Vehicle Tracking and Alert System for Mini-Buses on the University of Ghana, Legon Campus

Wiafe Owusu-Banahene
University of Ghana, Legon
Computer Engineering Department
School of Engineering, Accra, Ghana
+233-(0)-209047639
wowusu-banahene@ug.edu.gh

Boateng Rene Korankye Aputae
University of Ghana, Legon
Computer Engineering Department
School of Engineering, Accra, Ghana
+233-(0)-578189085
rene.boat23@gmail.com

ABSTRACT

Most often decisions are made based on the amount of information available at a point in time. A good decision made implies that enough or accurate information was available. Mini buses are deployed on the University of Ghana Campus to facilitate movements of students and general users. Currently, there is no system in place to assist in locating these mini buses so they can access them. The work presented in this paper is an attempt to offer cost effective hardware-software integration to track the movement of mini buses that provide shuttle services from one location to the other on the University of Ghana, Legon campus in Accra, Ghana. The system aimed at disseminating information available to management and users of the bus service. The system consists of three modules namely**,** *data logging system (tracking hardware)*, w*eb-mapping application and* m*obile application*. The tracking hardware (made up of an Arduino Uno Microcontroller, a SIM 900 GSM/GPRS module and a GPS shield) is placed on a mini bus to log the coordinates into a database. A web server calls the data and plots it on an embedded Google map. Administrator can view the location of the mini bus on a Web page, register drivers as well as assign mini buses to them. The location of the shuttle can be viewed on an Android phone by users. The distance between the shuttle's position and the position of the user can be determined. The route can be traced on the map for users to see. The user can make decisions based on the information gathered to access the shuttle. This project would help students cut down the cost incurred when boarding taxis to their destinations on campus. On the side of management of the Mini buses, the system tracks mini buses on campus regularly. This project can offer a cost-effective approach to vehicle tracking.

Keywords

Android, Vehicle; GPS; GSM; GPRS; location; tracking; mobile

1. INTRODUCTION

Vehicle Tracking Systems (VTS) have emerged over the past few years. It has proven to be very useful in some areas of life. The commercial fleet management uses this technology to track its deployed mini buses. The military also uses this technology to track enemy vehicles and assets during combat. In each area of application, the stakeholder wants to be highly informed or wants first- hand knowledge of locations of assets. The days of following a bus to keep an eye on the driver to know where he is at a point in time are over.

VTS technology is employed to determine the location of a vehicle using methods such as GPS and other radio navigation systems that operate through satellites and ground based stations [1]. A GPS tracking unit is a device, normally carried by a moving vehicle or person, that uses Global Positioning System to determine and track its precise location, and hence that of its carrier, at intervals. The data or coordinates recorded can be transmitted to a central location database, or internet connected computer, using a cellular, radio, or satellite modem embedded in the unit [2].

The management of the mini buses deployed on the University of Ghana, Legon Campus to facilitate movements of students and general users. Shuttle transport system finds it difficult to locate shuttles that have been deployed to work on campus. Students also find it difficult to locate and access the services of shuttles at any given time on campus. In other to reach their lecture halls early enough they board taxis or trek longer distances. Cost is incurred when they board taxis from their halls to lecture rooms and vice versa thus increasing their cost of living. Students end up being late for classes especially early in the mornings when they decide to walk. They end up tired before the beginning of the lecture itself. It will be helpful for management to know the location of mini buses anywhere and anytime just by accessing a webpage.

This research seeks: 1) to design a GPS tracking device that would log coordinates from the moving vehicle into a database; 2) to develop a web application that would allow an administrator to track and view the location of a moving vehicle on a map; 3) to develop a web application that would allow an administrator register drivers and assign a bus to them; 4) to develop a mobile application that would log user coordinates and show their location on a map; 5) to develop a mobile application that would calculate the distance of the bus with respect to the user and trace the route. This project serves as a Vehicle Tracking System for administrators and a Vehicle Information System for users as well. All these can be done inside or outside the office.

The remaining paper is structured as follows. Section presents a review of related work. In section 3, the system design is presented. Implementation, results and discussions are described in section 4. The paper ends with conclusions and future work in section 5.

2. REVIEW OF RELATED WORK

Most of the Vehicle Tracking System ([12]-[17]) make use of a GPS tracking component. This tracking device comes in different types depending on one's specification or system requirement.

Some of the GPS trackers are already built together as a small component by manufacturers for sale or can be assembled using these three key hardware components; a microcontroller, a GSM module and a GPS shield. These are the basic components of most tracking devices. The tracking device is usually installed on the car or vehicle that needs to be tracked.

After the installation the administrator needs to receive information about the vehicle. This gives rise to a system, mechanism or interface where monitoring could be done with or without the knowledge of the driver. The purpose of the tracking is also carefully looked at in this section.

2.1 Related Work

2.1.1 Vehicle Tracking System using GSM and GPS

In this project [1], an AT89S52 microcontroller was interfaced serially to a GSM Modem and GPS Receiver. A GSM modem was used to send the position (Latitude and Longitude) of the vehicle from a remote place. Many parameters were outputted but only the NMEA data coming out was read and displayed on an LCD in the vehicle. The same data was sent as a message to a mobile phone at the other end upon demand. The request was also made by sending an SMS to the GMS/GPRS modem in the vehicle. [1]

This project was easy to implement. The position (Latitude and Longitude) was displayed on an LCD only to those in the vehicle. The administrator also received the location through an SMS.

2.1.2 GPS, GPRS, GIS for Tracking Systems

This project [2] was done to solve the problem of sending coordinates to the server through SMS. The system has two parts – the tracking device and the server side. The server side included a database server, a map server and GIS platform selected for the map server. The system used GPRS as the main method of communication between the unit to track and server. The device is attached with the moving bus and gets the position from GPS satellite in real-time. It then sends the position information with the International Mobile Equipment Identity (IMEI) number as its own identity to the server. The data is checked for validity and the valid data is saved into the database. When a user wants to track the bus, she/he logs into the service provider's website and gets the live position of the bus on Google Map.

