FINAL YEAR PROJECT II

DISSERTATION

Title: Integration of Remote Vehicle Tracking System Location using GSM Modem and Web Mapping Service Application

Muhammad Ridhwan Bin Ahmad Fuad

Electrical and Electronics Engineering

12081

Integration of Remote Vehicle Tracking System Location using GSM Modem and Web Mapping Service Application

by

Muhammad Ridhwan Bin Ahmad Fuad

(Supervisor:Dr. Micheal Drieberg)

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronics Engineering)

SEPTEMBER 2012

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Integration of Remote Vehicle Tracking System Location using GSM Modem and Web Mapping Service Application

by

Muhammad Ridhwan Bin Ahmad Fuad

A project dissertation submitted to the

Electrical and Electronics Engineering Programme

UniversitiTeknologi PETRONAS

in partial fulfillment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by,

_____

(Dr. Micheal Drieberg)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

September 2012

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

_____

MUHAMMAD RIDHWAN BIN AHMAD FUAD

ABSTRACT

In Malaysia, vehicles theft cases keep increasing from days to days and the recovery rate of the stolen vehicle is very minimal. Some of service provider companies in Malaysia lack of fleet management which could cause decrease in efficiency of the services or reduce their profit as the manager or owner of the company could not monitor any transportation activities related to the companies. Remote Vehicle Tracking System could overcome or become a solution to those problems. This project is about the integration of Remote Vehicle Tracking System location using the GSM (Global System for Mobile Communication) Modem with Google Map. This project will deal with SMS (Short Message Service) enabled vehicle tracking system. The GSM modem will receive the coordinates through SMS send by transmitter (in this project, the transmitter which will act as location information sender will be simulated by using a mobile phone) and updates the location information into a database. The information then will be parsed to a preferred website (online/local-host) and the position of the vehicle will be displayed through a Google Maps application. The core of this project is the integration of the GSM modem which will act as a location information receiver and the Google Maps itself. A website will be developed to aid the user to track and view the vehicles location easily. The website will display the vehicle location through the Google Map and can be access anywhere as long as Internet connection is available. To improvise and add more functionality to the project, more functions regarding this project was added. There are three working functions for this project which are Display the Latest Tracked Vehicle Location, Display the Route History of Tracked Vehicle, lastly, Route and Destination Planner Base on Current Location.

ACKNOWLEDGEMENT

In completion of this Final Year Project, I would like to express my deepest gratitude to all parties that involved in making this project as a meaningful and great achievement for me. Throughout the project timeframe, I have learned a lot of new things and gained so much experience that will be a huge value for me in the future.

First and foremost, I would like to thank UTP's Electrical and Electronics Engineering Department for providing me guidance to complete the project and in the same time arranging the project activities for me. With their strong support and advice, I was able to plan and conduct my project very well.

Special thanks should go to my Final Year Project supervisor, Dr. Micheal Drieberg who becomes the most important person to teach and guide me in conducting this project. His strong support and willingness to share his knowledge and wide expertise has given me a very important experience in this project study.

Finally, my deepest appreciation goes to my family and my friends in UTP for their continuous support and encouragement which have enabled me to do my best for this project. I hope that after this project completion, all the findings and knowledge that I have shared through this dissertation would be useful for everyone in the study field.

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background

Remote Vehicle Tracking system is one of the systems that use basic communication system architecture to work. It needs a transmitter to send information, a medium for the information to be transferred and a receiver to receive the information. For the sake of this project, the transmitter part will be simulated by using mobile phone to send location information through SMS. GSM Cellular Network will be the medium for the information to be transferred. The GSM Modem which installed and attached at the Control Center (author personal computer) will act as receiver to receive the SMS and stored it in a MySQL Database.

To display the tracked vehicle location information in a graphical method, Google Map was chosen among other Web Mapping Service Application. Google had provided a Google Map API under Google Developer site for users who are interested to develop their own Google Map Application. Google also provide a good support to the developers by documented all the JavaScript function and syntax under the Google Developer site. Google Map API v3 is the latest version from Google which added extra functions but for this project, Google Map API v2 is used instead of v3 as v2 has a lot more technical support from previous users and developers.

The integration between the GSM Modem and Google Map is the core and the most important part in this project. The integration between the GSM Modem and Google Map can be divided into several components. The first component is the SMS Server which will serve as a platform to enable the GSM Modem at the control center to receive location information from the transmitter through a SMS. The second component is the MySQL database which will store all the received SMS. The third component is the Google Map API (Application Programming Interface) which will display the location information through a Google Map. All the system functions which consist of several JavaScrip, PHP, and HTML languages are compiled in a website template to ease the users to access and use it.

1.2 Problem Statement

1.2.1 Problem Identification

Remote vehicle tracking system had become very important nowadays as it could give lots of benefits to the users. The remote vehicle tracking system enables the user to determine and track the whereabouts and locations of their vehicle. By knowing the locations of the vehicles, a lot of things can be done by the user. For examples, the users can ensure the security of their vehicles as if their vehicles been stolen; they still can track down their vehicles and report it to the authorities. For public transport users, they can determine the location of next buses or taxi thus they can manage their time better and predict the arrival of next buses. Services Company such as PETRONAS petrol pump station, SHELL petrol pump station, Pos Malaysia courier services and many more also could get benefits from this system as they could easily locate their fuel tank lorry or courier van location and route log.

Although most of people know GPS can provide same remote tracking services for the vehicles but the main reason people does not apply it because of the cost to maintain the service. Advance car security system is too expensive. Cost for the gadget is too high and people also must pay for the service monthly to the service company.

To overcome above problems, a simpler and cheaper device and service maintenance need to be developed. This project is depend and rely on the GSM cellular network to track the vehicle which are more cheaper to maintain as it only use SMS to send the location information. Besides that, the users can view their vehicles positions and log anytime and anywhere they want through the internet.

1.2.2 Significant of the Project

The project is very significant as upon completion of this project, it can become the alternative solution and option for current and existing remote vehicle tracking systems (GPS based) which are more complicated and expensive.

This project will give lots of benefits to the users and communities. Remote Vehicle Tracking will become one of the most important services in Malaysia if the services and applications are introduced properly and communities have the knowledge regarding the service and technology.

This Remote Vehicle Tracking System which integrated with Google Map if implemented properly and widely, it will improve the quality of the public transports, reduce vehicles theft cases, improving the service company services such as PosLaju courier services and many more.

With the wide area coverage of GSM cellular network in Malaysia [1], the users not only could use the network for daily mobile phone usage, but also use it for remote vehicle tracking.

1.3 Objective & Scope of Study

1.3.1 Objectives of Project

There are three main objectives that need to be achieved for this project which are:

1. Connecting and configure the GSM modem to the control center (personal computer) and ensure it able to receive information (SMS).
2. Integrate and display the information received from GSM modem to the Google Map.
3. Developed a website to ease the users to use it and display the tracked vehicle on the Google Map and add extra functions for the system.

1.3.2 Scope of Studies

There are four main elements in this scope of studies which are:

1. GSM modem set up and configuration
2. Set up SMS system to GSM modem
3. Integrate and display location information on the Google Map
4. Website development

The GSM modem will be connected to the control center where it receives SMS from the desired transmitter and stored in a user database such as Microsoft Access Database or MySQL. There are SIM card connected to the device as well which will enable the GSM Modem to function in GSM Network.

SMS system is use to receive any incoming SMS to the GSM modem and helped to automatically saved in a database. Google Map will be used as a map and medium to display the location of desired and tracked vehicles. To use and display the map on user preferred website, a Google Map API

(Application Programming Interface) key need to be obtained by sign it up at Google development website. (*Noted that author use Google Map API v2 as newly released Google Map API v3 does not require key to functioning*)

For website development, the language scripting language will be PHP (Hypertext Pre-Processor) as it offers excellent connectivity to many databases. Java Script is used to design an interactive website and html scripting. Others software and program that had been used to developed the website will be Wampserver and also Macromedia Dreamweaver.

1.4 The Relevancy of Project

This project will be very relevant judging on certain criteria and circumstances. From above abstract and project background, this project depends solely on the GSM cellular network for location signal and the Google Map to display and illustrate the signal location.

In Malaysia, the GSM cellular network coverage had covered almost places in Malaysia and this made the development of this project is very relevant. Besides that, Internet also can be accessed almost anywhere in Malaysia through a personal computer and also mobile phone. In addition, the rates (RM/SMS) offered by most GSM Cellular Network Service Company in Malaysia are very cheap and cost only around 10 sen per SMS (average cost) thus conclude that to maintain the remote vehicle tracking system is very cheap.

Google also successfully introduce one of their applications which is Google Map to the communities. Google Map had become one of the most used Google's application and this make the project become more relevant as the communities already had been exposed with the Google Map.

The communities nowadays also could get a lot of benefits and advantages from this project. The public transport users could have a better time management as they could predict next buses or taxi arrival. Service

Provider Company such as PosLaju could update the customers courier whereabouts location information and many more benefits and advantages that can be gain from this project implementation.

Judging from above criteria and circumstances, the development of this project is very relevant.

1.5 Feasibility of the Project within the Scope and Time frame

The development and completion of the project is feasible judging from its objective and scope of studies stated. The time given to complete the project is approximately 8 months.

This project won't focus much on the hardware (GSM modem) as it is already provided. The author only needs to focus on integration of GSM modem, method on how to receive SMS from transmitter and development of website which will display the Google Map as a medium to present the location. The project also can be done in house as it involves a lot of programming.

Based on above statement, the project is feasible as the time given will be enough for the author to do research on his project and prepare a working prototype.

CHAPTER 2: LITERATURE REVIEW

This literature review will theoretically covers every elements and foundation of the project. The basis of Remote Vehicle Tracking System lies on the GSM Modem, GSM Cellular Network, SMS notification system from the transmitter to the receiver (GSM Modem), integration of the Google Map and development of website. This chapter will introduce all the related theories and information related to this project.

2.1 Remote Vehicle Tracking System

The development of tracking system is originate from the shipping industry as they wanted to determine the location and whereabouts of the ships at any given time because the company' owner found it difficult to keep track on what had happened to the ship over the wide expenses of ocean and sea [2]. The system at first place was developed in passive mode which requires real time location information of the tracked vehicles (ship) but the system could not be employed because the location information only accessible when the vehicles are available and present. To enable the location information to be accessed without the present or availability of the vehicle, an active tracking system was developed. This system also could be called Remote Vehicle Tracking System as the vehicles were able to be tracked from distances.

Remote Vehicle Tracking System integrates a hardware device which is installed in the vehicle (transmitter) to a remote Tracking Server. The location information is transmitted to the Tracking Server GSM Modem through a GSM cellular network by using SMS or direct TCP/IP connection through GPRS. For this project, the author choose the SMS method as the usage of SMS is charged base on quantities of SMS sent while the direct TCP/IP connection method through GPRS is charged base on amount of data used which will be costly if the usage of data is not controlled or overused [3].

There are four important elements and components that need to be concerned regarding this project, they are: the GSM cellular network, the GSM Modem, integration of location information with Web Mapping Service Application (Google Map) and website development.

2.2 GSM Cellular Network

The analogue cellular network was rapidly developed in Europe during early 1980's. Each country develops its own system, which was not compatible with each other [4]. Each network was limited to its boundaries thus make the system incompatible. A study group called the Group Special Mobile was formed in early 1982 to study and develop a European public land mobile system. Although the GSM was standardized in Europe, but it is not only a European Standard and during end of 1990's, there are over 200 GSM networks operated in 110 countries with 1.3 million subscribers in the beginning of 1994, the numbers were increasing rapidly as there are 350 million of subscribers by 2001 (Theodore, 2002).

There are three main elements that build up a GSM network. They are they Mobile Station (MS), the Base Station System (BSS) and the Network Switching Subsystems (NSS). Figure 1 shows the diagram of general architecture of GSM network.

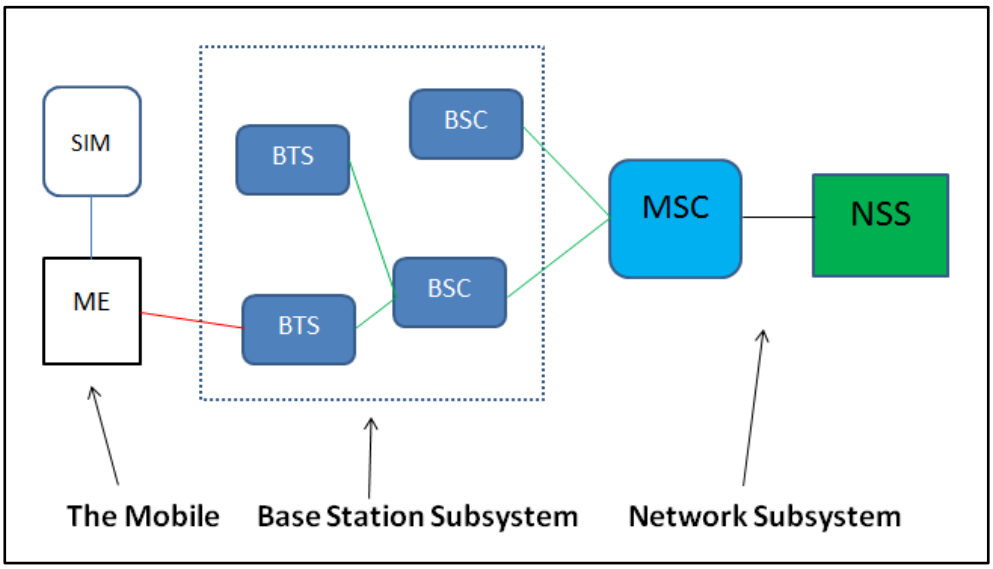


Figure 1: GSM Network General Architecture Diagram

Mobile Station (MS)

Mobile Station has two important elements which are the Mobile Equipment (ME) and also the Subscriber Identity Module (SIM). The ME refers to the physical mobile phone itself [5]. Each ME is uniquely defined by the International Mobile Equipment Identity (IMEI) number which is 15 digits long. The IMEI number can be identified by dialing *#06# and the IMEI number will be displayed by the mobile phone.

The SIM is in form of a mini smart card with the size of 10 cent coin and it is inserted to a mobile phone. The SIM card is obtained when the user or subscriber subscribed to the service provider and carries information specific to the subscriber. The SIM card also able the user to store phone numbers into it. The SIM card can be protected by a 4 digit Personal Identification Number (PIN) and if enters incorrectly three times in a row, the SIM card then can only be unblocked with an 8-digit Personal Unblocking Key (PUK) which is also stored in the SIM card. Figure 2 shows example and size comparison of SIM card.

Figure 2: Example of SIM card [6]

Base Station System (BSS)

The BSS is the system of base station equipment which is viewed by the Mobile Switching Center (MSC) [7]. Mobile Station communication is depends on the BSS. BSS is responsible in handling traffic and signaling between a mobile phone and the Network Switching Subsystem (NSS). A BSS may consist one or more Base Station Controller (BSC) and one or more Base Transceiver Station (BTS).The BTS contains the equipment to transmit and receive the radio signals, antennas and equipment for encrypting and decrypting communications with the BSC [8]. The Base Station Controller manages the radio resources for one or more BTS.

Network Switching Subsystem (NSS)

The function of the NSS is to carries out call switching and mobility functions for the mobile phone roaming on the network of base stations. It is owned by the service provider. Generally, the NSS is used for GSM services such as voice calls, SMS, and circuit switched data cells. The system architecture is expended to provide packet-switched data services known as GPRS core network which allows services such as WAP, MMS and the Internet for the mobile phone.

2.3 GSM Modem

A GSM modem is a specialized type of modem which operates over a subscription to a mobile operator service provider. A GSM modem can be inserted with a SIM card. From the mobile operator perspective, the GSM modem looks just like a mobile phone. The function of the GSM Modem for this project is to act as a receiver for incoming location information through a SMS from a transmitter devices. The GSM modem used in this project is the U-Blox EVK-G26H evaluation kit.

EVK-G26H comes with a built-in GPS receiver module which adds the flexibility to either test GSM/GPRS functionality alone or to integrate it together with GPS technology. This GSM Modem can be connected to personal computer via USB port (data communication and debug port). Figure 3 shows how the EVK-G26H looks like while Figure 4 shows the block diagram of the hardware.

