REAL TIME BUS TRACKING SYSTEM

By

TINOTENDA KATSAMBA

(H190124G)

HIT400 Capstone project Submitted in Partial Fulfillment of the

Requirements of the degree of

Bachelor of Technology

In

Software Engineering

In the

School of Information Sciences and Technology

Harare Institute of Technology

Zimbabwe

Supervisor: Miss Zindove

……..

05/23

Table 1 Marking guide

HIT 400 /200 Project Documentation Marking Guide

ITEM	TOTAL MARK /%	ACQUIRED/%
PRESENTATION- Format-Times Roman 12 for ordinary text, Main headings Times Roman 14, spacing 1.5. Chapters and sub-chapters, tables and diagrams should be numbered. Document should be in report form. Range of document pages. Between 50 and 100.Work should be clear and neat	**5**	
Pre-Chapter Section Abstract, Preface, Acknowledgements, Dedication & Declaration	**5**	
Chapter One-Introduction Background, Problem Statement, Objectives – smart, clearly measurable from your system. Always start with a TO… Hypothesis, Justification, Proposed Tools Feasibility study: Technical, Economic & Operational Project plan –Time plan, Gantt chart	**10**	
Chapter Two-Literature Review Introduction, Related work & Conclusion	**10**	
Chapter Three –Analysis Information Gathering Tools, Description of system Data analysis –Using UML context diagrams, DFD of existing system Evaluation of Alternatives Systems, Functional Analysis of Proposed System-Functional and Non-functional Requirements, User Case Diagrams	**15**	
Chapter Four –Design Systems Diagrams –Using UML Context diagrams, DFD, Activity diagrams Architectural Design-hardware, networking Database Design –ER diagrams, Normalized Databases Program Design-Class diagrams, Sequence diagrams, Package diagrams, Pseudo code Interface Design-Screenshots of user interface	**20**	
Chapter Five-Implementation & Testing Pseudo code of major modules /Sample of real code can be written here Software Testing-Unit, Module, Integration, System, Database & Acceptance	**20**	
Chapter Six –Conclusions and Recommendations Results and summary, Recommendations & Future Works	**10**	
Bibliography –Proper numbering should be used Appendices –templates of data collection tools, user manual of the working system, sample code, research papers	**5**	
	100	**/100**

Certificate of Declaration

This is to certify that work entitled Real Time Bus Tracking System *is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology (Hons) in Software Engineering ,Harare Institute of Technology .It is further certified that no part of research has been submitted to any university for the award of any other degree .*

(Supervisor) Signature…….. Date…….

(Mentor) Signature…….. Date……

(Chairman) Signature…….. Date……..

Table of Contents

List of figures

Table of figures

Pre-Chapter Section

Abstract:

Real-time bus tracking systems have become increasingly popular in recent years, as they offer a more efficient and convenient way for commuters to plan their journeys and reduce the time spent waiting for public transportation. This paper proposes a real-time bus tracking system that utilizes GPS and mobile data networks to track the location of buses in real-time. The system allows commuters to access information about the location and estimated arrival time of buses through a mobile application or a web interface. The proposed system is designed to be scalable, reliable, and easy to use, and it can be implemented in any city with a public transportation system. Results from a pilot study show that the system is effective in reducing waiting times and improving the overall commuting experience for users.

Preface:

Public transportation systems play a vital role in meeting the transportation needs of modern cities. However, one of the significant challenges faced by commuters is the uncertainty and unpredictability of bus schedules, leading to long waiting times and inconvenience. Real-time bus tracking systems are designed to address this issue by providing commuters with up-to-date information on bus locations and arrival times. The aim of this paper is to present a real-time bus tracking system that utilizes modern technology to improve the efficiency and convenience of public transportation. The proposed system is designed to be user-friendly, scalable, and reliable, and it can be customized to meet the specific needs of different cities and transportation networks. This paper presents the design, implementation, and evaluation of the system, along with a pilot study that demonstrates the effectiveness of the system in reducing waiting times and improving the overall commuting experience for users. The proposed system has the potential to revolutionize public transportation by making it more efficient, reliable, and user-friendly.