The system reduces the cost in sending SMS to the database server since it uses only GPRS. It is easier and less expensive to implement. The location of the vehicle can be accessed by only one user.

2.1.3 Smart Vehicle Tracking System

In this project [10], the design was an embedded application which continuously monitored a moving Vehicle. The system reported the status of the Vehicle on demand. A microcontroller was interfaced serially to a GSM Modem and GPS Receiver. The GSM modem was used to send the position (Latitude and Longitude) of the vehicle from a remote place. The GPS modem continuously gave the data i.e. the latitude and longitude indicating the position of the vehicle. To find the location of the vehicle, the owner sends a message to the vehicle tracking system. When the user request is sent to the number in the modem, the system sends a return reply automatically to that mobile which indicate the position of the vehicle with latitude and longitude. The software can produced reports in quick time. The project was easy to implement. The owner could access the location through SMS [10].

2.1.4 Vehicle Tracking and Alert System for Commercial Inter-City Buses using GPS/GSM Technologies

The project [11] makes use of a Vehicle Tracking component built using a Microcontroller, GSM module and a GPS shield. It has alert subsystems that informs the administrator. These alerts include an accident alert and a theft alert. The positions of the bus can be viewed by an administrator as he logs onto the Vehicle Tracking System Interface. Passengers are informed by viewing an LED board installed on the bus. The locations or bus stops are displayed only to passengers on the LED board as the bus moves. [11]

2.1.5 Smart Bus for Human Safety

This system [12] and research work has the ability to detect/sense the obstacles causing accidents. This system makes use of sensors to detect in bound threats or accidents. It has the Alcohol Sensor which detects the amount of alcohol present in the body of the driver. The system uses MQ3 gas sensors to detect alcohol and transfer the data to the controller. If the detected value is higher than the threshold value, the ignition system shuts down. The Passive Infrared sensor of the avoidance system is used if the obstacle been approached is human or animal. If the avoidance is not possible and accident happens then this system generate an SMS, also internet based alert through GSM module, including tracking the position of accident using GSM [12].

2.2 The Mobile, Web-Mapping and Satellite Based Positioning Approach

The work presented in this paper was designed to track the movement of mini buses that provide shuttle services from one location to the other on the University of Ghana, Legon campus in Accra, Ghana. The system aimed at disseminating information available to management and users of the bus service. The system consists of three modules namely**,** *Data logging system (Tracking device), Web-Mapping Application***,** *Mobile Application*. The *Data logging system (Tracking device)* module is a hardware designed as a tracking device that can be installed on a moving shuttle. The hardware tracking device logs coordinates (Latitude and Longitude) of the moving bus into a database on a web server to be processed. The *Web-Mapping Application* module is responsible for providing only the management or an administrator the privilege of viewing the moving shuttle on an embedded Google map. The web interface allows an administrator to register and assign mini buses to drivers. The third module is the mobile application interface. This mobile application runs on any Android supported smart phone of users or passengers who will access the shuttle. This module is responsible for logging user or passenger coordinates (Latitude and Longitude) into a database on the web server. It is responsible for displaying the distance between the shuttle and a user after finding their respective coordinates. The locations of the shuttle and user can be viewed on an embedded Google map and their route traced. All these information will be available to the user to enable him access the services of the shuttle.

3. SYSTEM DESIGN

3.1 System Architecture

The general architecture of the system consists of a hardware designed tracking device which logs coordinates of a moving bus into a database. The web server calls the information received from the database to be used by the web and mobile application modules by the administrator and user respectively.

The block diagram representation of the system (see Figure 1) provides a detailed abstraction of the system in modules. The whole system demarcated into three blocks; Data logging system (Tracking device), Web Application Interface and Mobile Application Interface modules makes it simple to implement.

The data logging system (Tracking device) logs coordinates of the moving bus into a database. The Web application calls or gets the coordinates logged into the database. The coordinates are plotted and viewed on an embedded Google map Application Programming Interface (API) by the administrator. The GPS capabilities of the user's Android smart phone is employed here. The location of the user is also tracked using the phone and the coordinates logged into the database. These coordinates are called by the mobile application when the user hits a button on the Graphic User Interface (GUI).

The Google map API has functions that allow routes to be traced and distances between two points to be calculated. These two functions were implemented in this project.

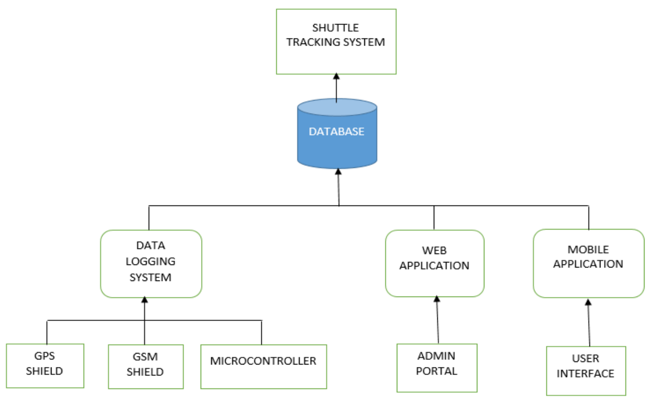

Figure 1 Block diagram of the system

3.2 System Requirements

The requirements of the system were derived as follows.

Requirement 1. *The system should be able to log coordinates of a moving bus into a database.*

Requirement 2. *The system should be able to log coordinates of a user into a database.*

Requirement 3. *The system should be able to show the location of a moving bus on a map to an administrator at any point in time.*

Requirement 4. *The system should allow an administrator to register drivers.*

Requirement 5. *The system should allow an administrator to assign a bus to a registered driver.*

Requirement 6. *The system should be able to show the location of a user on a map relative to that of a bus.*

Requirement 7. *The system should be able to display the distance between the user and bus.*

3.3 Software and Hardware Used

In this section the software and hardware used to develop the system are briefly presented. The following software were used

3.3.1 Software Specifications

3.3.1.1 Arduino: IDE and Development Libraries

Arduino Integrated Development Environment (IDE) - or Arduino Software[6] - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. Arduino development libraries are written in C or C++ (.c, .cpp) which provide user sketches with extra functionality. Some of these functions include the ability to control a sensor, control an LED matrix, or read an encoder [6].