Figure 3: EVK-G26H Quad Band GSM/GPRS Evaluation Kit [9]

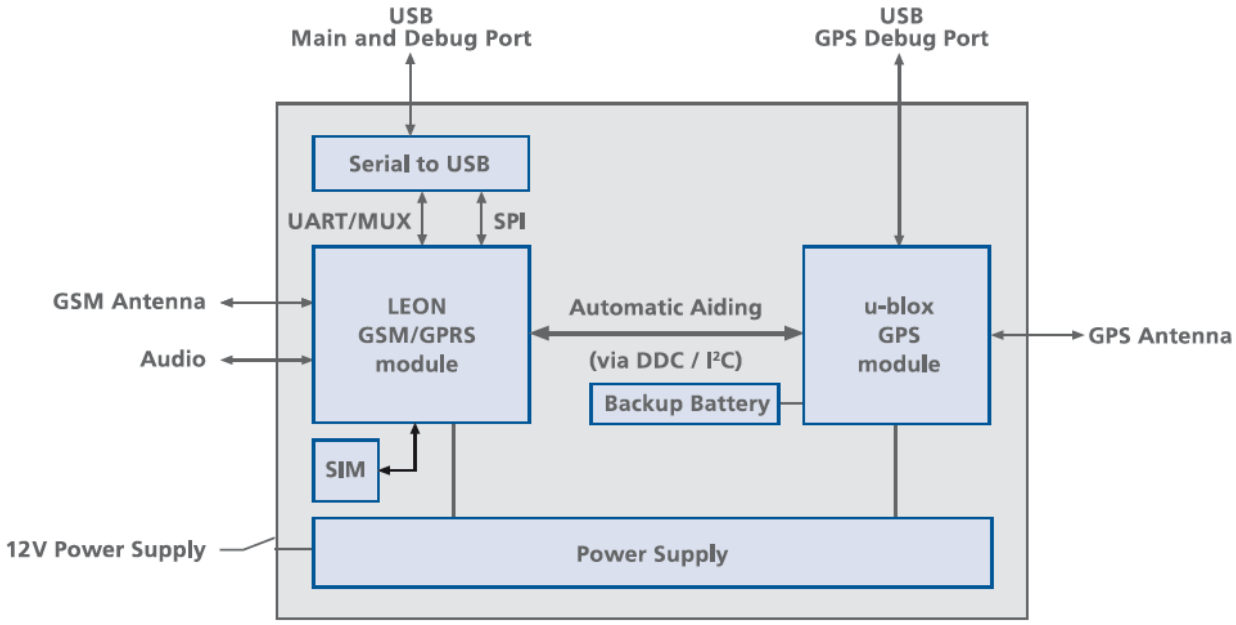


Figure 4: EVK-G26H Block Diagram [9]

2.4 Web Mapping Service Application (Google Map)

Google Map is a free web mapping service application and technology provided by the Google. The Google Map offers street maps, route planner for travelling and can be act as location locator. Beside Google Map, Google also had introduced Google Earth (originally called EarthViewer 3D) which is a stand-alone program and a more powerful tools compare to Google Map. Google Earth offers more globe-viewing features including showing polar areas compare to Google Map which cannot show areas around the poles [10]. For this project, the Google Map is favors over the Google Earth because the Google Map can be access at any web browser while Google Earth is application that needs to be downloaded and installed in order to be used. Figure 5 shows example of Google Map view from the website.

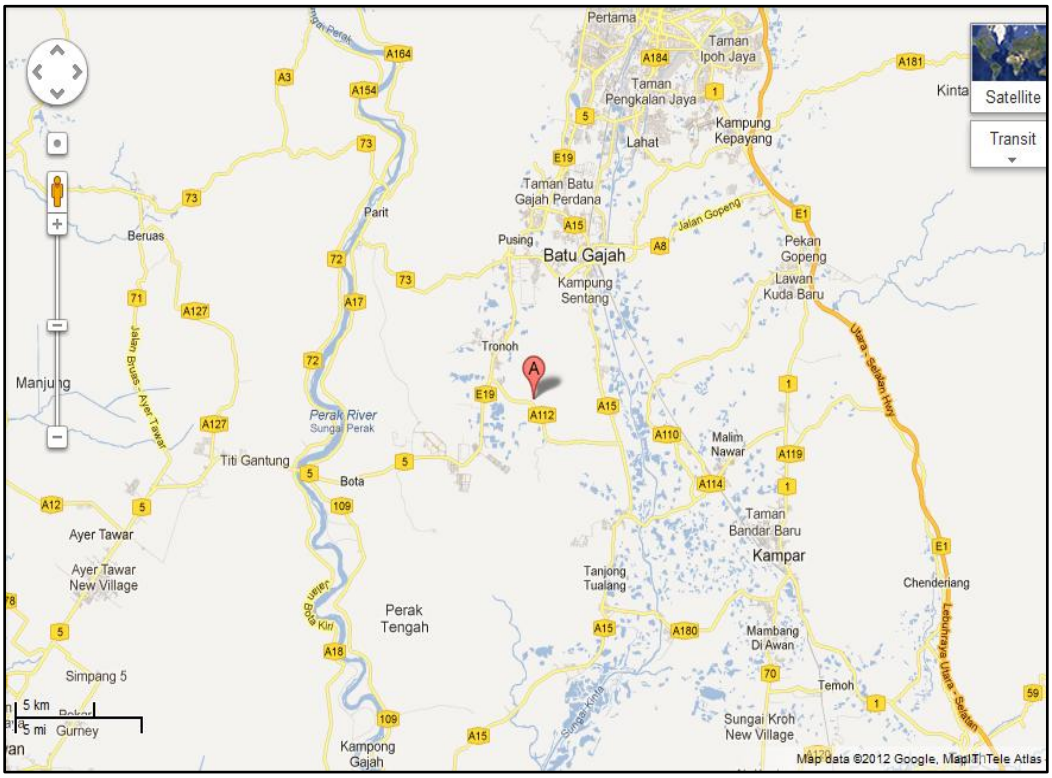


Figure 5: Example of Google Map Screenshot

June 2005, Google had launched the Google API to allow the users and developers to integrate the Google Map into their own websites [11]. Generally, API is a specification intended to be used as an interface by software components to communicate each other. Currently, Google Map API is a free service provided by Google without any ads (advertisement). To use the Google Map API, users and developers need to register for a Google Map API key at the Google developer website. (https://developers.google.com/maps/). The Google Map API allows users to creatively use the Google Map for their own applications such as, build location-based application, visualize Geospatial Data by creating 3D images with the Earth API, build maps for mobile application and also customize the maps. Figure 6 shows the example of location-based application developed by using the Google Map API.

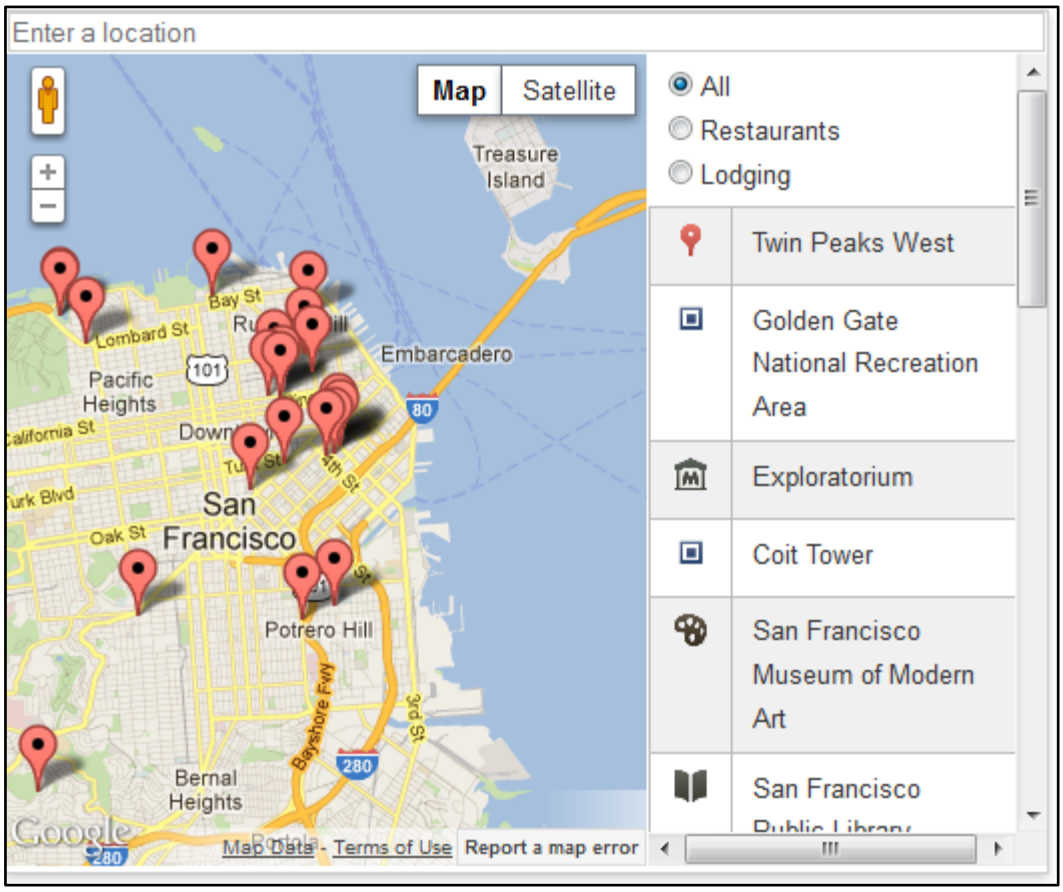


Figure 6: Location-Based Application [12]

For this project, the author will integrate the tracked location information from the transmitter through the GSM modem and display the location via the Google Map embedded on the website.

2.5 Website Development

This project will involve a lot of website development programming. The main programming languages that will be used for the web development are the PHP and Java Script. Others related software that will be used to develop the website are Wampserver, Macromedia Dreamweaver and ODBC.

Hypertext Preprocessor (PHP)

PHP is a widely-used open source general-purpose scripting language for the web development and can be embedded into HTML. HTML (HyperText Markup Language) use to display the web pages and other related

information that can be displayed in a web browser such as Modzilla Firefox, Internet Explorer and Google Chrome.

PHP script is enclosed in special start and end processing instructions which are <?php and ?> and this allow the web developer to jump into and out of the "PHP mode". Figure 7 shows example of PHP script.

Example #1 An introductory example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <title>Example</title>
    </head>
    <body>

        <?php
            echo "Hi, I'm a PHP script!";
        ?>

    </body>
</html>
```

Figure 7: Example of PHP script

The PHP code could not be viewed by clients since it is executed on the server [13]. PHP also offers excellent connectivity to many databases such as Microsoft Access through ODMC which is very essential for this project as the PHP could link and integrate the database from server side and make it online.

The Google API key will be inserted to the PHP and JavaScript and upon execution, the Google Map will embed to the desired website.

JavaScript

JavaScriptwas originally developed by Netscape. JavaScript enable users and web developers to create an interactive website. JavaScript uses syntax influenced by C language. Even though JavaScript copies many names and naming conventions from Java, but both of the languages are unrelated as Java is more towards high end and advance programming language. Figure 8 shows example of simple JavaScript and Figure 9 and 10 shows the result of the script.

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function displayDate()
{
document.getElementById("demo").innerHTML=Date();
}
</script>
</head>
<body>

<h1>My First Web Page</h1>
<p id="demo">This is a paragraph.</p>

<button type="button" onclick="displayDate()">Display Date</button>

</body>
</html>
```

Figure 8: Examples of JavaScript script

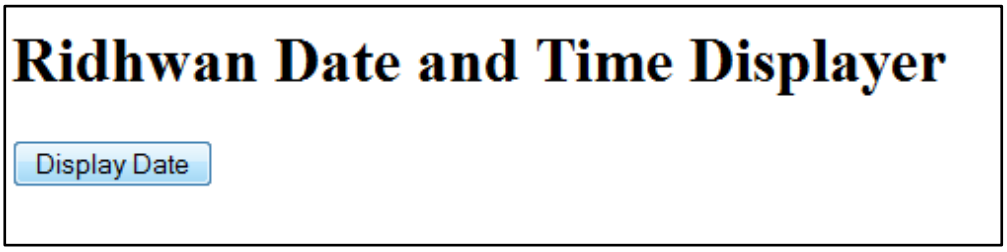


Figure 9: JavaScript Page 1

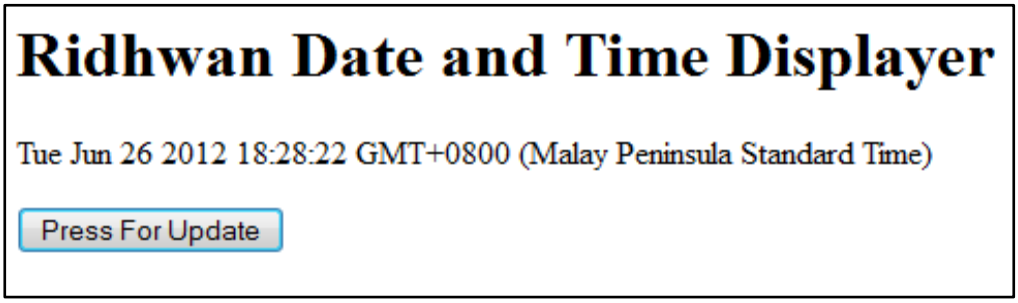


Figure 10: JavaScript page 2

For this project, the JavaScript is use to design an interactive and user friendly interface Remote Vehicle Tracking website.

WampServer

Wampserver is an open source cross-platform web server package Apache distribution that contain database such as MySQL , PHP and Pearl. The function of Wampserver is it will be used as a web development tool for web designer and programmers to execute and test their work offline which can be done without Internet access. This Wampserver can be used to try out the website in PHP or html before hosting it online through the World Wide Web (WWW).

Macromedia Dreamweaver

Adobe Dreamweaver is a web designing software which provides an interactive visual interface for HTML website editing. Web designer able to create fully- functional websites and see the layout and view of the page. For this project, the software is used to do the PHP coding.

CHAPTER 3: METHODOLOGY

In general, methodology is a set of procedure and method used to do a research and project. Methodology can be used to set a guideline for the project. By having a good methodology, the project can be executed in a more systematic order and more manageable in term of time, cost and feasibility of the project itself.

3.1 Research Methodology

3.1.1 Project Planning

This phase involves a lot with project background research and reading project related materials. Journal, research papers and discussion in forum had helped the author to have a better understanding regarding the project topic and scopes. Discussion with project supervisor, Dr. Michael Drieberg also had helped the author to have a wider view regarding the project implementation and expansion. After gather the required information and knowledge regarding the project, the development of the project can be carried out and planned in more specific steps.

This project will be involved a lot on software and programming related task and activities and some hardware related configuration. The project is emphasizing on manipulation of the Google Map usage and expansion of its current existing features as a tracked vehicle location displayer. More time will be consumed to add more functions and features for the Remote Vehicle Tracking system. By using existing GSM modem, hardware side only involve on the modem configuration and integration with the SMS notifying system and integrate the location information to the database and display it online through the Google Map. Figure 11 shows the flow of project planning.