Acknowledgements

Glory be to God Almighty, for with Him all things are possible. I am extremely grateful to him for giving me the strength to work on this project in difficult times. Thanks to all my friends and relatives who helped me. I would like to thank Miss Zindove, my supervisor, for her tremendous support, helpful comments, notes, feedback and involvement in the learning process of this research project. I am grateful for all her support during this journey. I was also lucky to have the help of the lectures who wrote me useful suggestions and recommendations. I would also like to thank the University Of Harare Institute Of Technology for providing me with adequate resources and knowledge needed to develop my project. Furthermore, I would also like to express my deep gratitude to my colleagues Adrian Nzvimbo, Christopher Meki, Kudakwashe Koti and classmates for their generous help. Last but not least, I thank my parents, my sister and family for showing me such unwavering love and support. I am so grateful for all the sacrifices you have made to ensure my happiness and success.

Dedication

To all the commuters who rely on public transportation to get to work, school, and other important destinations, i dedicate this real-time bus tracking system. We believe that everyone deserves to have access to reliable and efficient transportation, and we hope that this system will make your daily commute a little bit easier and more convenient. Thank you for choosing to ride with us, and we look forward to serving you for years to come.

DECLARATION

I, Tinotenda Katsamba, declare that this project document is my own work. Submitted in partial fulfillment of the requirements for the award of the Bachelor of Technology in Software Engineering at the Harare Institute of Technology. Not previously submitted for any degree or examination at any other university.

Student's Signature……Date 05/08/2023

(Tinotenda Katsamba)

Supervisor's Signature……Date 05/08/2023

(Miss S Zindove)

CHAPTER 1

There are buses available for passengers traveling to different locations in the cities and towns but not many commuters have complete information about these buses.Complete information regarding number of buses that commute a specified route,bus arrival schedule,bus numbers ,time taken by a bus to reach its destination would greatly help commuters with various routes, track the current location of the bus and give the correct time for the bus to reach its destination.The proposed system is meant to deal with the above stated problems.The system is a mobile application that gives information about buses plying the town of Harare.The application will be developed on an android platform given the popularity and abundance of android devices.

1.1.BACKGROUND

Public transport has been there for many years but due to different circumstances, the system has become unreliable. The inconveniences are arising due to many factors for example congestion delays arrival time for buses in bus terminus which renders fixed time tables useless. As it stands, there are no current working bus time tables in the bus terminus of Harare for local transport. Coupled with the shortage of buses commuting the local routes, commuters are found just waiting for buses during peak hours without an accurate measure of time they have to invest in waiting in long queues. In the end commuters find themselves stranded when the buses do not come and out of desperation resort to private commuters who squander their hard earned money and much needed safety.

Vehicle tracking systems have been implemented in taxi companies like VAYA where one can actually track to see the position of their ride. I am going to apply real time bus tracking using GPS on ZUPCO buses to enable commuters to have a real time access to the buses current location and expected time of arrival. This will in turn help in time management and decongest the ever full bus terminus and long queue because instead of just standing idly, one can carry on with daily activities and only go to board the bus when its actually very near the rank.

In recent years many vehicles are being imported into the country by many people which is resulting in a lot of traffic during peak hours. This situation however could be lessened if the country had a stable and reliable transport system. Considering the fact that many people have mobile smart phones, many people would find confidence in the transport system if they use the app for daily travels.