3.3.1.2 Sublime Text 2 Text Editor

Sublime Text is a cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and its functionality can be extended by users with plugins, typically community-built and maintained under free-software licenses [8].

3.3.1.3 Android Studio IDE

Android Studio [9] is an integrated development environment (IDE) for Android platform development. It has Gradle-based build support with Android-specific refactoring and quick fixes, Lint tools to catch performance, usability, version compatibility and other problems. It has a template-based wizard to create common Android designs and components, a rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations [9].

3.3.1.4 XAMPP Apache Web Server Solution

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.

3.3.1.5 MySQL

MySQL stands for My Structured Query Language. It is a file based database management system. It is an open source database management system which is not accessed from a server but is integrated with the client application.

3.3.1.6 BOOTSTRAP

Bootstrap is a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development.

3.3.1.7 JAVASCRIPT

JavaScript is a dynamically typed web scripting language that introduces dynamism to the client side of the web application. It is very useful for client side data validation before the submission of data.

3.3.2 Hardware Specifications

3.3.2.1 Personal Computer

A personal computer (PC) is a general-purpose computer operated directly by an end-user. This is needed to install all software to be used in the project.

3.3.2.2 ROYALTEK REB-4216 CT GPS Shield

RoyalTek REB-4216 is a GPS module powered by latest SiRF Star IV GSD4e ROM chip and RoyalTek proprietary navigation technology that provides stable and accurate navigation data. It has 48 track verification channels, 5V Power supply and removes in-band jammers up to 80 dB-Hz. It has excellent sensitivity for urban canyon and foliage environments.

3.3.2.3 ATMEGA328P (on Arduino Uno Board)

The Arduino Uno is a microcontroller board based on the ATmega328P. It has fourteen (14) digital input/output pins. Six (6) can be used as Pulse Width Modulation outputs, six (6) analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an In-Circuit Serial Programming(ICSP) header and a reset button. [7]

3.3.2.4 SIM 900 GSM/GPRS Shield

The SIM900 GSM/GPRS Shield provides a way to use the GSM cell phone network to receive data from a remote location. The shield achieves this through any of the three methods: Short Message Service, Audio and General Packet Radio Service (GPRS). The GPRS Shield is compatible with all boards which have the same form factor (and pin out) as a standard Arduino Board. The GPRS Shield is configured and controlled via its UART using simple AT commands. It is based on the SIM900 module from SIMCOM and has 12 GPIOs, 2 PWMs and an ADC.

3.3.2.5 Serial Cable

The serial cable is a cable used to transfer information between two devices using a serial communication protocol. It used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART). It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output.

4. IMPLEMENTATION, RESULTS AND DISCUSSIONS

This section presents the implementation of a prototype of the bus tracking system**.** The main modules namely, *Data logging system (Tracking device)*, *Web Application Interface*, *Mobile Application Interface* and *Web mapping API*, were implemented (see sections 4.1 to 4.4) and integrated (see section 4.5). The implementation of the *Web mapping API* is presented as part of the *Web Application Interface.* This section discusses how the requirements are satisfied with the implementation of the system in section 4.6. The software and hardware described in section 3.3 were used to implement the system based on the design (section 3.1) and requirements (section 3.2).

4.1 The Data Logging (Tracking) Module

The ground (GND), Tx (7) and Rx (6) pins of the RoyalTek GPS shield were used for the connections to the Arduino Microcontroller. The GPS Shield Tx (7) and Rx (6) pins communicated with the Tx (1) and Rx (0) pins of the Arduino Microcontroller. The SIM 900 GSM/GPRS module also used the ground (GND), Tx (4) and Rx (3) pins to communicate with the Arduino Microcontroller on Tx (4) and Rx (3) pins. Both the GPS shield and GSM/GPRS module were supplied with 5V from a Power pin on the Arduino microcontroller. The data logging module was programmed in C++ using the Arduino IDE and burned or uploaded onto the microcontroller. TinyGPS and Software Serial libraries were included in the C++ codes in order to get the GPS working. The Software Serial library has a function SoftwareSerial SIM900 (4, 3) that configures the software serial ports Tx and Rx pins according to desired specifications to enable communication. The function Serial.print("power up") turns on the SIM 900 GSM/GPRS module. The AT commands below starts and ends the HTTP request on the SIM 900 GSM/GPRS module.

```
void loop()
{  Serial.println("SubmitHttpRequest - started" );
  SubmitHttpRequest();
  Serial.println("SubmitHttpRequest - finished" );
  delay(10000);
}
```

The Software Serial library has a function SoftwareSerial GPS (7, 6); which configures the software serial ports Tx and Rx for the GPS pins 7 and 6 respectively to communicate with the microcontroller. Figure 2 shows the current location table in the MySQL database.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	current_id	int(50)			No	None	AUTO_INCREMENT
2	current_longitude	varchar(50)	latin1_swedish_ci		No	None	
3	current_latitude	varchar(50)	latin1_swedish_ci		No	None	
4	loc_datetime	varchar(50)	latin1_swedish_ci		No	None	

Figure 2 Current location table in database

4.2 The Web Application Interface

The Web application interface gets coordinate information from the hardware tracking device. The tracking device (tracker) writes its information into the database. This interface serves as the administrator's tracking view system. The administrator accesses this web application by entering the following address www.bustracker.triqlegh.com in a web browser. The administrator can have a dedicated personal computer at his office to use or could access the website on his phone when he is out of the office. To keep track of the moving shuttle, data was constantly been written to the database. The database kept updating itself to prevent to prevent congestion or traffic on the system. The database was created together with tables to hold the admin, bus and user information.