Project Planning Flow Chart

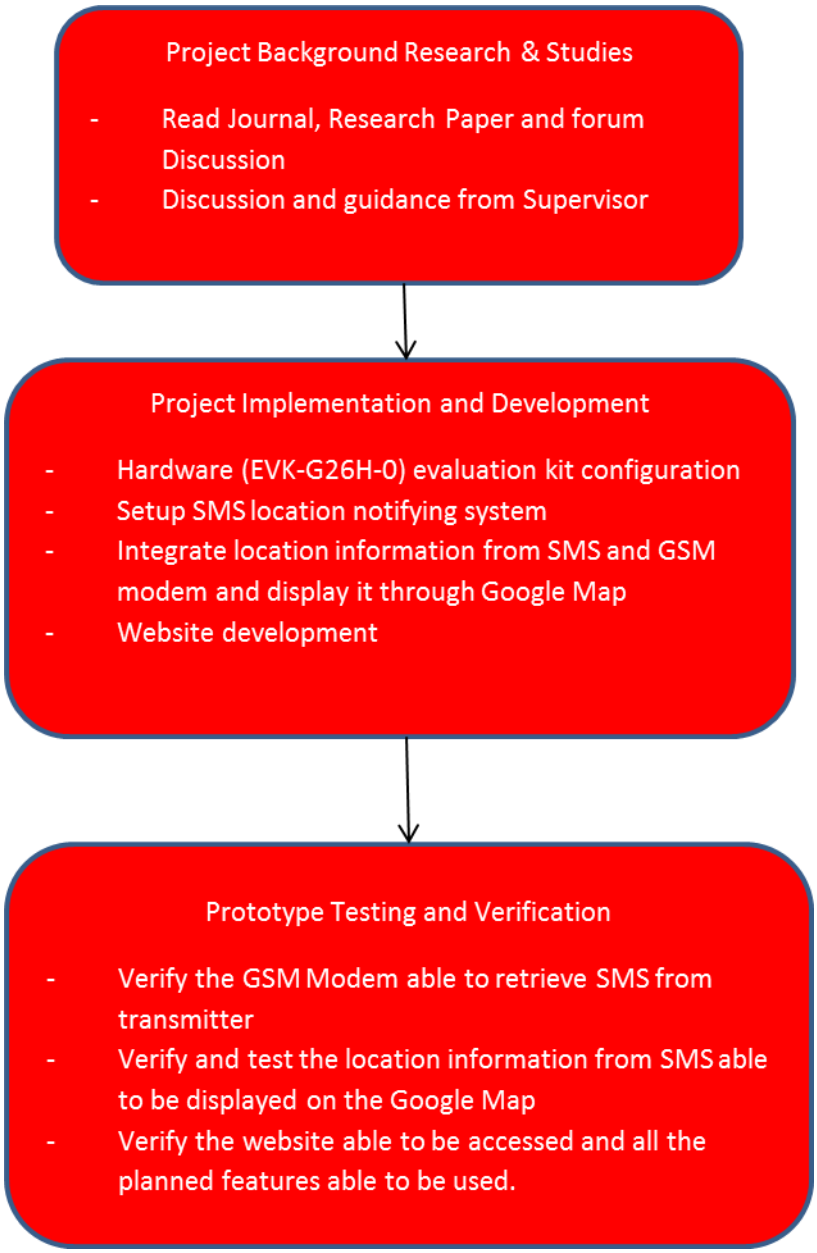


Figure 11: Project Planning Flow Chart

3.1.2 System Architecture Diagram

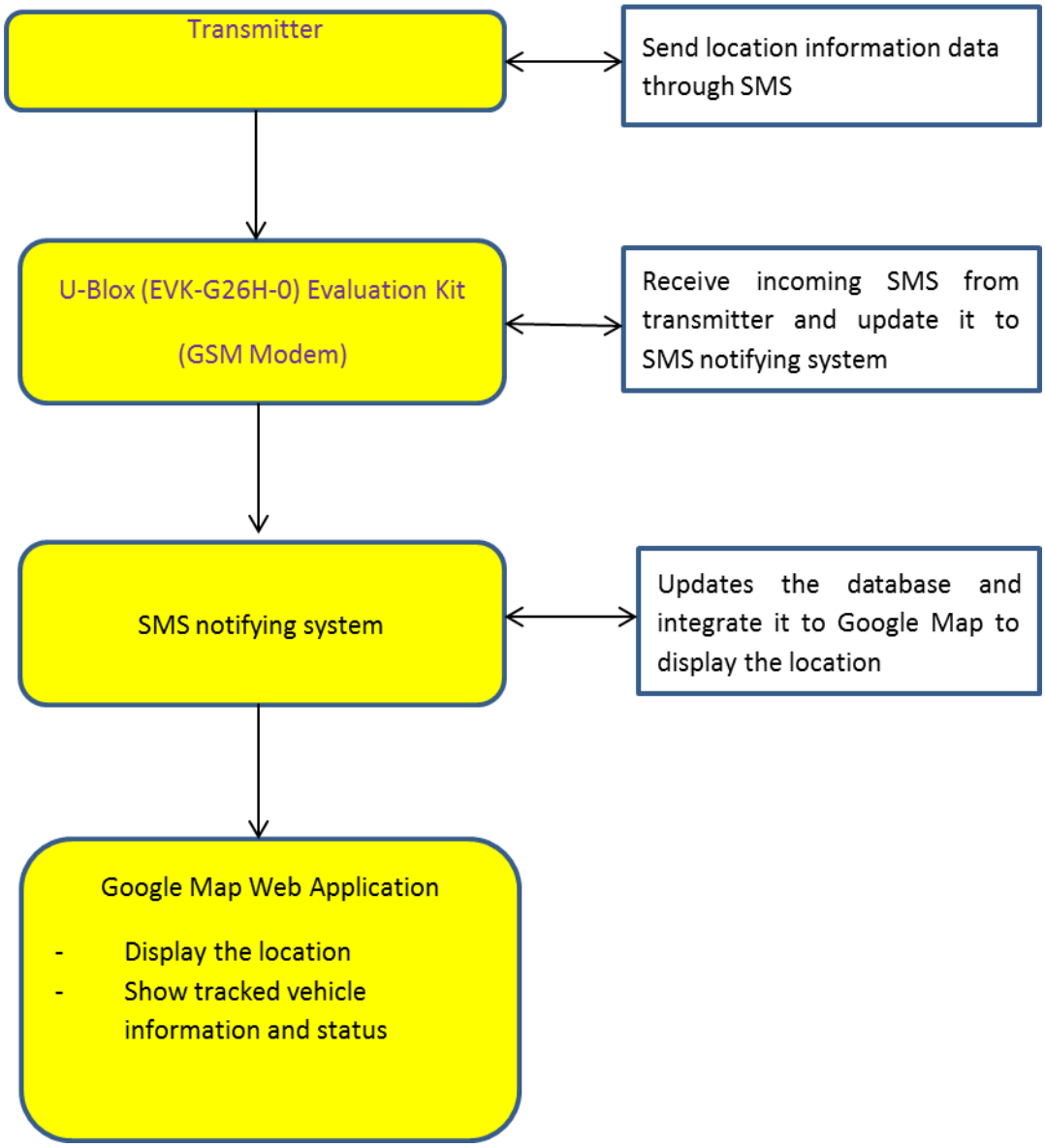
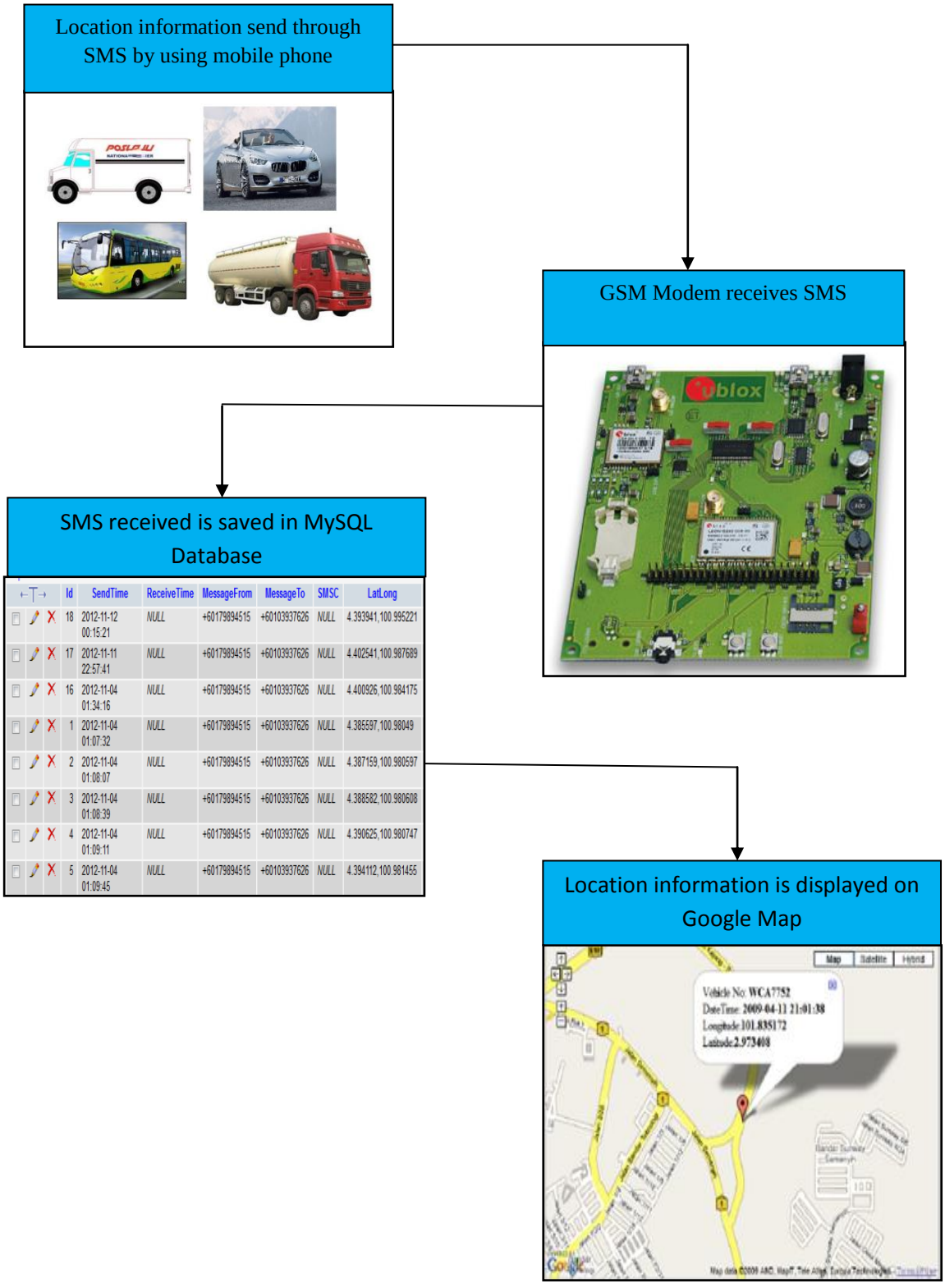


Figure 12: System Architecture Diagram

3.1.2 System Block Diagram



Id	SendTime	ReceiveTime	MessageFrom	MessageTo	SMSC	LatLong
18	2012-11-12 00:15:21	NULL	+60179894515	+60103937626	NULL	4.393941,100.995221
17	2012-11-11 22:57:41	NULL	+60179894515	+60103937626	NULL	4.402541,100.987689
16	2012-11-04 01:34:16	NULL	+60179894515	+60103937626	NULL	4.400926,100.984175
1	2012-11-04 01:07:32	NULL	+60179894515	+60103937626	NULL	4.385597,100.98049
2	2012-11-04 01:08:07	NULL	+60179894515	+60103937626	NULL	4.387159,100.980597
3	2012-11-04 01:08:39	NULL	+60179894515	+60103937626	NULL	4.388582,100.980608
4	2012-11-04 01:09:11	NULL	+60179894515	+60103937626	NULL	4.390625,100.980747
5	2012-11-04 01:09:45	NULL	+60179894515	+60103937626	NULL	4.394112,100.981455

Figure 13: System Block Diagram

3.2 Project Activities

The time given to complete the project is almost full 2 semesters of studies which are approximately 8 months and it comprises of project background research, project implementation, project prototype verification and also reports writing. For the project activities, the time frames are divided into three which are early project development, mid project development and project verification and testing phase.

During early project development phase, everything will be focusing on the project background research and theoretical reading so that the author had more understanding regarding the project and how to implement and develop the project. Discussion with project supervisor also had given a better understanding and widen the view of the project scope. Dr. Micheal Drieberg also suggested extra features and function for the project and this had helped a lot for the project expansion.

Mid project development phase will be the most crucial and important part of project activities as it will put all the theoretical knowledge and information obtain and gathered and transform it into practical work. The practical work start with the GSM modem configuration and set up. The GSM modem is already provided by the project supervisor. Next step is to figure out how to retrieve location information from transmitter (through SMS) to the GSM modem. After that, the author need to find a way to update the location information from the SMS received and display the location on the Google Map. This will involve with PHP programming as it will embedded the Google Map API to the website. Beside display the location of tracked vehicle, the author also planned to implement some extra features including the time of location recorded, the distance travelled, type and registration number of vehicles and also vehicle route prediction. All of this information will be available and will be able to be viewed online. Project prototype verification and testing phase will be the period for the author to try the system either it meet the project objectives and working properly as wanted. After verified that the project prototype is successful and achieve the project objectives, a full project report will be written and the project will be open for future expansion and modification.

3.2.1 Problems during Project Development

The author had managed to setup the GSM Modem and also had written a source code for a basic Google Map API application which is display the location from given latitude and longitude. The problem that the author faced is, unable to extract the SMS received from the m-center. The author had forward the problems to the technical advisor of the product and the response from the technical advisor is the m-center won't allow interface to other software and he suggest the author to write a program to allow the interface occurred which required a higher knowledge of programming language. The only source that the author could use is the terminal log record by the software which is saved in the installation folder. The terminal log consist all the command received and send by the m-center software including the SMS receive. The terminal log is in the .txt format which is in the notepad form. Figure below shows screenshot of the terminal log in .txt format

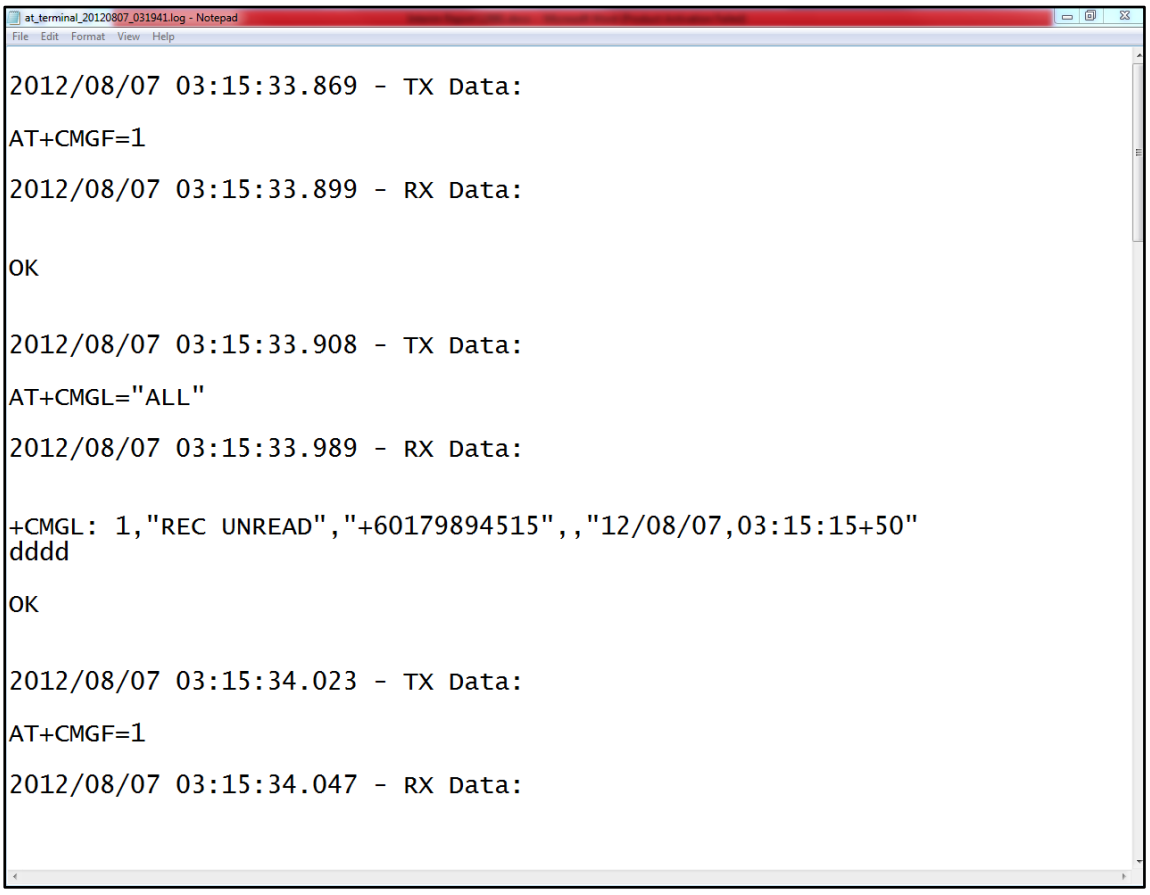


Figure 13: M-center Terminal Log

The author need to extract the information required from the terminal log which is the SMS received and sent it to the database in order for the Google Map API able to display the real time location base on the latitude and longitude information received by the SMS. To doing so, the author need to use or write a specific program to detect the string or information required from the terminal log. Judging from the terminal log above, the log is displaying all the commands received and sent by the terminal.

The author had tried to use third party software to overcome the problems which is AutoIt.

