verification													
Chapter 1 submission													
System development													
Chapter 2 submission													
Chapter 3 submission													
Project demonstration													
Chapter 4 submission													
Project demonstration													
PSM 1 report draft													
Final presentation and project demonstration													
PSM 1 report draft correction													



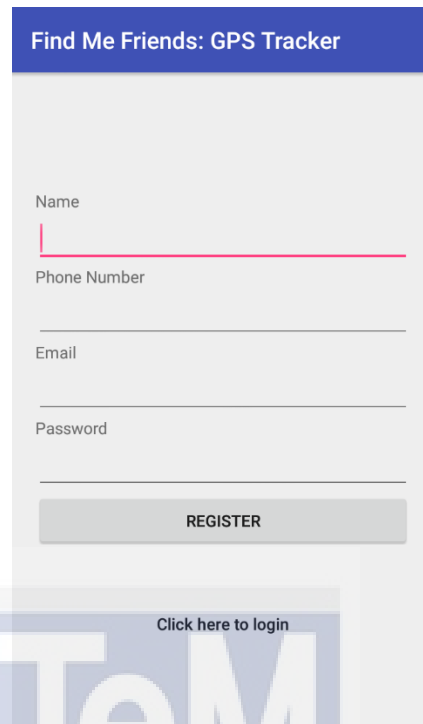
## APPENDIX B: USER MANUAL

1. Register
  - a. Click the link to register.
  - b. Fill in all the required form.
  - c. Click the 'Register' button.
  - d. Proceed with login.
2. Login
  - a. Enter phone number and password.
  - b. Click the 'Login' button.
3. Map view
  - a. Click on the 'View map' button.
  - b. User location is shown using this  pin.
  - c. Friend location is shown using this  pin.
4. Friend list
5. Add friend
  - a. Click on the  button.
  - b. Search the friend to send request to using phone number or name.
  - c. Click on  button to open phone contacts.
  - d. Nothing will show if the friend is not registered yet.
  - e. Click  button to send friend request.
6. Friend request
  - a. Click on the 'Friend request' button.
  - b. Nothing will show if there is no new friend request.
  - c. If there is friend request waiting for approval, click the  button to accept or  to reject.
  - d. A message will appear at the bottom of the screen if the action is successfully performed.