1.1 Problem statement

Due to the inconsistent and unavailable of bus time tables in bus terminus, commuters have been on many occasions been left stranded.Despite government efforts to ease transport problems by offering subsidized ZUPCO buses on different routes to ease transport costs and provide safe and cheap transport to the commuting public, commuters are still facing a lot of problems in accessing this transport service and in some cases turn to private commuters who overcharge, drive recklessly hence endangering the lives of civilians and in some cases they turn to be criminals who rob ,rape unsuspecting commuters who are desparate to get home especially after a long day work.In most cases commuters spend hours waiting for buses without the slightest hint on when they may come .This in itself is a wasting of time because rather than just checking on your phone to see whether there is a bus nearby to take the commuter home and do something else

productive,one would be stuck in a queue adding to stress and other non communicable tensions to an individual.

1.2 Objectives

i. To design a mobile app
ii. To design a bus tracking device
iii. To send GPS coordinates to customers via text messages
iv. To link the mobile app with the tracking device

1.3 Justification

This project would be viable in this country because of a number of factors.

- Most people in Zimbabwe depend on public transport therefore there is a need for them to be able to pinpoint the location of their preferred transport
- The system can be used as a medium for feedback in terms of complaints, reccomendations and accident reports to the bus companies by the public transport users
- The he system help Passengers save lots of time and also allows them to make decisions quickly for example to wait for the bus or use other means of transport whilst there is still time
- The system dismisses fake bus update news on social media WhatsApp groups.

1.4 Proposed tools

- Arduino IDE
- Java(Springboot)
- Javascript(React and React Native)
- Intelij IDE
- Visual studio code
- Xampp

1.5 Feasibility study

A feasibility study is an analysis that takes all of a project's relevant factors into account including economic, technical, legal, and scheduling considerations to ascertain the likelihood of completing the project successfully. The following are the feasibility factors that were taken into consideration

1.6 Technical feasibility

Technical feasibility aims at analyzing the mobile app from a technical perspective, that is whether the required technology is available or not and also the availability of the required resources. Technical feasibility of the proposed system may be done by just looking at the software and hardware available and analyzing them without necessarily building the system. The technology for the proposed system is shown below:

1.7 Hardware requirements

The hardware and infrastructure that is needed for the implantation of the project. Minimum specification of computer to be used are Intel P4 3 GHz processor speed, 4GB RAM, 500Mb Hard Disk Drive, Windows Operating System (64-bit windows 10) or Ubuntu.

Table 2 Hardware requirements

Description	Quantity	Availability
Laptop	1	Yes
Mobile phone(for testing mobile app)	1	yes
ESP8266 microcontroller	1	yes
GPS module	1	yes
GSM module	1	yes
Sim card	1	yes

As shown on the table above the hardware required to complete this project is readily available, therefore making the proposed system technically feasible.

1.8 Software requirements

Table 3 Software requirements

Description	Quantity	Availability
Windows 10	1	Yes
Visual studio code	1	Yes
Google chrome	1	Yes
JDK	1	Yes
Android studio	1	Yes
Springboot	1	yes
Microsoft SQL server	1	yes
Arduino IDE	1	yes

1.8.1 Economic feasibility

The focus here is on checking if the cost of developing the whole system won't surpass the proposed budget. Cost–benefit analysis showed that it was possible for the development of the application to progress. Most developing software I will be using are open source hence they will be available free of charge. The hardware with required processing power is also already available hence the cost is covered hence the project is economically feasible. For the tracking device the total cost of the components amounts to 30 United States dollars

1.8.2 Project plan

A project plan is a formal document designed to guide the control and execution of a project. A project plan is the key to a successful project and is the most important document that needs to be created when starting any business project.