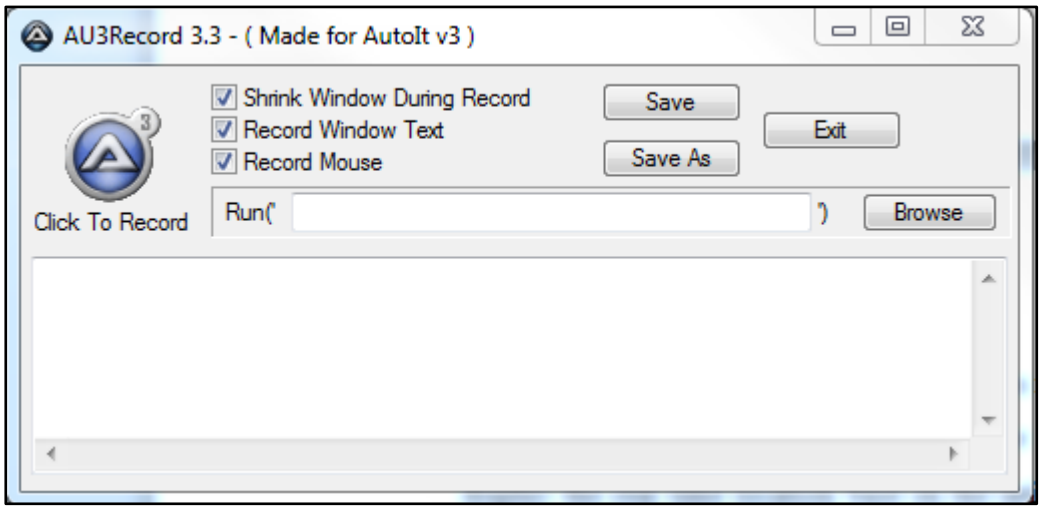


Figure 14: AutoIt software

The AutoIt software used to record the movement of mouse and keyboard input in window. The author intends to use the software to copy the location information send by the SMS and paste it to the database. But it involve too complex mouse movement and keyboard input to do so which is unable to recorded by the AutoIt software. To enable the features, it may need a window scripting language to automate the command needed.

3.3 Key Milestone

Below are the key milestone that need to be achieve by the author throughout both of the semester of Final Year Project 1 (FYP I) and Final Year Project 2 (FYP II) which will be the total of 26 weeks. There are some milestone that already successfully completed.

Milestone	**Week**
Project background research and discussion with supervisor	COMPLETED
Project Implementation - GSM Modem (EVK-G26H) setup and configuration - SMS System Setup - Integrate the SMS location information with the Google Map - Website Development	 COMPLETED COMPLETED COMPLETED COMPLETED
Prototype verification and troubleshooting	COMPLETED

Table 1: Key Milestone of Project

3.4 Gantt Chart

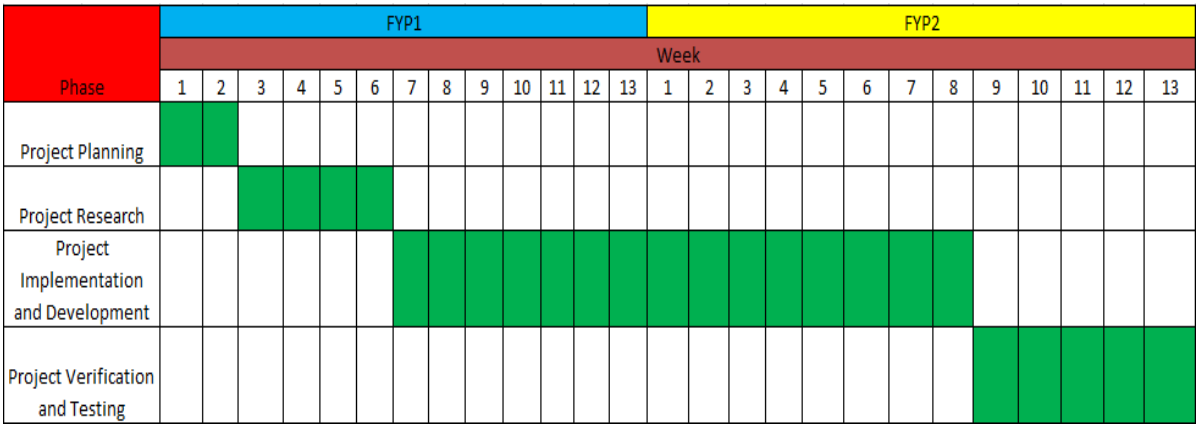


Table 2: Project Gantt Chart

3.5 Tools and Equipment

To complete and execute this project, it required some tools and equipment which are mostly software. Below are the list of software and tools required.

1. Transmitter (To simulate the system and send location information through SMS)

   For this project, a mobile phone will act and simulate as transmitter to transmit the location coordinate in a (latitude, longitude) format via SMS

2. GSM Modem (U-Blox (EVK-G26H-0) Evaluation Kit)

   The GSM Modem will act as receiver to receive the SMS from the transmitter (mobile phone). A normal broadband dongle could be use as GSM Modem as well.

3. Macromedia Dreamweaver Software

   Software for PHP and JavaScript scripting. Feature such as design view in this software is essential for author to check and design the layout of the website before execute and host it online.

4. SMS System Software (U-Blox : M-Center)

   The software is provided together with the EVK-G26H-0 Evaluation Kit and act as SMS Center for the modem. The disadvantage of the software is the SMS received by the software faces difficulty to integrate with MySQL database.

5. Diafaan SMS Server

   Alternative solution for M-Center, Diafaan SMS Server offers connectivity between SMS Server and MySQL Database. For development and theoretical usage it gives great aid to the author.

6. Wampserver

   Wampserver use to host the PHP and HTML webpage via localhost and good for experimental webpage before hosting it online.

CHAPTER 4: RESULT AND DISCUSSION

4.1 Data Gathering & Analysis

4.1.1 Integration difficulty between SMS Server with MySQL Database

Quoting from previous discussion, the author faced difficulty to integrate the SMS received from the transmitter to the preferred database (MySQL). The integration between the SMS Server and database is very important as the Google Map code developed by the author need to fetch the location information (latitude and longitude) only through a database such as Microsoft Access or MySQL. After discussion with project supervisor, several solutions and options are proposed such as below:

1. Developed a dummy MySQL Database to test the Google Map code
2. Find alternative solution to integrate the SMS Server with Database
   - Use MATLAB for input output extraction and export
   - Use third party software to replace the U-Blox (M-Center)

4.1.2 Option 1: Developed a dummy MySQL Database

Due to time constraint, a dummy database had been developed for theoretically purpose of project implementation and development.

log_ID	UnitID	Latitude	Longitude	Type_of_Vehicle	Vehicle_ Registration_Number	DateTime
6	0001	4.413923	100.986959	Perodua Kancil	ST6748K	2012-10-01 22:31:11
5	0001	4.411452	100.986364	Perodua Kancil	ST6748K	2012-10-01 22:30:28
4	0001	4.403408	100.984948	Perodua Kancil	ST6748K	2012-10-01 22:29:51
3	0001	4.398573	100.982974	Perodua Kancil	ST6748K	2012-10-01 22:28:58
2	0001	4.394251	100.980914	Perodua Kancil	ST6748K	2012-10-01 22:27:05
1	0001	4.388175	100.979261	Perodua Kancil	ST6748K	2012-10-01 22:26:18
7	0001	4.415913	100.987603	Perodua Kancil	ST6748K	2012-10-01 22:31:51
8	0001	4.417763	100.987206	Perodua Kancil	ST6748K	2012-10-01 22:32:32
9	0001	4.419293	100.987013	Perodua Kancil	ST6748K	2012-10-01 22:33:15
10	0001	4.42124	100.986831	Perodua Kancil	ST6748K	2012-10-01 22:33:46
11	NULL	4.3333323,2534444	NULL	NULL	NULL	NULL

Check All / Uncheck All With selected:

Figure 15: Dummy MySQL Database

Figure 15 show a screen-shot of Dummy MySQL Database created by the author. There are three important and core information used from this database which are the

Latitude, Longitude, and DateTime. The Latitude and Longitude will act as variable and will be display as a location point in the Google Map. The DateTime is used to determine and differentiate the latest Latitude and Longitude data selection from the table.

After developed the dummy database, the author firstly try to extract the required and demanded information from the database to the webpage.

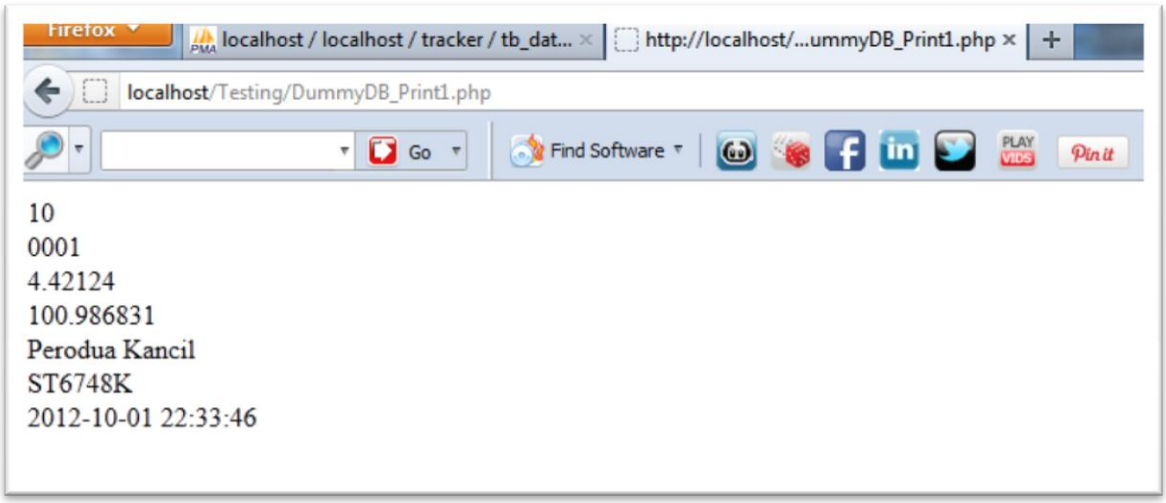


Figure 16: The latest information print out at webpage

Above figure shows that author managed to extract the latest data from the table and displayed it on the webpage. The next step is to transform the data and information obtains to a Google Map with latest location coordinate.

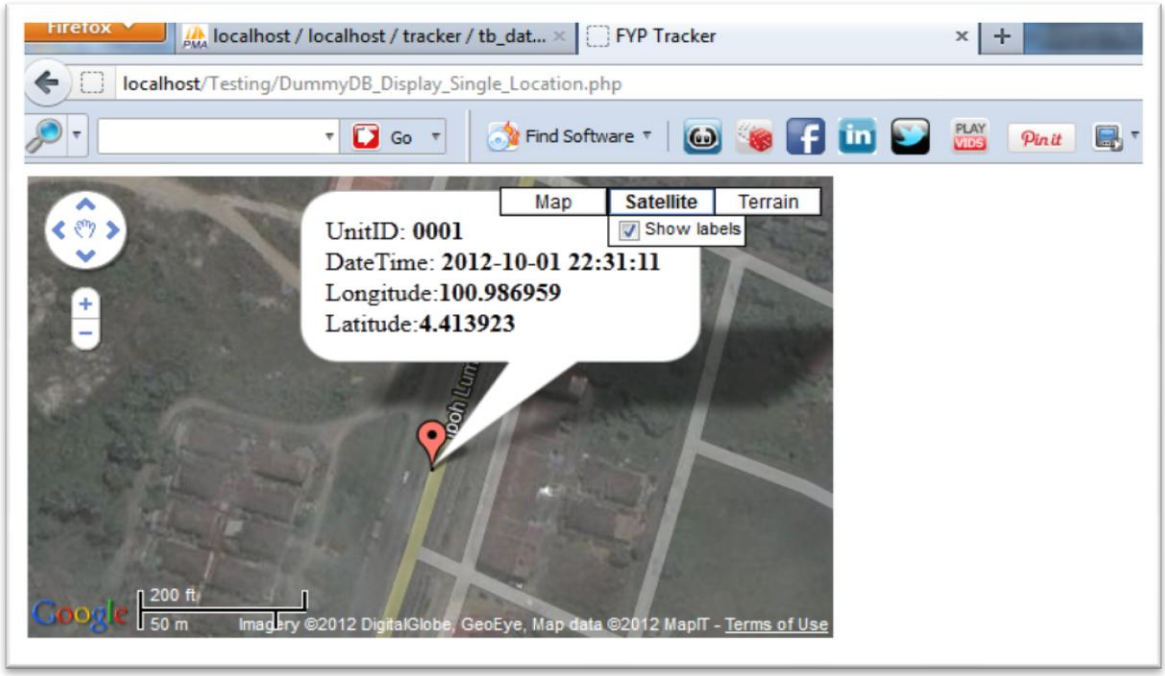


Figure 17: First Function (Track Latest Location)

Above figure shows that the author managed to present the data obtained from the dummy database in graphical way (Google Map) instead of print out the information in webpage. It is one of the basic functions for this project. The next function is to display 5 latest locations and present it as a route.

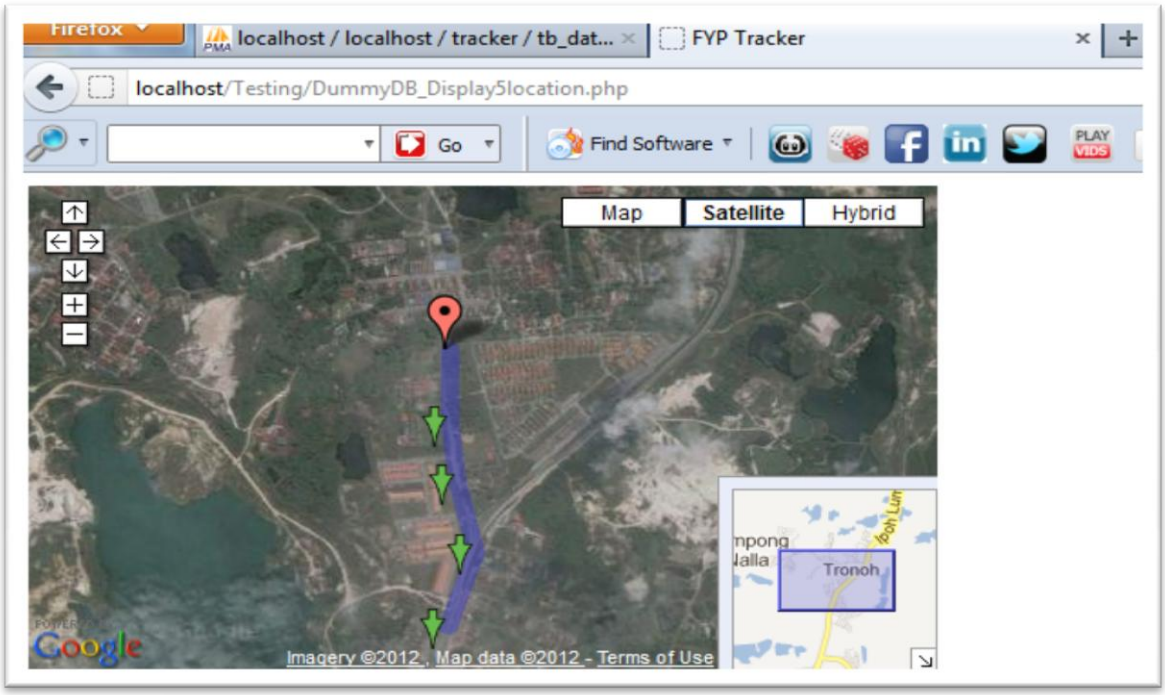


Figure 18: Second Function (Display 5 Latest Locations as Route)

By using the dummy database, the author managed to test and try his Google Map code and script and successfully developed two basic functions for this project. The author then decided to find a solution to overcome the integration problem between the SMS Server and the Database. The author's project supervisor advised and suggested the author to use MATLAB to overcome the problems.

4.1.3 Option 2: Use MATLAB to overcome the integration problems between M-Center and MySQL Database

MATLAB is a powerful tool that enables the users to do a lot of things including manipulating file, extract required information from notepad, data exports and many more. The author had tried to extract the required information for the M-Center Log.