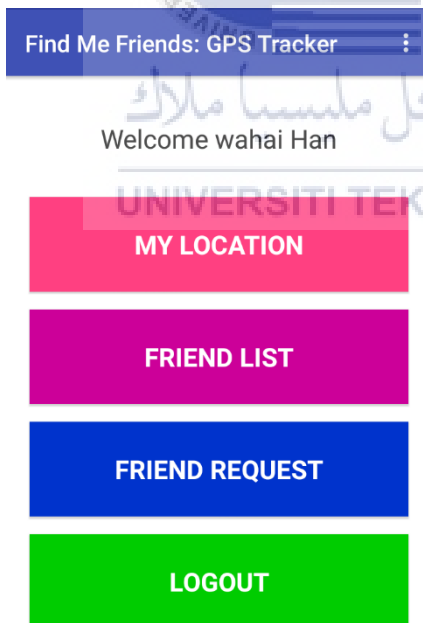
**APPENDIX C: FIND ME FRIENDS: GPS TRACKER SCREENSHOT**



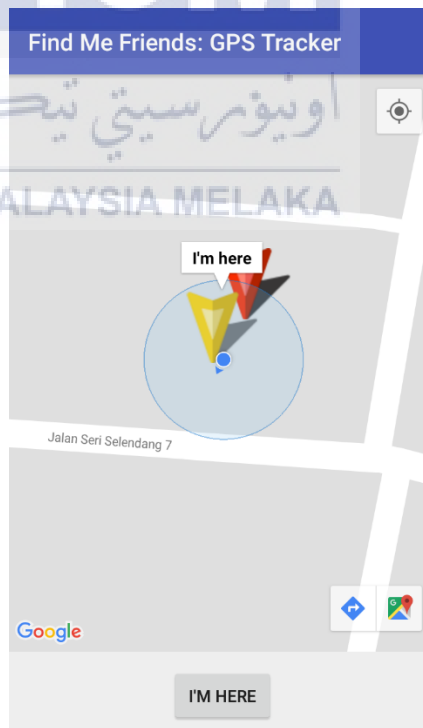
Login screen



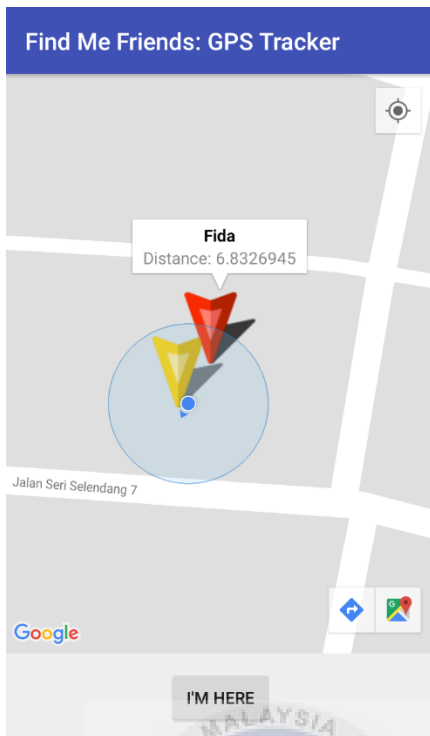
Register screen



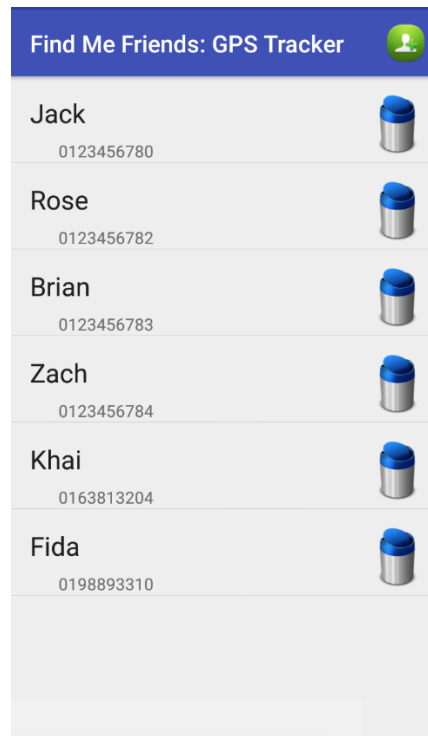
Home screen



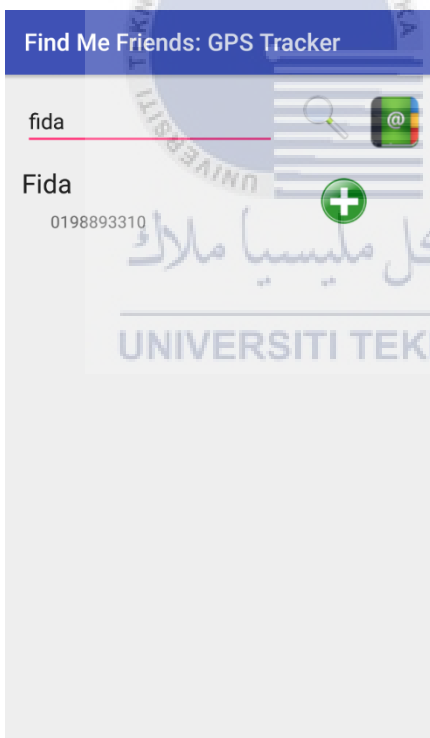
User location



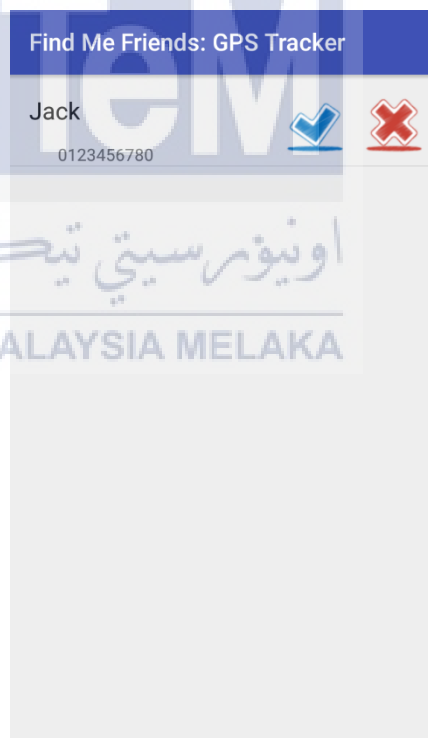
Friend's location and distance



Friends list



Search friend



Friend request