Table 4 Project plan

Phase	Duration(days)	Starting Date	Ending Date
Requirements analysis and definition	35 days	12 August 2022	8 September 2022
System and software design	120 days	9 September 2022	6 January 2023
Implementation and unit testing	20 days	8 January 2023	28 January 2023
Integration and system testing	24 days	1 February 2023	24 February

1.8.3 Gantt chart

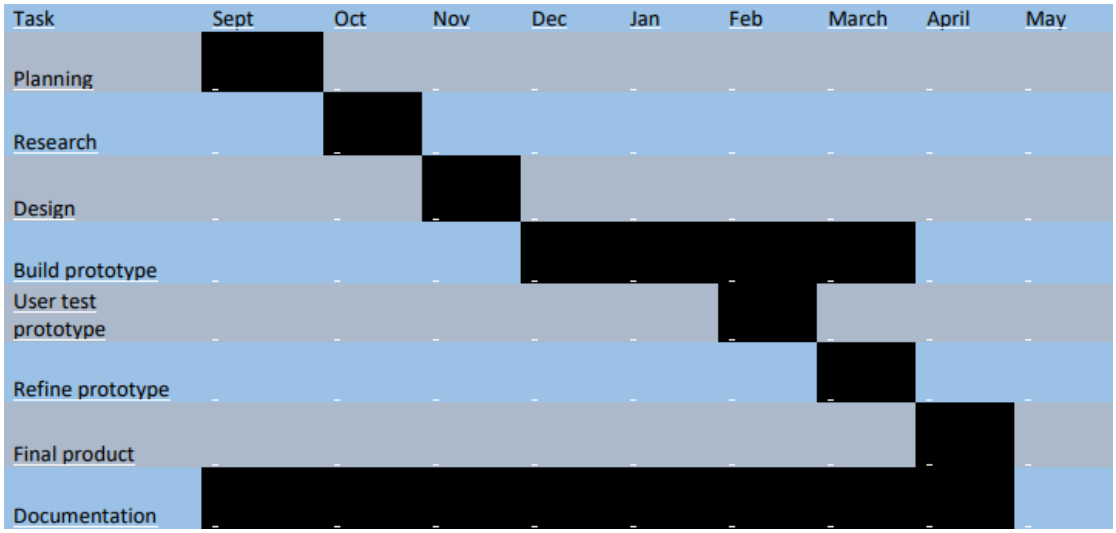


Figure 1 Gantt chart

CHAPTER 2

1.1Introduction

There has been a lot of work around the development of real time tracking systems be it for buses ,taxis or fleet management systems among other things.Here is the literature review of similar studies.

1.2Related work

Authors “Süleyman Eken, Ahmet Sayar” have implemented“ have implemented the system “A smart BusTracking System based on location- aware service and QR code.” In this paper, Bus tracking system, any passenger with Smartphone can scan QR code placed at bus stop to view estimated bus arrival times, current location of the bus. The drawback in this project was that the user had to be physically present at the bus stop to scan the QR code.

Author Shubham Jain et al., created an "Application-based bus tracking system",.This paper is based on a bus tracking system, in which a GPS Tracking application is used to track the bus. GPS technology is user-oriented, to receive the navigating instructions at any instant of time. Here, the location of the bus is received from the satellite and then with the help of cellular networks, it is further processed and sent to the web-server. The coordinates received are processed through Google Maps API. Google Maps API helps to collect data like latitudes and longitudes, locations, etc.The data received is processed in the user’s device, to display the real-time information.

Authors “M. A. Hannan, A. M.Mustapha,A.Hussain and H.Basri” have implemented the system “Intelligent BusMonitoring and Management System” The proposed system uses Artificial intelligence with the help of RFID module which is used in-order to reduce the manual work carried out in the Bus-Management & Monitoring System. In this a RFID is used to track a bus when it crosses the bus stop. Hence the exact location of the bus is not shown , only an approximate location is shown based on the bus stops. In today’s world,accuracy is very important and hence this was the limitation of this project.

Authors ``ManiniKumbhar, MeghanaSurvase, Pratibha MAvdhutSalunk” have implemented “Real Time Web Based Bus Tracking System” The proposed system reduces the waiting time of remote users for buses. A system is used to track the bus at any location at any time. All the current information is stored to the server and it is retrieved to remote users via web based application. This System is a web based system but nowadays people mostly tend to use Android apps since they are more portable and smartphones are used more widely in today’s world. Also a web based system is inconvenient for a user to use on a regular basis while waiting for a bus at the bus stop.