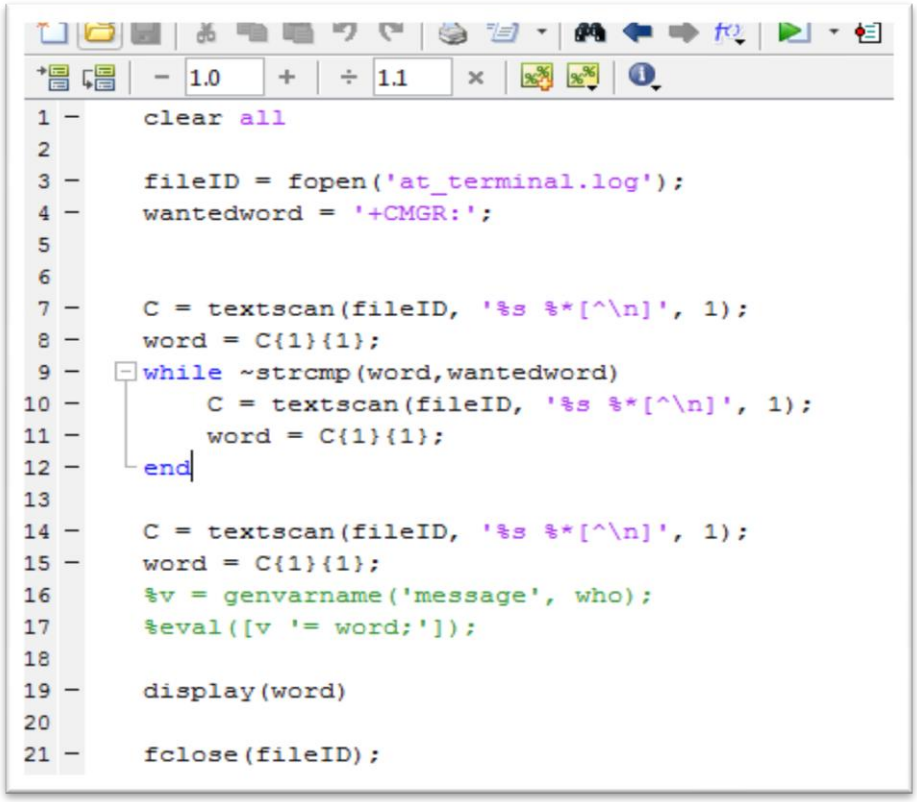


Figure 19: MATLAB m-file code

The code use to filter the text in the at_terminal.log (M-Center Log) in order to obtain and retrieve desired information. Unfortunately above code only select the upper top information in the log file only.

Figure 20: MATLAB m-file code result

The at_terminal.log sample is as below. Noted that the MATLAB m-file code only able to select the top information only.

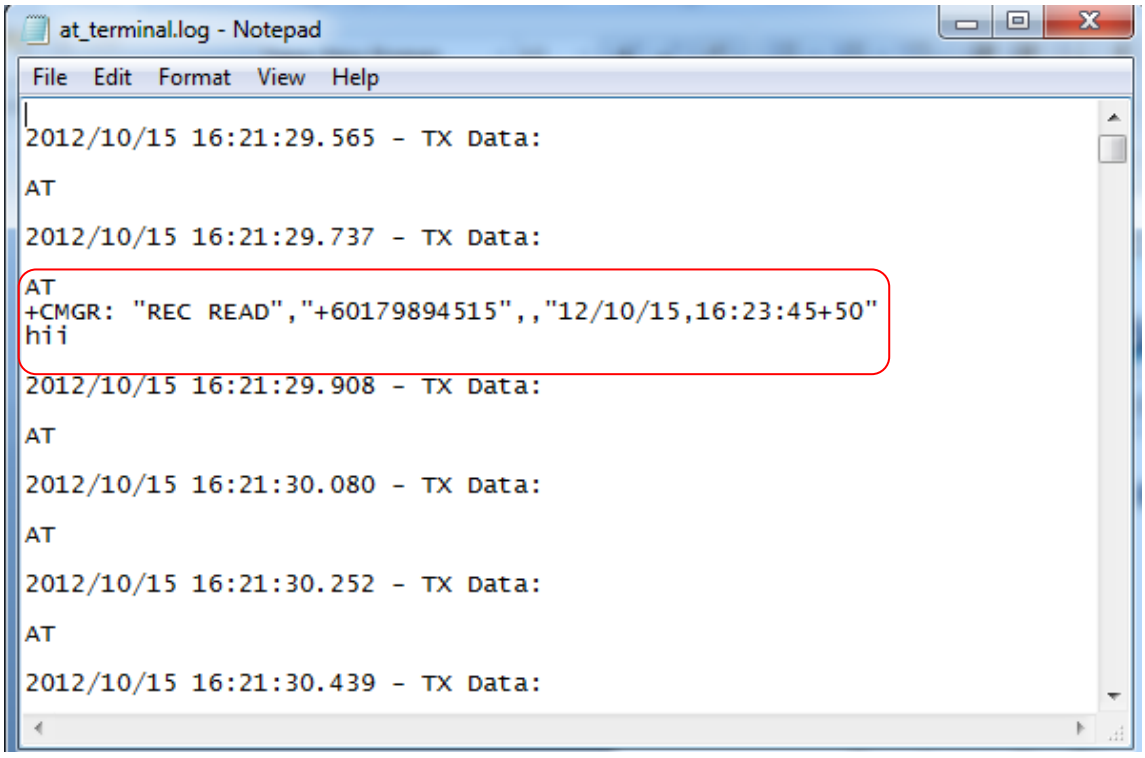


Figure 21: at_terminal.log

By using MATLAB it is possible to develop a code or program to integrate the SMS Server with the MySQL database if given and provided with sufficient time to implement and develop it. The author had tried to use MATLAB to solve the problem for almost two weeks but failed to extract the required information as too many criteria and conditions need to be considered during extraction processes. The

author then proceeds to use another alternative to overcome the integration problem between SMS Server and database.

4.1.4 Option 3: Diafaan SMS Server

Diafaan SMS Server is Window SMS gateway program for 3G/GSM modems. It is easy to set up with online manual and forum support. For this project, Diafaan SMS Server could be use as an alternative to replace the U-Blox (M-Center) SMS Server as Diafaan SMS Server offer great connectivity with MySQL and Microsoft Access Database.

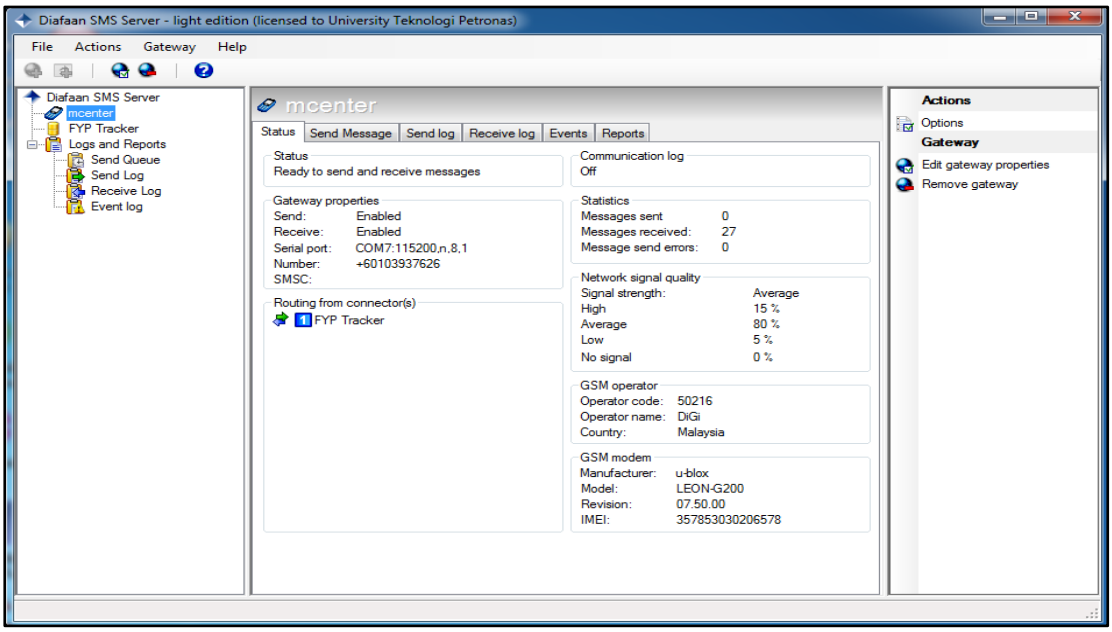


Figure 22: Diafaan SMS Server Interface

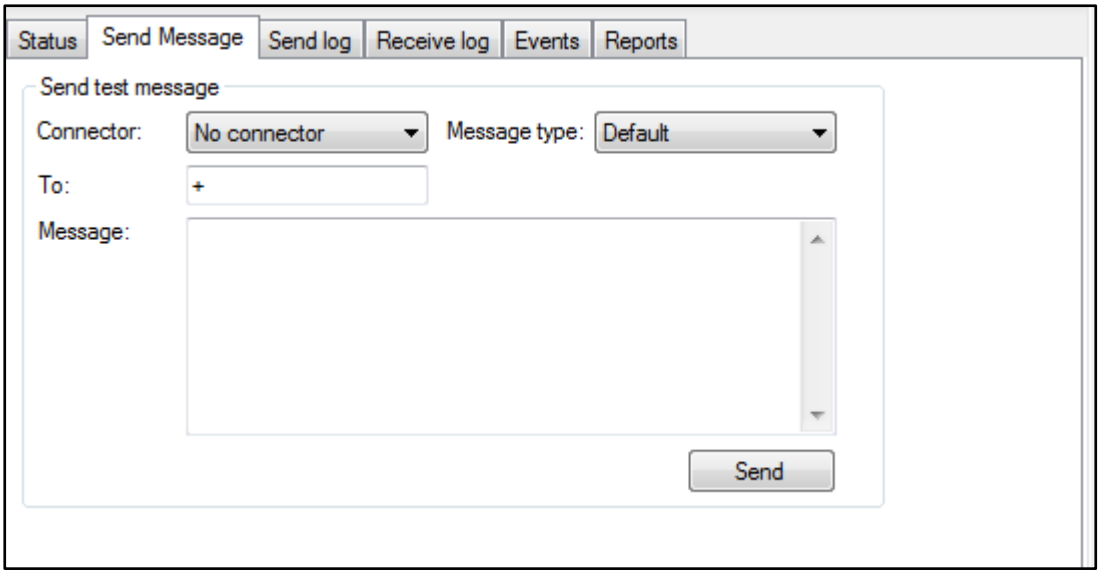


Figure 23: Diafaan SMS Server Send Message Function

Figure 23 shows that Diafaan SMS Server also could be use to send message which also offers in U-Blox M-Center SMS Server. What makes Diafaan SMS Server could be an alternative solution for the integration problem between SMS Server with database is, it has the 'connector' function to connect the SMS Server with preferred database such as MySQL and Microsoft Access.

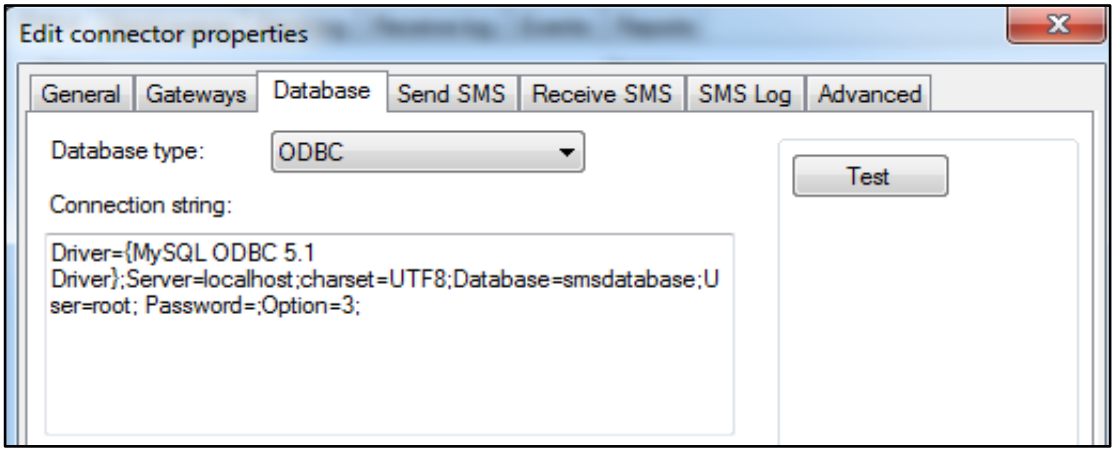


Figure 24: Connector String to Connect SMS Server with MySQL Database

Figure 24 show the string that enables the SMS Server to link with the MySQL database. This function had solved the integration problem between the SMS Server and database.

A new database had been created to replace the previous dummy database.

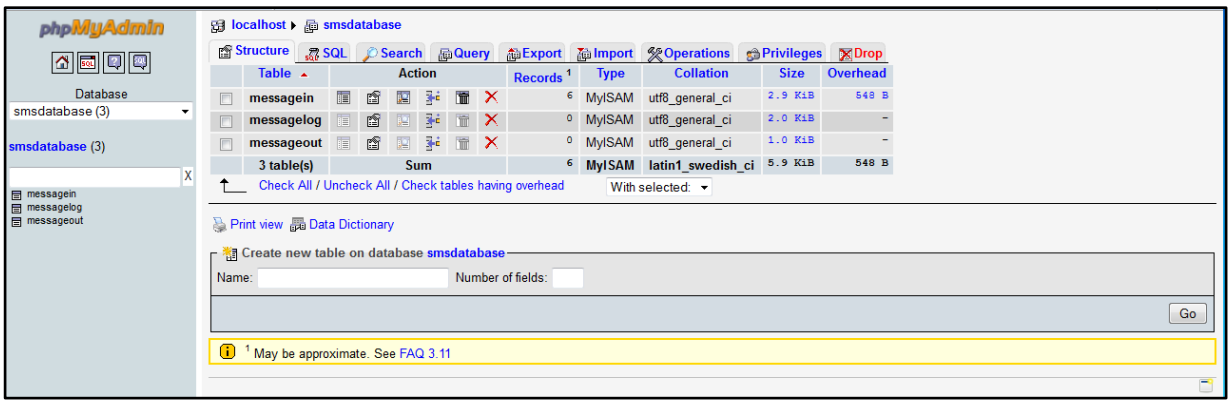
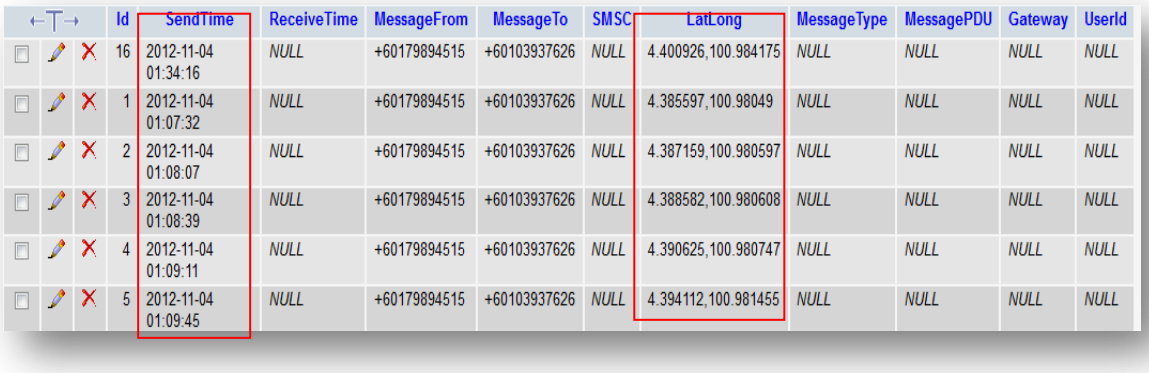


Figure 25: New Database

Noted that the new database contain 3 tables which are: messagein (message inbox), messageout (message outbox) and messagelog. From the three tables, only messagein table are needed and give priority to the project as all the required data input is come from the table only.



Id	SendTime	ReceiveTime	MessageFrom	MessageTo	SMSC	LatLong	MessageType	MessagePDU	Gateway	UserId
16	2012-11-04 01:34:16	NULL	+60179894515	+60103937626	NULL	4.400926,100.984175	NULL	NULL	NULL	NULL
1	2012-11-04 01:07:32	NULL	+60179894515	+60103937626	NULL	4.385597,100.98049	NULL	NULL	NULL	NULL
2	2012-11-04 01:08:07	NULL	+60179894515	+60103937626	NULL	4.387159,100.980597	NULL	NULL	NULL	NULL
3	2012-11-04 01:08:39	NULL	+60179894515	+60103937626	NULL	4.388582,100.980608	NULL	NULL	NULL	NULL
4	2012-11-04 01:09:11	NULL	+60179894515	+60103937626	NULL	4.390625,100.980747	NULL	NULL	NULL	NULL
5	2012-11-04 01:09:45	NULL	+60179894515	+60103937626	NULL	4.394112,100.981455	NULL	NULL	NULL	NULL

Figure 26: Screenshot of "messagein" table

There are two columns of data and information is taken as input for the Google Map code and for the project which are the "SendTime" and "LatLong". The "SendTime" will determine which row of information is the latest updated. While the "LatLong" represent the coordinate of location in a format of (latitude, longitude). All the coordinate sent by the transmitter able to immediately update into the database.