Timestamps of the logged coordinates were attached to keep track of the incoming data from the moving shuttle. The credentials of the administrator was saved in the database for validation during testing. The administrator enters a user name and a password to log onto the system. The administrator's web application had two main interfaces: the Google map interface and the Register Driver interface. The Google map interface was the main page for the administrator. There the moving shuttle could be viewed on an embedded Google map. The map updated every ten (10) seconds to show the change in position and location of the moving shuttle. The Register Driver interface gave the administrator the ability to assign a bus to a registered driver. Figure 3 shows the driver registration table implemented in the MySQL database. Figure 4 shows the use case diagram for the administrator.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	driver_id	int(50)			No	None	AUTO_INCREMENT
2	FirstName	varchar(255)	latin1_swedish_ci		Yes	NULL	
3	LastName	varchar(255)	latin1_swedish_ci		Yes	NULL	
4	Contact	int(11)			No	None	
5	ShuttleName	varchar(255)	latin1_swedish_ci		Yes	NULL	

Figure 3 Driver registration table in database

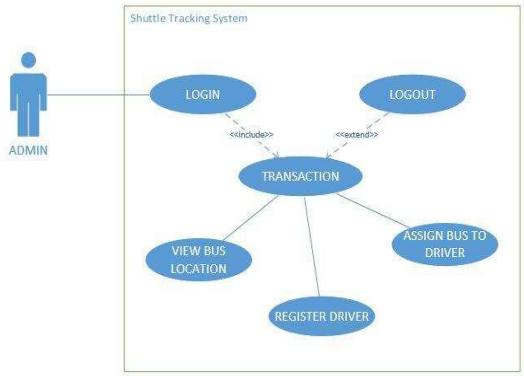

Figure 4 Administrator use case diagram

4.3 The Mobile Application Interface

The system is a real time system with response and interaction constraints and thus needs a minimum android version 2.3 to operate. This interface will be run on android smart phone platform hence was developed using Android Studio IDE.

Android Studio has some useful libraries that were implemented in the project. The android location library was implemented to get the locations of the user from the GPS capability on his phone. protected void onCreate(Bundle savedInstanceState) {} creates the first instance of the layout for the Bus tracker interface. The lines of code below was used to get the bus location and time.

```
Public void onClick(View view) {
  progress_layout.setVisibility(View.VISIBLE);
  getLocation(HomeActivity.this);
  if(isOnline()) {
    getData(URL);
        bus_location.setText("Lat :" + latitude_bus +
        " | Lng :"+ longitude_bus);
    ETA.setText(bus_time);
  }
```

To get the location of the user the code below was implemented. The getLocation function looks for the GPS updates of the phone, GSM network and passive updates.

```
private void getLocation(Context context) {
  TrackerSettings settings = new TrackerSettings()
        .setUseGPS(true)
        .setUseNetwork(true)
        .setUsePassive(true);
```

```
If (ContextCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION)!=PackageManager.PERMISSION_GRANTED&&ContextCompat.checkSelfPermission(context,Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) }
```

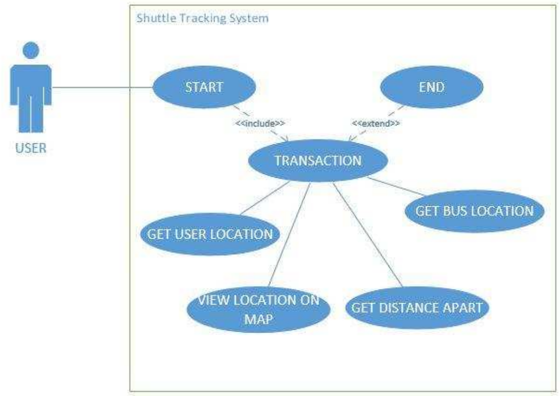

Figure 5 a use case diagram for users.

4.4 System (Hardware) Integration

After separate configuration of the three hardware modules, they were integrated to form the whole tracking system. The two modules that is the GPS Shield and the SIM 900 GSM/GPRS module were implemented together with the main controller in their individual C++ programs following a test process. The two (2) modules were put together and connected to the microcontroller mainly via a breadboard especially for power distribution and complexity problems. Figure 6 and Figure 7a shows the system (hardware) integration and schematic diagram respectively. The resulting hardware is shown in Figure 7b.