Authors “R.Maruthi, C.Jayakumari” implemented the system “SMS based Bus Tracking System using Open Source Technologies.” A bus tracker application to track a bus using GPS transceiver has been proposed in this paper. The objective of this work is to develop a system that manages and controls the transport using a tracking device to know the scheduled vehicle and the current location of the vehicle via SMS using a GPS tracking device

Authors “Md. Marufi Rahman, Jannatul Robaiat Mou,Kusum Tara, Md. Ismail Sarkar” have implemented the system “Real Time Google Map and Arduino Based VehicleTracking System” using GSM and Arduino coordinates sent by arduino is shown on google maps

1.3Conclusion

From the review of these various papers and studies, it can be concluded that the application of real time bus tracking in a mobile app is feasible and will significantly improve day to day operations. Considerations, suggestions and recommendations made in these papers will thus be used in this project in order to achieve the best capability of the mobile application.

CHAPTER 3: ANALYSIS

1.1 Information gathering techniques

The information gathering techniques were used included:

- Interviews
- Observation
- Questionnaires

1.2 Interviews

This data collection method was used by the developer to get information on the challenges being faced on a daily basis by the commuting public

1.2.1 Advantages of interviews

- Interviews allows the evaluation of facial expression which may strengthen one's facts
- Interviews allows clarity of thoughts as the interviewer can ask further questions on anything that seems unclear
- Interviews are interactive in nature therefore they save time as compared to other methods
- There was a direct communication with individuals who will use the system hence their recommendations and needs were collected first hand

1.2.2 Disadvantages of interviews

- The interviewer maybe biased and may interpret the information incorrectly in favor of his preconceived perceptions
- Some interviewees failed to come through
- Some responses were politically driven therefore biased

1.3 Observations

This technique was used at various bus termini in the Harare CBD. The goal was to observe what commuters go through in order for them to access transportation for the return journey to their respective homes. This was the simplest method of data collection and its major advantage is that everything was there for anyone to see and notice what was going home and the commuters were willing to provide answers like the route and amount of time they had been waiting in the queue and when the last bus left. However the major disadvantage is that not all occurrences open to observation can be observed when the observer is at hand

1.4 Questionnaires

Questionnaires were helpful in instances where the people who were supposed to participate in an interview were not available or occupied at the time. Questionnaires were mainly distributed to the health workers. These questionnaires were filled without any form of supervision and anonymity was ensured however the respondents would sometimes just fill in the answers without understanding the question

1.5 Data analysis

The analysis phase was a crucial activity and the goal was to provide a clear understanding of how the current system works, and its shortfalls. In this chapter, efforts were made to outline possibly all the requirements that the proposed system had to meet. Of concern was the analysis of existing systems, diagrammatic representation, strengths and weaknesses and coming up with possible alternatives. The study helped to produce a system that addressed the shortcomings of those previous existing systems.

1.6 Analysis of existing system

1.6.1 Level zero DFD of existing system

The existing system currently operating in Zimbabwe is a manual or traditional system whereby commuters go to the bus terminus to check if there are buses available for their desired destinations, If a bus happens to be there the commuter enters the bus or stations in line or if the bus is not there may choose to wait or do something else