4.2 Experimentation/Modeling

4.2.1 Defining Variables for Latitude and Longitude

With real time update of database, the Google Map code now able to update the latest location of tracked vehicle almost instantaneously. The author then proceeds to develop the Google Map function step by step. Some tweaks and changes of Google Map code are needed compare from previous dummy database.

	log_ID	UnitID	Latitude	Longitude	Type_of_Vehicle
	6	0001	4.413923	100.986959	Perodua Kancil
	5	0001	4.411452	100.986364	Perodua Kancil
	4	0001	4.403408	100.984948	Perodua Kancil

Figure 27: Snapshot of Dummy Database

Noted that from Dummy Database, the location coordinate came from two columns of information which are the “Latitude” and “Longitude” column. While from the real time updated database the information only extracted from single column which are “LatLong” as shown in Figure 28.

	Id	SendTime	ReceiveTime	MessageFrom	MessageTo	SMSC	LatLong
	16	2012-11-04 01:34:16	*NULL*	+60179894515	+60103937626	*NULL*	4.400926,100.984175
	1	2012-11-04 01:07:32	*NULL*	+60179894515	+60103937626	*NULL*	4.385597,100.98049
	2	2012-11-04 01:08:07	*NULL*	+60179894515	+60103937626	*NULL*	4.387159,100.980597

Figure 28: Snapshot of Real Time Update Database

The Google Map code needs to change and modify in order to read and interpret the “LatLong” as “Latitude” and “Longitude”. The reason the coordinate information needed to put under one column is due to the one full location coordinate sent per SMS compare to the Dummy Database which can be edited easily. Figure 29 shows the php code how the single input “LatLong” is split into two variables which are Latitude and Longitude.

```
$id = $_POST["id"];
$SQL = "SELECT * FROM messagein ORDER BY SendTime DESC LIMIT $id";
$result = mysql_query($SQL);
while ($data = mysql_fetch_array($result))
    {
        print $data['LatLong'] . " <br>";
        print $data['SendTime'] . " <br>";
        $latlong = $data['LatLong'];
        list($latitude[1], $longitude[1]) = explode(",", "$latlong", 2);


        print $latitude[1]   . " <br>";
        print $longitude[1]   . " <br>";
```

Figure 29: Snapshot of php code

The highlighted red is the command to split the "LatLong" information into two variables which are "Latitude" and "Longitude". The result of above code is as in Figure 30.

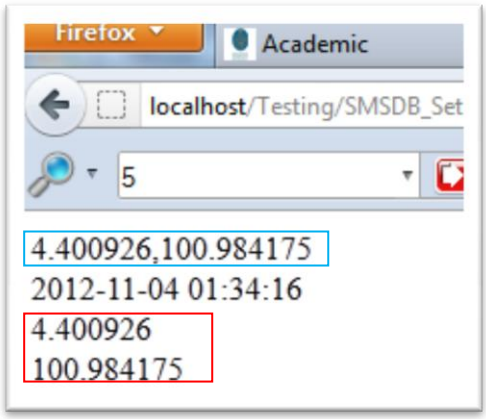


Figure 30: php code result (1)

The results show that the "LatLong" input (highlighted blue) had successfully split into two new variables which are "Latitude" and "Longitude" (highlighted red). After able to extract the required information and input from the database, the next task is to pass the input to the Google Map JavaScript. In order to do so, the variables in php need to re-declare in the JavaScript. The declaration code and program is as in Figure 31.

```
var count = "<?php echo $i ?>";
var UnitID = "0001" ;
lat = new Array();
long = new Array();
datetime = new Array();

<?php
    for ($z=0; $z<$i; $z++)
    {
        print "lat['$z']='$lat[$z]';";
        print "long['$z']='$long[$z]';";
        print "datetime['$z']='$datetime[$z]';";
    }
?>
```

Figure 31: Variable declaration in JavaScript (html)

The lat "latitude", long "longitude", and datetime need to declared as array as the input from the database could be more than one (depending on user input , example : User would like to display 5 latest location, thus the input of latitude, longitude, and datetime will have 5 different variables).

4.2.2 Transform the Latitude, Longitude and DateTime into Google Map

After successfully extract the required information from database and parse into php and html (JavaScript) code, the next task is to "transform" those data and information into more graphical way which is in Google Map format. The code and program (method) to transform those information into Google Map will be the same as previous Dummy Database. For time being, there are three functions successfully be implemented and mentioned as below:

First Function: Track Vehicle Latest Position

This is simple function which selects the "LatLong" information base on the most updated "SendTime" from the "Real-Time Update Database". Figure 32 displays the function result.

Figure 32: Track Latest Position Example

Second Function: Track User Input Vehicle Latest Position

This is the expansion of the first function which shows the latest user input of vehicle latest position. For an example, if user demands 4 latest position of the tracked vehicle, the user just need to key in 4 and the Google Map will display 4 latest position of the vehicle.

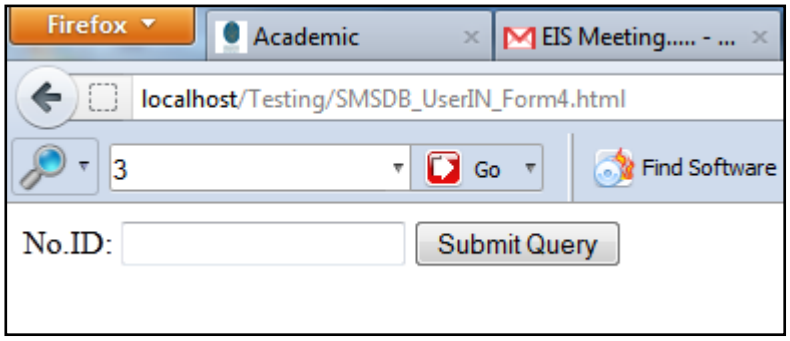


Figure 33: Form for user to fill in value of required latest location

Figure 34 shows the result after user fills the form with desired value. As in case below, the user input value will be 4 as it shows 4 latest positions in the map. Noted that if the user keys in values exceed the amount of data inside the database, the values will be return to the maximum number of data inside the database.

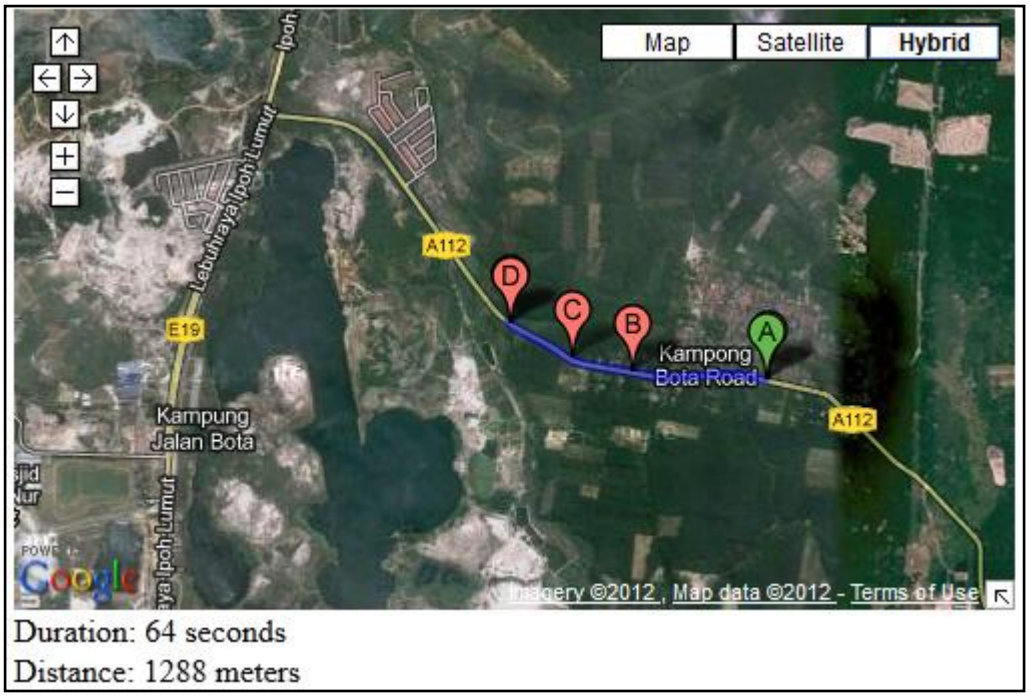


Figure 34: Track Latest Position (User Input)

Third Function: Route and Destination Planner Base On Current Location

The third function is to plan and simulate user preferred destination or place of interest base on user current latest position. The function almost similar with the Google Map offered Point-To-Point route planner but this function could animate the route travelled.

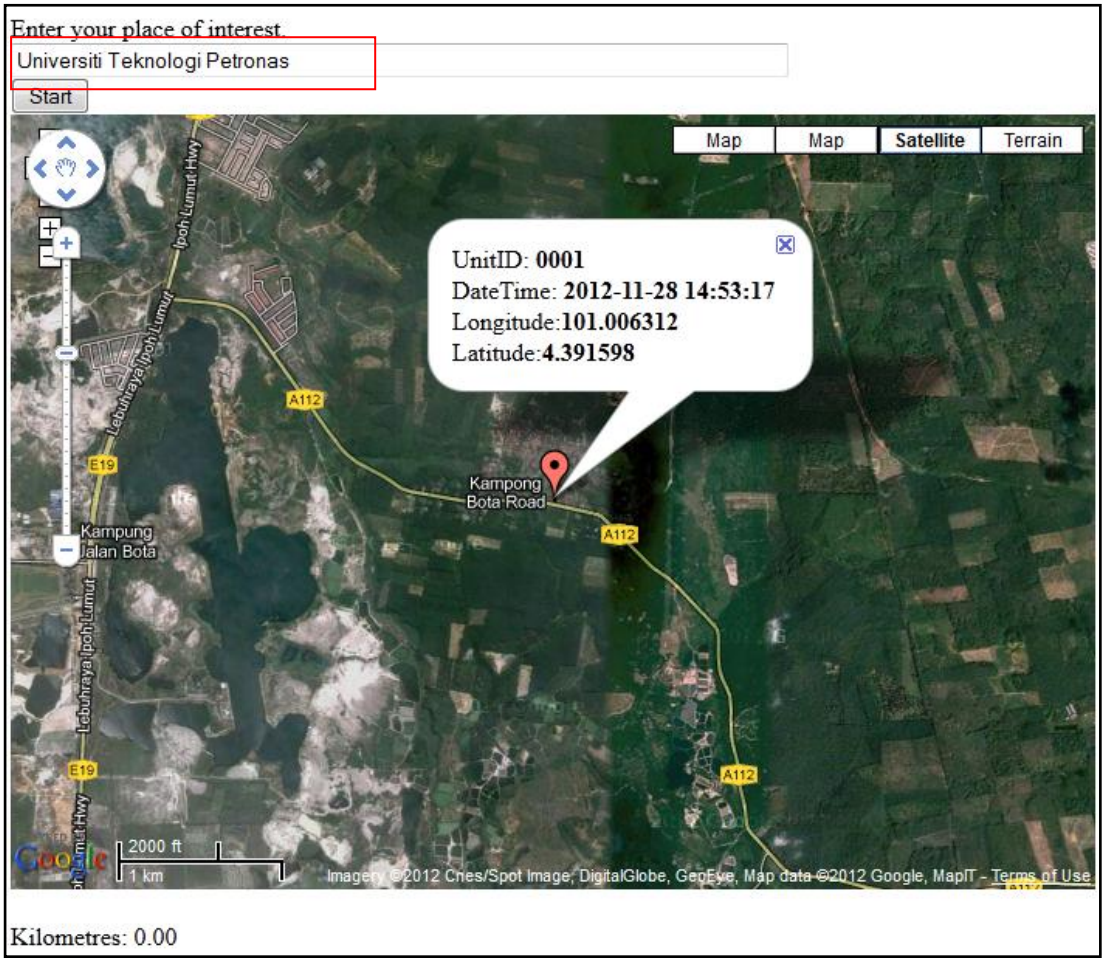


Figure 35: Route and Destination Planner (current location)

Figure 35 shows the current location of tracked vehicle. User could key in any preferred location (as shown in highlighted red box). The author use function provided by Google to convert the user input key in addresses or place of interest into set of coordinate (latitude,longitude) the function is called 'Geocoding'. Figure 36 and 37 shows the result of the third function after users key in their preferred location. (Example : Universiti Teknologi PETRONAS)

Figure 36: Route and Destination Planner(vehicle on moving)

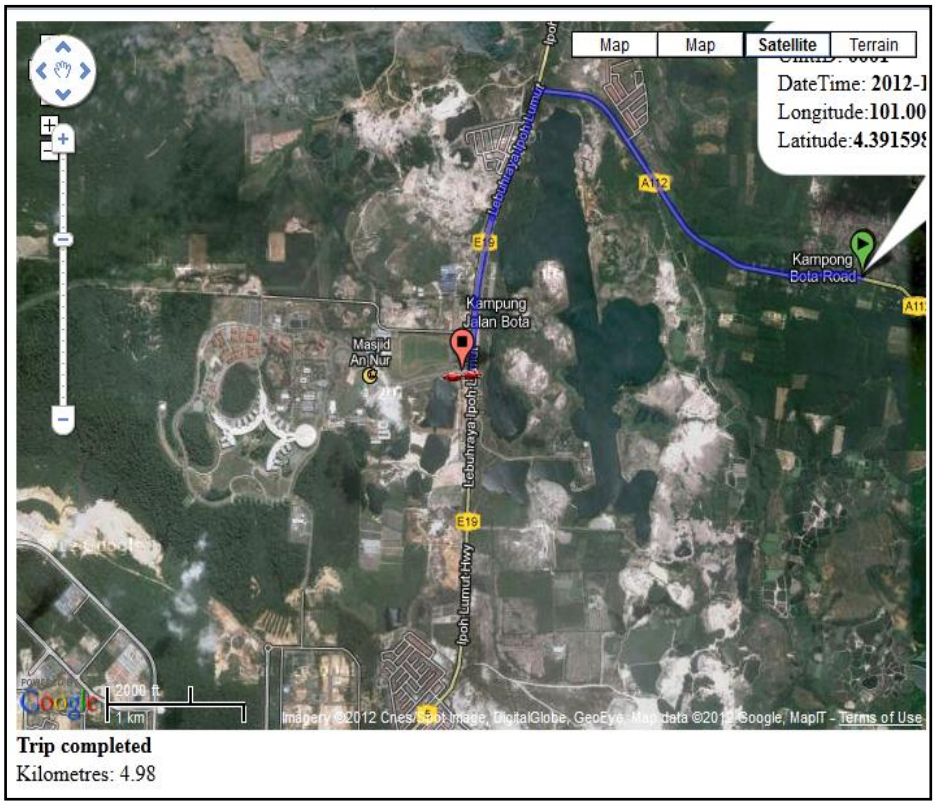


Figure 37: Route and Destination Planner (vehicle reached destination)

4.3 Prototype

The author had successfully extracts the required information from database and transform those information into Google Map. The prototype of the project will be in a well displayed webpage format. The author used existing website template provided online and modify it to suite the project purpose. Figure 38 shows the webpage template.