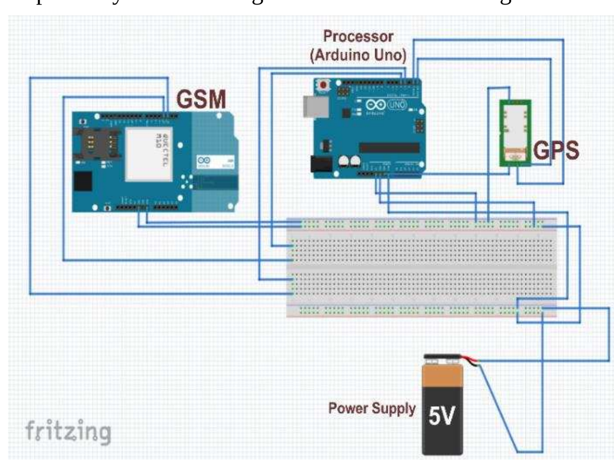

Figure 6 System (hardware) integration

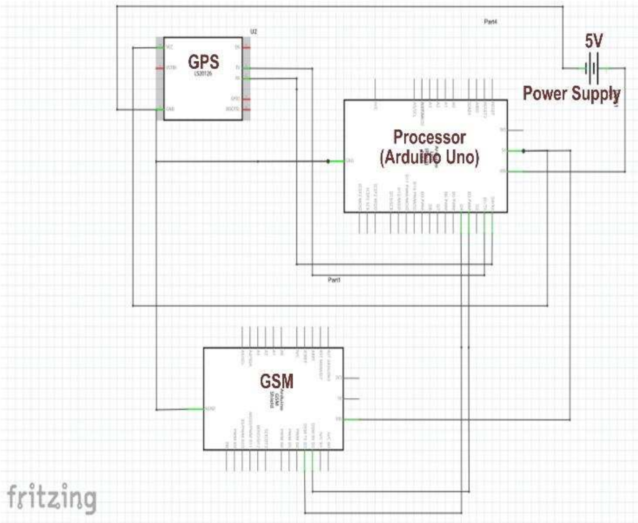

Figure 7a Tracker hardware system schematic diagram

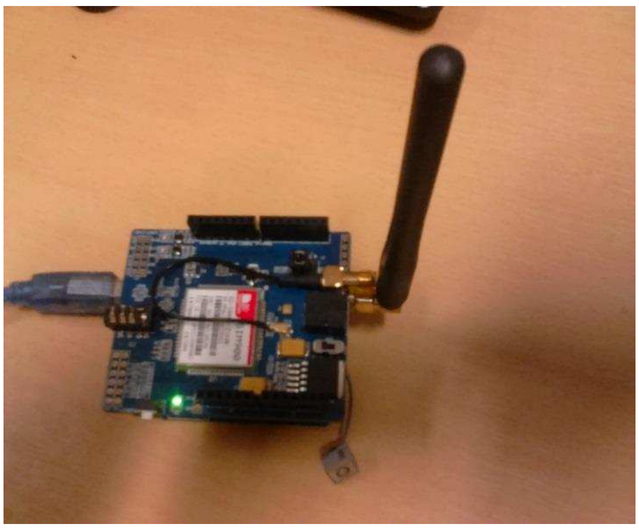

Figure 7b Tracker hardware system schematic diagram

4.5 Testing and Results

Testing of the system was done many times during and after implementation and integration on the University of Ghana, Legon campus in Accra, Ghana.

4.5.1 Hardware Module Testing and Results

The hardware tracking device was tested with functions to make sure coordinates were been logged into the database such as shown in Figure 8. This satisfies requirement 1: *the system should be able to log coordinates of a moving bus into a database.*

current_id	current_longitude	current_latitude	loc_datetime
1	-0.17774111	5.67841585	09-06-2016 04:09:29

Figure 8 Current location logged into database from tracker

4.5.2 Web Interface Testing and Results

This testing begun with the login page as shown in Figure 9 for the administrator. The admin was asked to login with his credentials. The credentials were his user name and password. He was then validated to make sure he was the legitimate admin of the system.

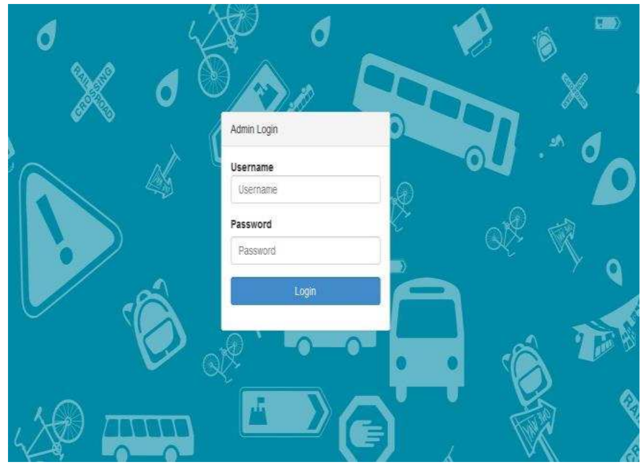

Figure 9 Admin login interface

After a successful login the admin visualizes the movement of the shuttle on a Google embedded map. The *View Bus Location* interface displays an icon of the moving bus and the coordinates to the admin as shown in Figure 10. The map view interface refreshes every 10 seconds to ensure real time visual of the moving shuttle. The timestamps of the coordinates show the time they were displayed to the admin as shown in Figure 11 to Figure 15. The navigation bar at the top has three buttons displayed. The *View Bus Location* displays the moving shuttle location.

The *Register Driver* web page shown in Figure 16 allows the admin to register drivers onto the system and assign shuttles to them. Figure 17 shows the successful registration of a driver whilst Figure 18 shows the results of attempting to register an existing driver. The Logout ends the session for the admin. With this the following is satisfied:

Requirement 3. *The system should be able to show the location of a moving bus on a map to an administrator at any point in time.*

Requirement 4. *The system should allow an administrator to register drivers.*

Requirement 5. *The system should allow an administrator to assign a bus to a registered driver.*