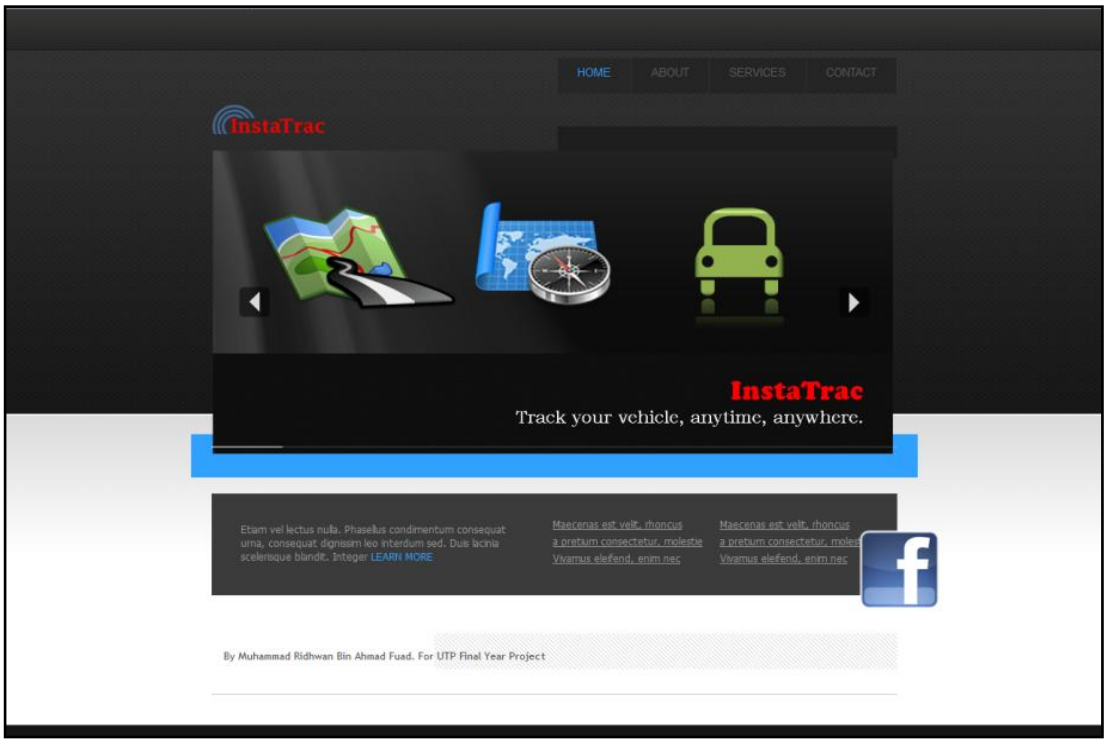


Figure 38: Example of website template

Above figure shows the front page of the prototype. All of the tracking features will be located in Services section as in Figure 39.

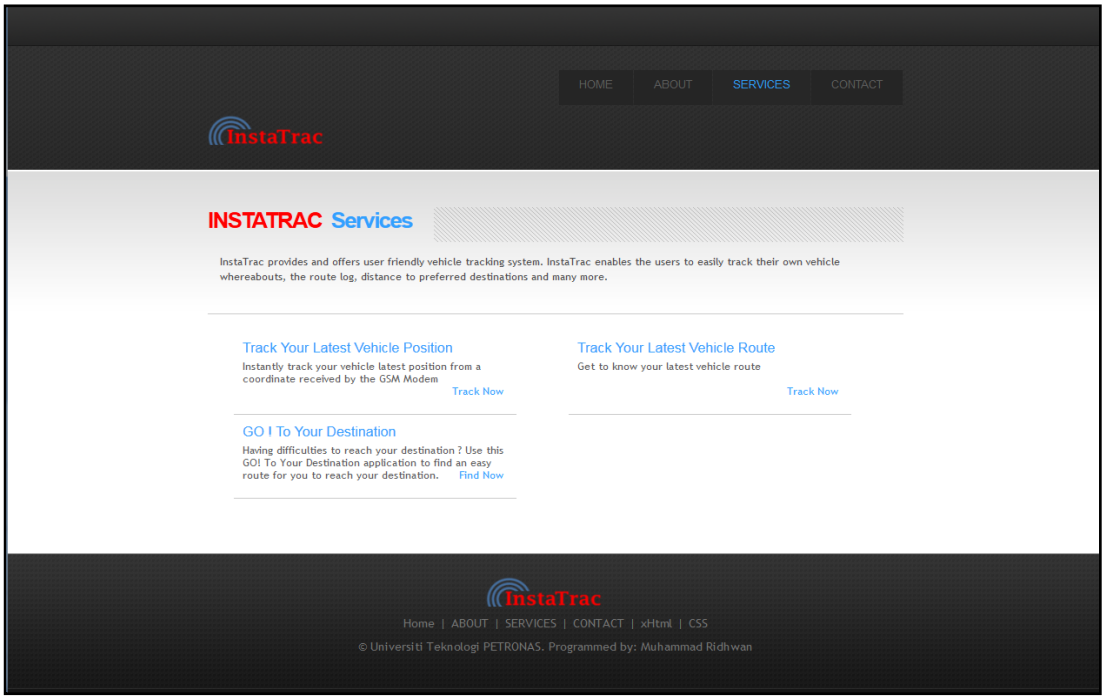


Figure 39: Prototype Services Page

In the Services page, it will display all the three available function. Which are Tracked Current Latest Vehicle Position, Track Latest Vehicle Route, Route and Destination Planner as shown in Figure 40,41 and 42.

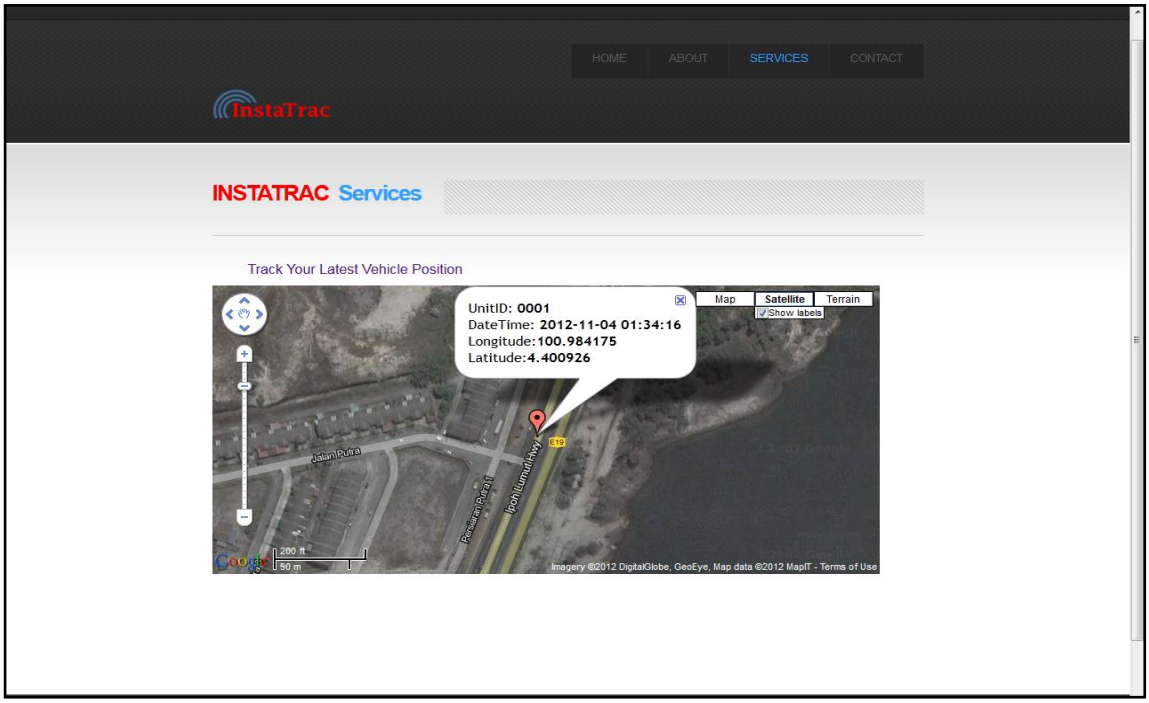


Figure 40: Track Latest Vehicle Position (Embedded into Website)

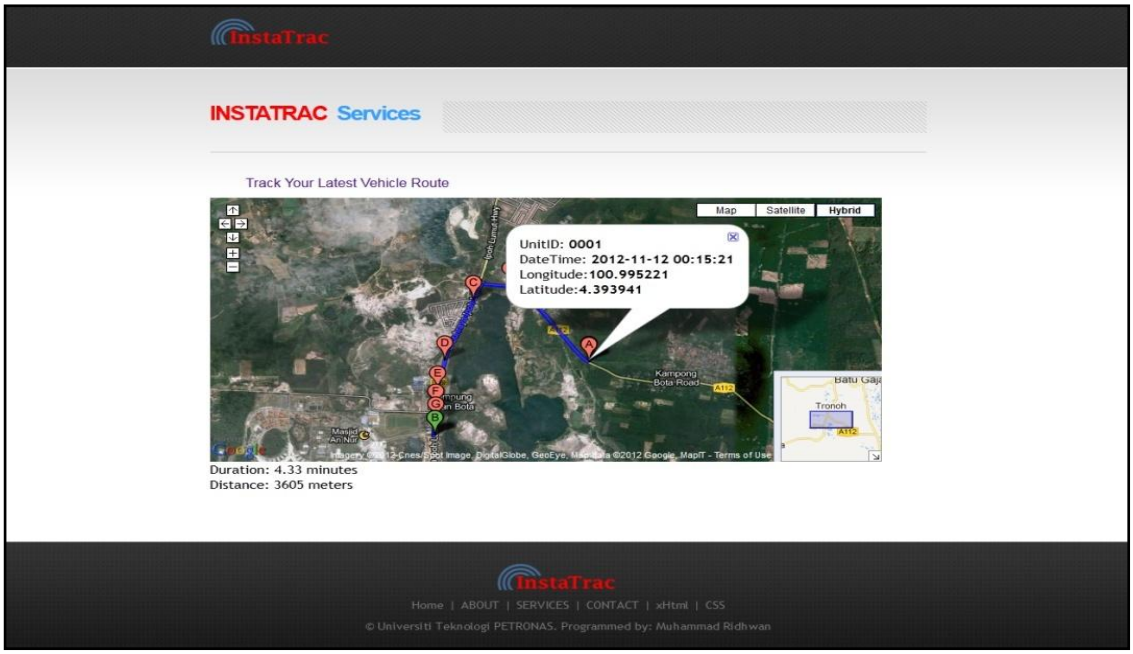


Figure 41: Track Latest Vehicle Route (Embedded into Website)

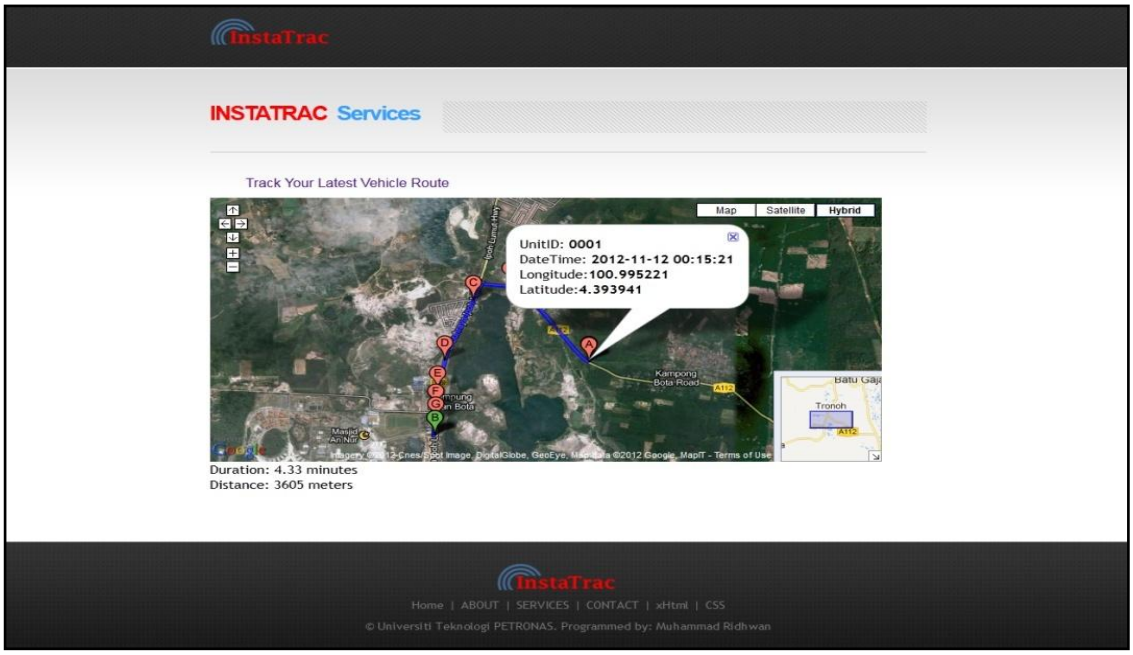


Figure 42: Route and Destination Planner (Embedded into Website)

More functions are planned to be developed, but due to time constraint only three functions are successfully working. Any other functions can be added and developed for future work.

CHAPTER 5: CONCLUSION & RECOMMENDATION

5.1 CONCLUSION

Remote Vehicle Tracking System not only benefits and improves the fleet management but also improve the security of the vehicle. The importance regarding the system need to be exposed to the public in order to increase their awareness to the system. The wide coverage of GSM Cellular Network in Malaysia also could be fully utilized if this system is implemented.

Google Map had become one of the most important applications in daily life. Before this, people only could use Google Map to locate or identify unknown or non-familiar places. After Google had introduced Google Map API, any interested developer could develop their own Google Map related applications including developing a Remote Vehicle Tracking System by using the Google Map.

This project successfully achieved the objective which is integrating the GSM Modem with the Google Map and display the tracked vehicle position. This project also proof that Remote Vehicle Tracking System can be simply developed for personal usage. The author plan to add and developed more significant function of the Vehicle Tracking System to the project, but due to time constraint only three working functions are able to be completed.

5.2 RECOMMENDATION

Some of the expansion of project that can be implemented in the future will be as below:

1. Conduct field test to verify the reliability of the system
2. Improve database management system to store and update the information required
3. Develop own program to integrate the SMS Server with Database

References

[1] Malaysia Maxis GSM/GPRS Coverage Area. 20 June, 2012 < http://www.mobilecomms-technology.com/projects/gprs_mal/gprs_mal8.html>

[2] Muruganandham ,P.R.Mukesh, 2010. *Real Time Web based Vehicle Tracking using GPS.*Research paper of World Academy of Science, Engineering and Technology 61, 2010

[3] *General Packet Radio Service*. 21 June, 2012 <http://en.wikipedia.org/wiki/General_Packet_radio_Service>

[4] *Theodore, 2002; Hernando & Perez-Fontan, 1999*

[5] *GSM For Dummies* (2011), 23 June, 2012, <http://gsmfordummies.com/architecture/arch.shtml>

[6] http://www.yasukawa.com/blog/archives/images/sim-card.jpg

[7] Abdul Mueed Khalid (2007). *Location Aware System Using Mobile Station in GSM Network*. 23 June, 2012 <http://dspace.fsktm.um.edu.my/handle/1812/614>

[8] *Base transceiver station* (2012, March 11), 23 June, 2012 <http://en.wikipedia.org/wiki/Base_station_subsystem#Base_station_controller>

[9] evk-g26h.pdf. EVK-G26H Evaluation Kit datasheet page 1

[10] *Google Map*, 24 June, 2012 < http://en.wikipedia.org/wiki/Google_Maps>

[11] *The world is your JavaScript-enabled oyster* (2005), 24 June , 2012

<http://googleblog.blogspot.com/2005/06/world-is-your-javascript-enabled_29.html>

[12] https://developers.google.com/maps/location-based-apps

[13] *PHP,* 24 June, 2012 <http://www.webopedia.com/TERM/P/PHP.html>

APPENDICES

Track Latest Vehicle Position Source Code

```
<?PHP

$user_name = "root";
$password = "";
$database = "smsdatabase";
$server = "localhost";

$db_handle=mysql_connect('localhost', 'root', '');
$db_found=mysql_select_db($database,$db_handle);

if($db_found)
{
    $SQL = "SELECT * FROM messagein ORDER BY SendTime DESC";
    $result = mysql_query($SQL);
    if ($data=mysql_fetch_array($result))
    {
        $latlong = $data['LatLong'];
        $datetime = $data['SendTime'];
        list($latitude, $longitude) = explode(",", "$latlong", 2);


    }
        mysql_close($db_handle);
}


?>
<!DOCTYPE html "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-
type" content="text/html; charset=utf-8"/>
    <title>FYP Tracker</title>
    <script src="http://maps.google.com/maps?file=api&v=2&key=AIzaS
yANwgDDObXoTZUOrD6PRZRXdVkMhD3t8ek&sensor=true_or_false"
            type="text/javascript"></script>
    <script type="text/javascript">

    var UnitID="0001";
    var datetime="<?php echo $datetime ?>";
    var lat="<?php echo $latitude ?>";
    var long="<?php echo $longitude ?>";
     function initialize()
    {
      if (GBrowserIsCompatible())
      {
        var map = new GMap2(document.getElementById("map_canvas"));

        map.setMapType (G_HYBRID_MAP);
        map.setCenter(new GLatLng(lat, long), 18);
        map.openInfoWindowHtml(map.getCenter(),("UnitID: <b>" + Uni
tID + "</b>" + "<br>DateTime: <b>" +
```

```
49.         datetime + "</b>" + "<br>Longitude:<b>" + long + "</b>" + "
    <br>Latitude:<b>" + lat + "</b>"));
50.         map.setUIToDefault();
51.         map.enableRotation();
52.
53.         var baseIcon = new GIcon();
54.         baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.
    png";
55.         baseIcon.iconSize = new GSize(20, 34);
56.         baseIcon.shadowSize = new GSize(37, 34);
57.         baseIcon.iconAnchor = new GPoint(9, 34);
58.         baseIcon.infoWindowAnchor = new GPoint(13, 30);
59.         baseIcon.infoShadowAnchor = new GPoint(18, 25);
60.
61.         function createMarker(point, index)
62.         {
63.             var letteredIcon = new GIcon(baseIcon);
64.             letteredIcon.image = "http://www.google.com/mapfiles/ma
    rker.png";
65.             markerOptions = { icon:letteredIcon };
66.             var marker = new GMarker(point, markerOptions);
67.             GEvent.addListener(marker, "click", function() {
68.             marker.openInfoWindowHtml("UnitID: <b>" + UnitID + "</b
    >" + "<br>DateTime: <b>" + datetime +
69.             "</b>" + "<br>Longitude:<b>" + long + "</b>" + "<br>Lat
    itude:<b>" + lat + "</b>");
70.             });
71.             return marker;
72.         }
73.             var latlng = new GLatLng(lat,long);
74.             map.addOverlay(createMarker(latlng, 0));
75.       }
76.
77.     }
78.
79.
80.     </script>
81.   </head>
82.   <body onload="initialize()" onunload="GUnload()">
83.     <div id="map_canvas" style="width: 750px; height: 500px"></div>

84.
85.   </body>
86. </html>
```