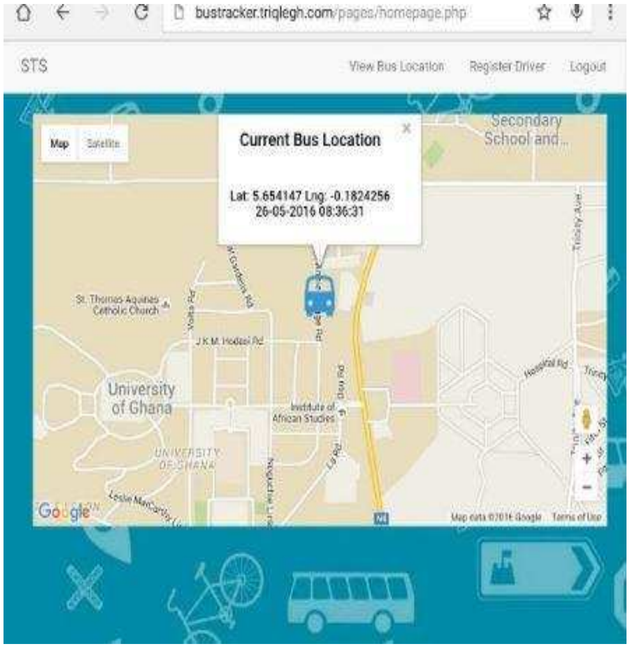

Figure 10 Current bus location at 8:36:31

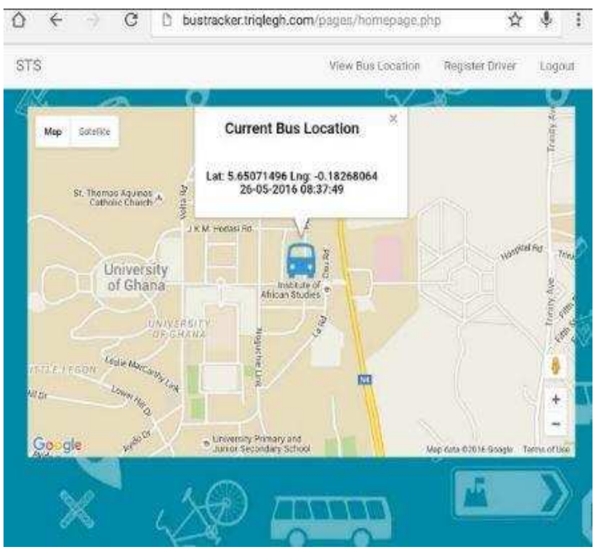

Figure 11 Current bus location at 8:37:49

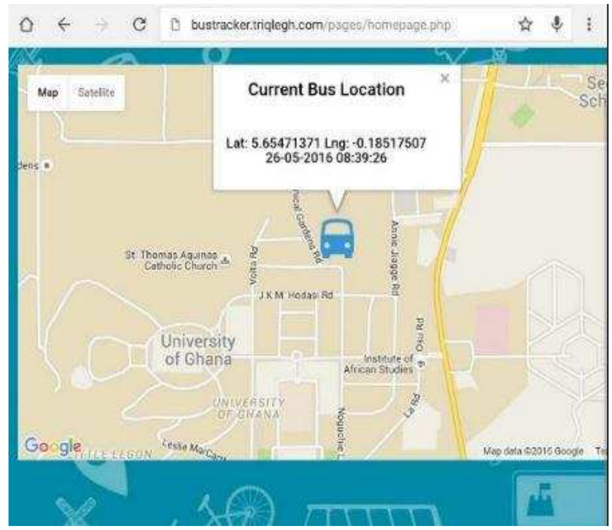

Figure 12 Current bus location at 8:39:26

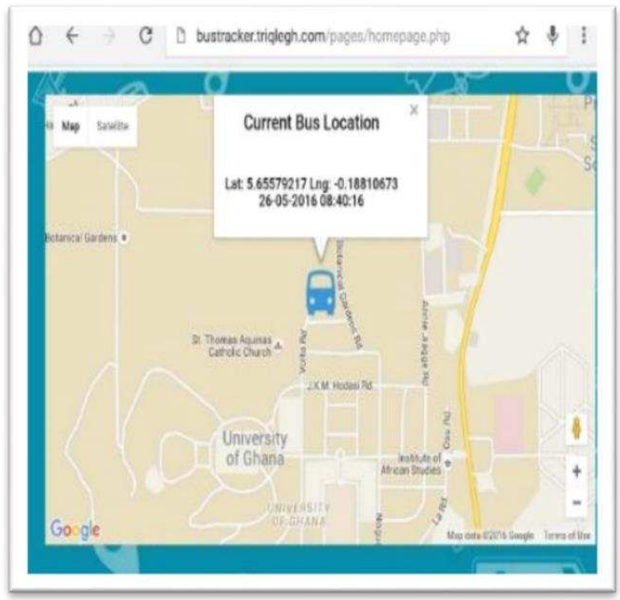

Figure 13 Current bus location at 8:40:16

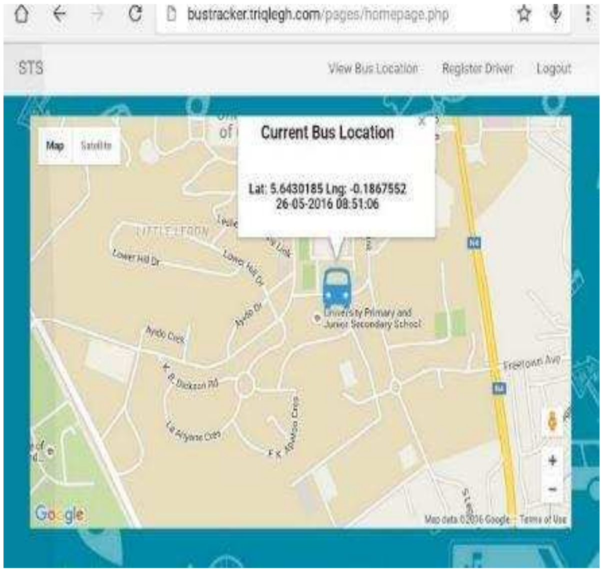

Figure 14 Current bus location at 8:51:06

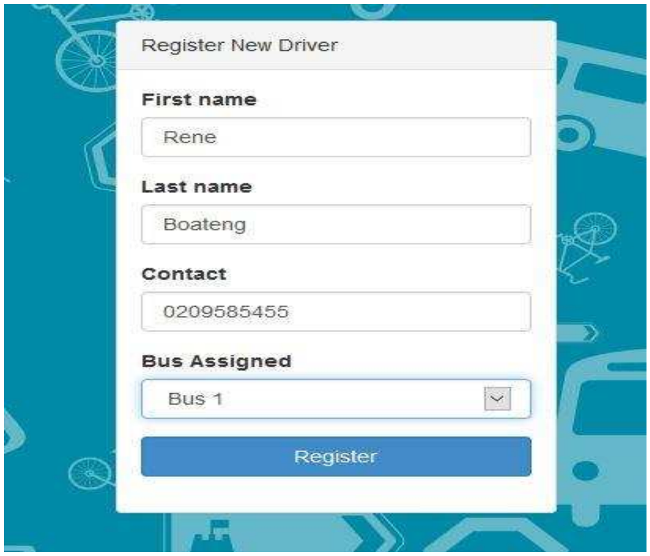

Figure 16 Register driver panel

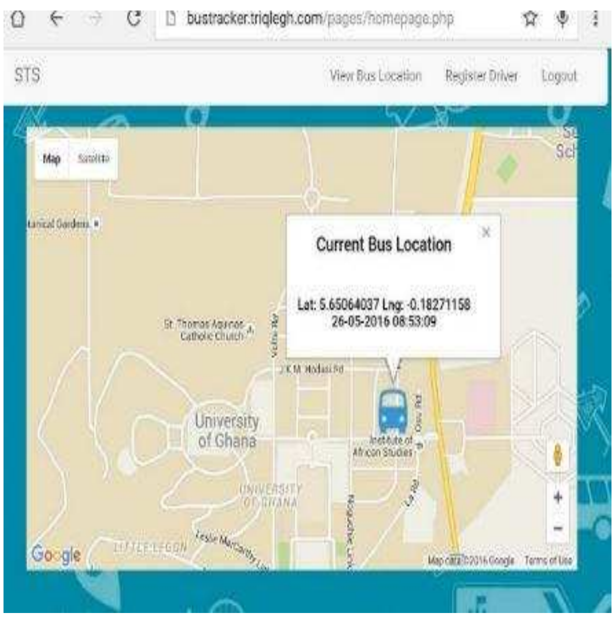

Figure 15 Current bus location at 8:536:09

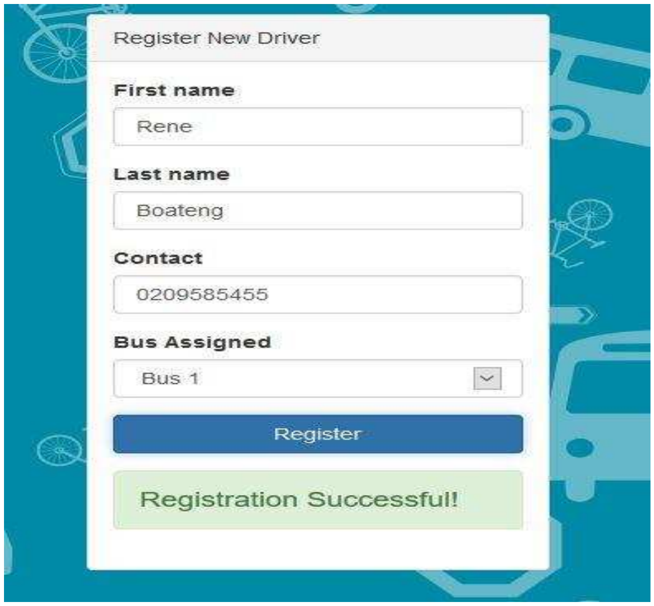

Figure 17 Successful driver registration and bus assignment

Figure 18 Unsuccessful driver registration and bus assignment

4.5.3 Mobile Interface Testing and Results

The user interface needs no login and authentication. The mobile app is installed and launched. The location interface displayed as shown in Figures 19 and 21. The mobile app gets the shuttle coordinate location, user coordinate location and calculates the distance between them and displays it for the user to see as shown in Figures 20 and 22. In lieu of requirement 7 is satisfied thus: *the system should be able to display the distance between the user and bus*. The user can then view the locations on the embedded Google map when he clicks on the *View Bus Location button*.

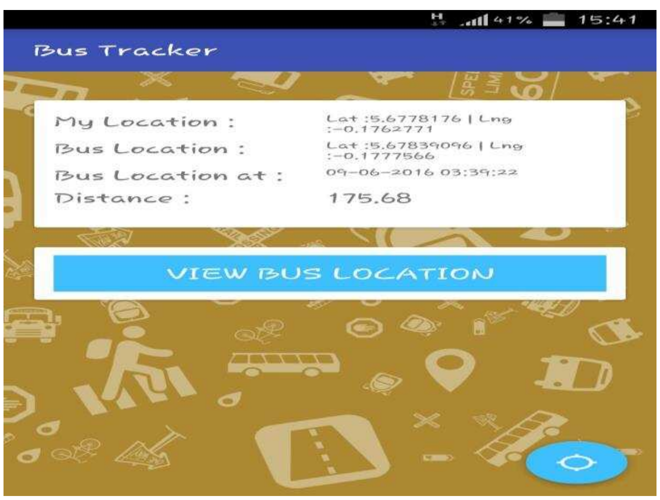

Figure 19 User Mobile Interface – Accessing the location of the bus

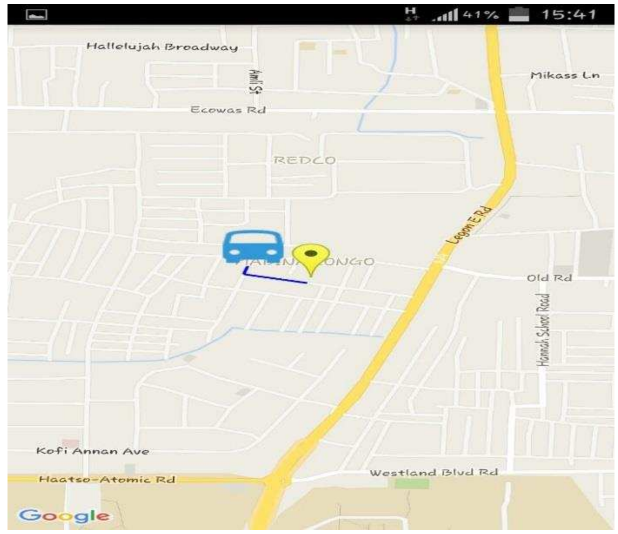

Figure 20 User Mobile Interface – Viewing location with route traced

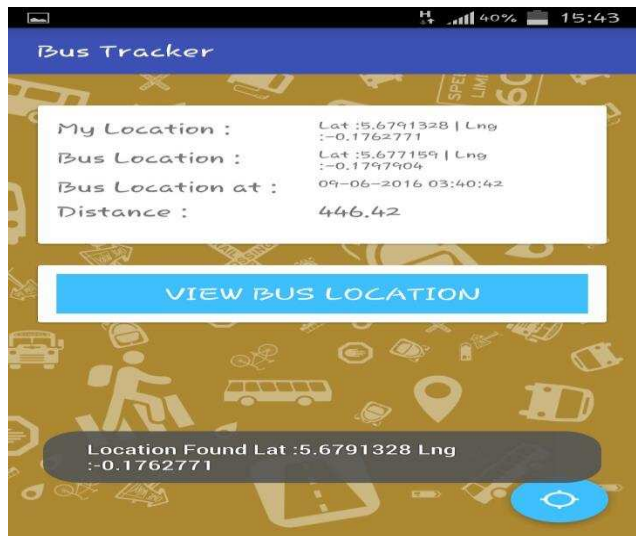

Figure 21 User Mobile Interface – Accessing the location of the bus

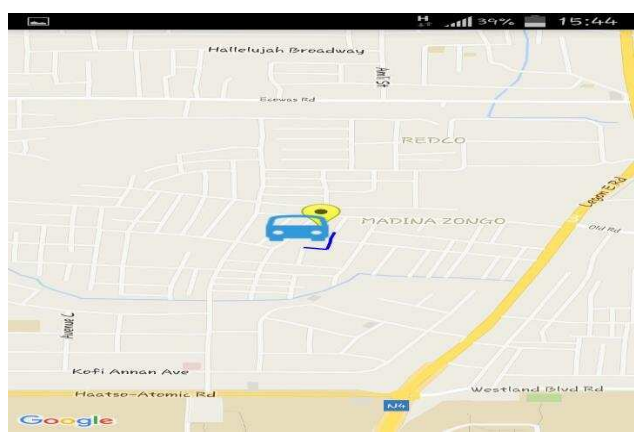

Figure 22 User Mobile Interface – Viewing location with route traced

4.6 Discussions

The system consists of three modules namely, *Data logging system (Tracking device), Web-Mapping Application, Mobile Application*. All modules were successfully implemented and all seven requirements were met.

The data logging system (Tracking device) consists of an Arduino Uno Microcontroller, a SIM 900 GMS/GPRS module and a RoyalTek GPS Shield. These three hardware components were connected together to log coordinates of the moving bus into the database of a web server.

The Web application calls or gets the coordinates logged into the database. The coordinates are plotted and viewed on an embedded Google map Application Programming Interface (API) by the administrator.

The Mobile Application Interface makes use of the GPS capabilities of an Android smart phone. The location of the user is tracked using his phone and the coordinates logged into the database. These coordinates are called by the mobile application when the user hits a button on the Graphic User Interface (GUI).

The Web mapping API is in the form of Google map API. Google map API has functions that allow routes to be traced and distances between two points to be calculated. These two functions were implemented in this project for the user to visualize the location of the shuttle.

The tests results from Figure 8 shows the logged coordinates into the database. The test results from Figures 18 – 20 shows the coordinates of the moving bus and the user at different locations on campus. The response time from the tracker, the web app interface and the mobile interface shows the efficiency of the system. The coordinates are posted in real time making the system really reliable. The admin web interface has a very simple interface. The mobile app is user friendly.

5. CONCLUSIONS AND FUTURE WORK

The work presented in this paper aimed to track the movement of mini buses that provide shuttle services from one location to the other on the University of Ghana, Legon campus in Accra, Ghana. The system consists of three modules namely, *Data logging system (Tracking device), Web-Mapping Application, Mobile Application*. All modules were successfully implemented and all seven requirements were met. The system is user friendly and can operate with minimum system specifications. It is not expensive to implement.

The system can be improved by adding more mini buses and other functionalities to it.

REFERENCES

[1] Bharath C.H. et al 2014. *Vehicle Tracking System using GPS and GSM.* Sagar Institute of Technology, India,

[2] Bidgoli A. et al, 2011 "GPS, GPRS, GIS for Tracking Systems" *IJCSET*.l 1, 8 (Sep 2011), 527-529.

[3] Michael Barr. *"Embedded Systems Glossary". Neutrino Technical Library. .*

[4] *Heath, Steve (2003). Embedded systems design. EDN series for design engineers (2 ed.). Newnes. p. 2. ISBN 978-0-7506-5546-0. An embedded system is a microprocessor based system that is built to control a function or a range of functions*

[5] *Giovino, Bill. "Micro controller.com – Embedded Systems supersite".*

[6] Arduino Libraries. https://www.arduino.cc/en/hacking/libraries

[7] In-system-programming. https://en.wikipedia.org/wiki/In-system_programming

[8] Sublime Text. https://en.wikipedia.org/wiki/Sublime_Text

[9] Android Studio. https://en.wikipedia.org/wiki/Android_Studio

[10] Kamble, K. P. 2012. SMART BUS FOR HUMAN SAFETY*", International Journal of Distributed and Parallel Systems (IJDPS)*.3, 4, (July 2012)

[11] Fleischer P. B. and Nelson, A. Y. 2013. Vehicle Tracking and Alert System for Commercial Inter-City Mini buses using GPS/GSM Technologies, *Bachelor Dissertation*. University of Ghana.

[12] Dhake P.,A.,Deshpande A. and Patil. N.2016. Smart Bus for Human Safety, *IJARIIE, 2,2, (2016).*2395-4396.

[13] Lee, S., Tewolde, G. and Kwon, J., 2014. Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on* (March, 2014).353-358). IEEE.

[14] Verma, P. and Bhatia, J.S., 2013. Design and development of GPS-GSM based tracking system with Google map based monitoring. *International Journal of Computer Science, Engineering and Applications*, *3*,3, (2013).33.

[15] Khan, A. and Mishra, R., 2012. GPS–GSM based tracking system.*International Journal of Engineering Trends and Technology*, *3*,2,(2012).161-164.

[16] Shete, G.R., Shah, D.A., Gaidhani, A.R. and Sharma, S.R., 2015. Smartphone based Vehicle Tracking and Accident Prevention System.*International Journal of Computer Applications & Information Technology*,*8*(1), (Jan, 2015).172.

[17] Patel, A. and Parakh, A., 2016. Automobile Tracking and Parameter Monitoring using GPS and GSM. *Journal of Electrical and Power System Engineering*, *2*,2, (Jul,2016).