Track Latest Vehicle Route

(User Input Form)

```
1. <html>
2. <body>
3.
4. <form action="SMSDB_DisplayUserInput - Copy -
    Copy.php" method="post">
5. No.ID: <input type="text" name="id" />
6. <input type="submit" />
7. </form>
8.
9.
10.
11.</body>
12.</html>
```

(Generate Route Function)

```
1. <?PHP
2.
3. $user_name = "root";
4. $password = "";
5. $database = "smsdatabase";
6. $server = "localhost";
7.
8. $db_handle=mysql_connect('localhost', 'root', '');
9. $db_found=mysql_select_db($database,$db_handle);
10.
11.if($db_found)
12.    {
13.        $id = $_POST["id"];
14.        $SQL = "SELECT * FROM messagein ORDER BY SendTime DESC LIMI
   T $id ";
15.        $result = mysql_query($SQL);
16.        $i=0;
17.        while ($data = mysql_fetch_array($result))
18.            {
19.                    $latlong[$i] = $data['LatLong'];
20.                    $datetime[$i] = $data['SendTime'];
21.                    list($lat[$i], $long[$i]) = explode(",", "$latl
   ong[$i]", 2);
22.                    $i=$i+1;
23.            }
24.
25.        mysql_close($db_handle);
26.    }
27.?>
28.<!DOCTYPE html "-//W3C//DTD XHTML 1.0 Strict//EN"
29.  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
30.<html xmlns="http://www.w3.org/1999/xhtml">
31.  <head>
32.    <meta http-equiv="content-
   type" content="text/html; charset=utf-8"/>
```

```
33.     <title>FYP Tracker</title>
34.     <script src="http://maps.google.com/maps?file=api&v=2&key=AIzaS
    yANwgDDObXoTZUOrD6PRZRXdVkMhD3t8ek&sensor=true_or_false"
35.             type="text/javascript"></script>
36.     <script type="text/javascript">
37.
38.     var count = "<?php echo $i ?>";
39.     var UnitID = "0001" ;
40.     lat = new Array();
41.     long = new Array();
42.     datetime = new Array();
43.
44.     <?php
45.         for ($z=0; $z<$i; $z++)
46.         {
47.             print "lat['$z']='$lat[$z]';";
48.             print "long['$z']='$long[$z]';";
49.             print "datetime['$z']='$datetime[$z]';";
50.         }
51.     ?>
52.     function setupMap()
53.     {
54.         if (GBrowserIsCompatible())
55.         {
56.             var map = new GMap2(document.getElementById("map"));
57.             map.setCenter(new GLatLng(lat[0], long[0]), 30);
58.
59.             points = [];
60.             points.push(new GLatLng(lat[0], long[0]));
61.             points.push(new GLatLng(lat[count-1], long[count-1]));

62.             directions = new GDirections(map, document.getElementBy
    Id("route"));
63.             directions.loadFromWaypoints(points);
64.
65.             var directions = new GDirections(map);
66.             directions.loadFromWaypoints(points);
67.             GEvent.addListener(directions, "load", function()
68.             {
69.                 document.getElementById('distance').innerHTML += di
    rections.getDistance().meters + " meters";
70.                 document.getElementById('duration').innerHTML += di
    rections.getDuration().seconds + " seconds";
71.             });
72.
73.             map.setMapType(G_HYBRID_MAP);
74.             map.openInfoWindowHtml(map.getCenter(),("UnitID: <b>" +
     UnitID + "</b>" + "<br>DateTime: <b>"
75.             + datetime[0] + "</b>" + "<br>Longitude:<b>" + long[0]
    + "</b>" + "</b>" + "<br>Latitude:<b>" +
76.             lat[0] ));
77.             map.addControl(new GOverviewMapControl());
78.             map.enableDoubleClickZoom();
79.             map.enableScrollWheelZoom();
80.             map.addControl(new GMapTypeControl());
81.             map.addControl(new GSmallMapControl());
82.
83.
```

```
84.             var baseIcon = new GIcon();
85.             baseIcon.shadow = "http://www.google.com/mapfiles/shado
    w50.png";
86.             baseIcon.iconSize = new GSize(20, 34);
87.             baseIcon.shadowSize = new GSize(37, 34);
88.             baseIcon.iconAnchor = new GPoint(9, 34);
89.             baseIcon.infoWindowAnchor = new GPoint(13, 30);
90.             baseIcon.infoShadowAnchor = new GPoint(18, 25);
91.
92.             function createMarker(point, index)
93.             {
94.                 if(index<26)
95.                 {
96.                     var letter = String.fromCharCode("A".charCodeAt
    (0) + index);
97.                     var letteredIcon = new GIcon(baseIcon);
98.                     letteredIcon.image = "http://www.google.com/map
    files/marker"+letter+".png";
99.                 }
100.                    else
101.                    {
102.                        var letteredIcon = new GIcon(baseIcon);

103.                        letteredIcon.image = "http://www.google.c
    om/mapfiles/marker.png";
104.                    }
105.
106.                    markerOptions = { icon:letteredIcon };
107.                    var marker = new GMarker(point, markerOptions
    );
108.
109.                    GEvent.addListener(marker, "click", function(
    )
110.                    {
111.                        marker.openInfoWindowHtml("UnitID: <b>" +
     UnitID + "</b>" + "<br>DateTime: <b>" +
112.                        datetime[index] + "</b>" + "<br>Longitude
    :<b>" + long[index] + "</b>" + "</b>" +
113.                        "<br>Latitude:<b>" + lat[index] );
114.                        });
115.                        return marker;
116.                }
117.                for (var a = 0;a < count; a++)
118.                {
119.                    var latlng = new GLatLng(lat[a],long[a]);
120.                    map.addOverlay(createMarker(latlng, a));
121.                }
122.
123.
124.
125.             }
126.         }
127.     </script>
128.      </head>
129.      <body onload="setupMap()" onunload="GUnload()">
130.        <div id="map" style="width: 750px; height: 350px"></div>

131.        <div id="duration">Duration: </div>
```

```
132.          <div id="distance">Distance: </div>
133.
134.
135.        </body>
136.      </html>
```

Route and Destination Planner

```
1. <?PHP
2.
3. $user_name = "root";
4. $password = "";
5. $database = "smsdatabase";
6. $server = "localhost";
7.
8. $db_handle=mysql_connect('localhost', 'root', '');
9. $db_found=mysql_select_db($database,$db_handle);
10.
11. if($db_found)
12. {
13.     $SQL = "SELECT * FROM messagein ORDER BY SendTime DESC";
14.     $result = mysql_query($SQL);
15.     if ($data=mysql_fetch_array($result))
16.     {
17.         $latlong = $data['LatLong'];
18.         $datetime = $data['SendTime'];
19.         list($latitude, $longitude) = explode(",", "$latlong", 2);

20.
21.     }
22.         mysql_close($db_handle);
23. }
24. ?>
25.
26. <!DOCTYPE html PUBLIC "-
    //W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/x
    html1-strict.dtd">
27. <html>
28. <head>
29.     <meta http-equiv="content-
    type" content="text/html; charset=UTF-8"/>
30.     <title>Google Maps</title>
31. <script src="http://maps.google.com/maps?file=api&v=2&key=AIzaSyANw
    gDDObXoTZUOrD6PRZRXdVkMhD3t8ek&sensor=true_or_false"
32.             type="text/javascript"></script>
33.     <script src="epoly.js" type="text/javascript"></script>
34.   </head>
35.   <body onunload="GUnload()">
36.
37.     <div id="controls">
38.      <form onsubmit="start();return false" action="#">
39.       Enter your place of interest.<br />
40.       <input type="text" size="80" maxlength="200" id="endpoint" va
    lue="" /><br />
41.       <input type="submit" value="Start"  />
```

```
42.      </form>
43.    </div>
44.
45.    <div id="map" style="width: 700px; height: 500px"></div>
46.    <div id="step"> </div>
47.    <div id="distance">Kilometres: 0.00</div>
48.
49.    <script type="text/javascript">
50.
51.    var UnitID="0001";
52.    var datetime="<?php echo $datetime ?>";
53.    var lat="<?php echo $latitude ?>";
54.    var long="<?php echo $longitude ?>";
55.
56.    if (GBrowserIsCompatible())
57.    {
58.      var map = new GMap2(document.getElementById("map"));
59.      map.addControl(new GMapTypeControl());
60.      map.setMapType(G_HYBRID_MAP);
61.      map.enableDoubleClickZoom();
62.      map.enableScrollWheelZoom();
63.      map.addControl(new GSmallMapControl());
64.      map.setCenter(new GLatLng(lat,long),14);
65.
66.      map.openInfoWindowHtml(map.getCenter(),("UnitID: <b>" + UnitI
   D + "</b>" + "<br>DateTime: <b>" +
67.        datetime + "</b>" + "<br>Longitude:<b>" + long + "</b>" + "
   <br>Latitude:<b>" + lat + "</b>"));
68.        map.setUIToDefault();
69.        map.enableRotation();
70.        var baseIcon = new GIcon();
71.        baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.
   png";
72.        baseIcon.iconSize = new GSize(20, 34);
73.        baseIcon.shadowSize = new GSize(37, 34);
74.        baseIcon.iconAnchor = new GPoint(9, 34);
75.        baseIcon.infoWindowAnchor = new GPoint(13, 30);
76.        baseIcon.infoShadowAnchor = new GPoint(18, 25);
77.
78.        function createMarker(point, index)
79.        {
80.            var letteredIcon = new GIcon(baseIcon);
81.            letteredIcon.image = "http://www.google.com/mapfiles/ma
   rker.png";
82.            markerOptions = { icon:letteredIcon };
83.            var marker1 = new GMarker(point, markerOptions);
84.            GEvent.addListener(marker1, "click", function() {
85.            marker1.openInfoWindowHtml("UnitID: <b>" + UnitID + "</
   b>" + "<br>DateTime: <b>" + datetime +
86.            "</b>" + "<br>Longitude:<b>" + long + "</b>" + "<br>Lat
   itude:<b>" + lat + "</b>");
87.            });
88.            return marker1;
89.        }
90.
91.        var latlong = new GLatLng(lat,long);
92.        map.addOverlay(createMarker(latlong, 0));
93.        var dirn = new GDirections();
```

```
94.         var step = 5; // metres
95.         var tick = 200; // milliseconds
96.         var poly;
97.         var poly2;
98.         var lastVertex = 0;
99.         var eol;
100.              var car = new GIcon();
101.                car.image="caricon.png"
102.                car.iconSize=new GSize(32,18);
103.                car.iconAnchor=new GPoint(16,9);
104.              var marker;
105.              var k=0;
106.              var stepnum=0;
107.              var speed = "";
108.
109.           function updatePoly(d)
110.           {
111.             // Spawn a new polyline every 20 vertices, because up
   dating a 100-vertex poly is too slow
112.             if (poly2.getVertexCount() > 20)
113.             {
114.               poly2=new GPolyline([poly.getVertex(lastVertex-
   1)]);
115.               map.addOverlay(poly2)
116.             }
117.             if (poly.GetIndexAtDistance(d) < lastVertex+2)
118.             {
119.                if (poly2.getVertexCount()>1)
120.                {
121.                  poly2.deleteVertex(poly2.getVertexCount()-1)
122.                }
123.                  poly2.insertVertex(poly2.getVertexCount(),poly.G
   etPointAtDistance(d));
124.                }
125.                else
126.                {
127.                  poly2.insertVertex(poly2.getVertexCount(),poly.g
   etVertex(lastVertex++));
128.                }
129.            }
130.
131.           function animate(d)
132.           {
133.             if (d>eol)
134.             {
135.               document.getElementById("step").innerHTML = "<b>Tri
   p completed<\/b>";
136.               document.getElementById("distance").innerHTML =  "K
   ilometres: "+(d/1000).toFixed(2);
137.               return;
138.             }
139.             var p = poly.GetPointAtDistance(d);
140.             if (k++>=180/step)
141.             {
142.               map.panTo(p);
143.               k=0;
144.             }
145.             marker.setPoint(p);
```

```
146.              document.getElementById("distance").innerHTML =  "Kil
   ometres: "+(d/1000).toFixed(2)+speed;
147.              if (stepnum+1 < dirn.getRoute(0).getNumSteps())
148.              {
149.                if (dirn.getRoute(0).getStep(stepnum).getPolylineIn
   dex() < poly.GetIndexAtDistance(d))
150.                {
151.                  stepnum++;
152.                  var steptext = dirn.getRoute(0).getStep(stepnum).
   getDescriptionHtml();
153.                  document.getElementById("step").innerHTML = "<b>N
   ext:<\/b> "+steptext;
154.                  var stepdist = dirn.getRoute(0).getStep(stepnum-
   1).getDistance().meters;
155.                  var steptime = dirn.getRoute(0).getStep(stepnum-
   1).getDuration().seconds;
156.                  var stepspeed = ((stepdist/steptime) * 2.24).toFi
   xed(0);
157.                  step = stepspeed/2.5;
158.                  speed = "<br>Current speed: " + stepspeed +" mph"
   ;
159.                }
160.              }
161.              else
162.              {
163.                if (dirn.getRoute(0).getStep(stepnum).getPolylineIn
   dex() < poly.GetIndexAtDistance(d))
164.                {
165.                  document.getElementById("step").innerHTML = "<b>N
   ext: Arrive at your destination<\/b>";
166.                }
167.              }
168.              updatePoly(d);
169.              setTimeout("animate("+(d+step)+")", tick);
170.            }
171.
172.            GEvent.addListener(dirn,"load", function()
173.            {
174.              document.getElementById("controls").style.display="no
   ne";
175.              poly=dirn.getPolyline();
176.              eol=poly.Distance();
177.              map.setCenter(poly.getVertex(0),17);
178.              map.addOverlay(new GMarker(poly.getVertex(0),G_START_
   ICON));
179.              map.addOverlay(new GMarker(poly.getVertex(poly.getVer
   texCount()-1),G_END_ICON));
180.              marker = new GMarker(poly.getVertex(0),{icon:car});

181.              map.addOverlay(marker);
182.              var steptext = dirn.getRoute(0).getStep(stepnum).getD
   escriptionHtml();
183.              document.getElementById("step").innerHTML = steptext;

184.              poly2 = new GPolyline([poly.getVertex(0)]);
185.              map.addOverlay(poly2);
186.              setTimeout("animate(0)",2000);  // Allow time for the
    initial map display
```

```
187.            });
188.
189.            GEvent.addListener(dirn,"error", function()
190.            {
191.              alert("Location(s) not recognised. Code: "+dirn.getSt
    atus().code);
192.            });
193.
194.            function start()
195.            {
196.              var startpoint = new GLatLng(lat,long);
197.              var endpoint = document.getElementById("endpoint").va
    lue;
198.              dirn.loadFromWaypoints([startpoint,endpoint],{getPoly
    line:true,getSteps:true});
199.            }
200.          }
201.
202.          </script>
203.        </body>
204.
205.      </html>
```