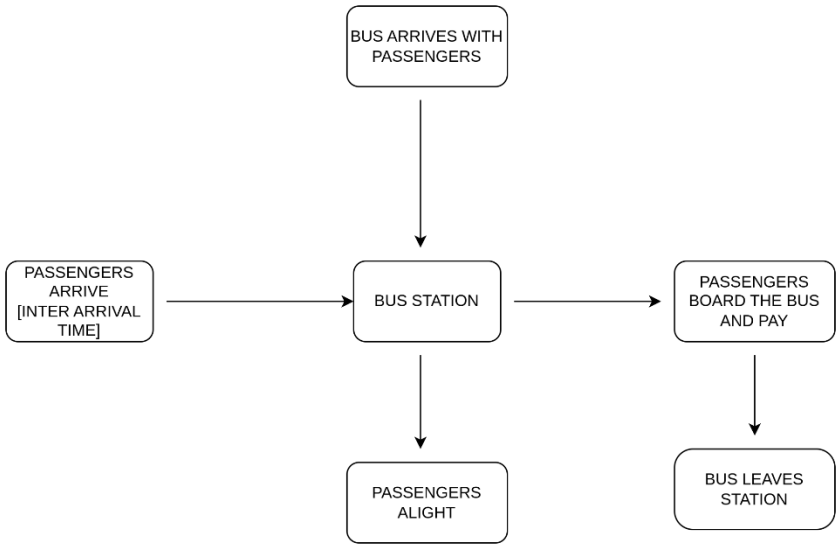


Figure 2 level zero of existing system

1.7 Activity diagram of existing system

An activity diagram is a graphical notation of flow chart which shows flow of one activity to another. Below activity diagram shows the activities, associations and conditions involved in information inquiry

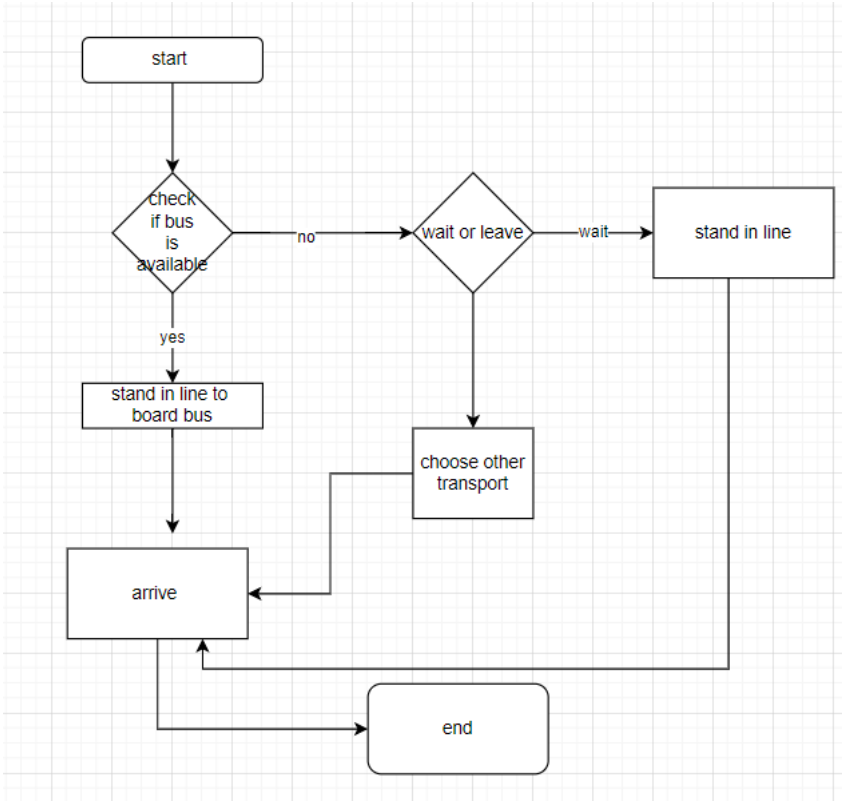


Figure 3 Activity diagram of existing system

1.8 Problems with existing system

- there are no functioning bus time tables for local commuters
- there is a waste of time in meaningless queues where the bus might not come
- there is no way to know when the bus might come
- some areas have fewer buses allocated than others regardless of demand

1.9 Advantages of existing system

- commuters get first hand experience of whether there is a bus or not
- the current system does not suffer from network or power challenges
- the current system is available even for those without mobile phones

1.10 Use case of proposed system

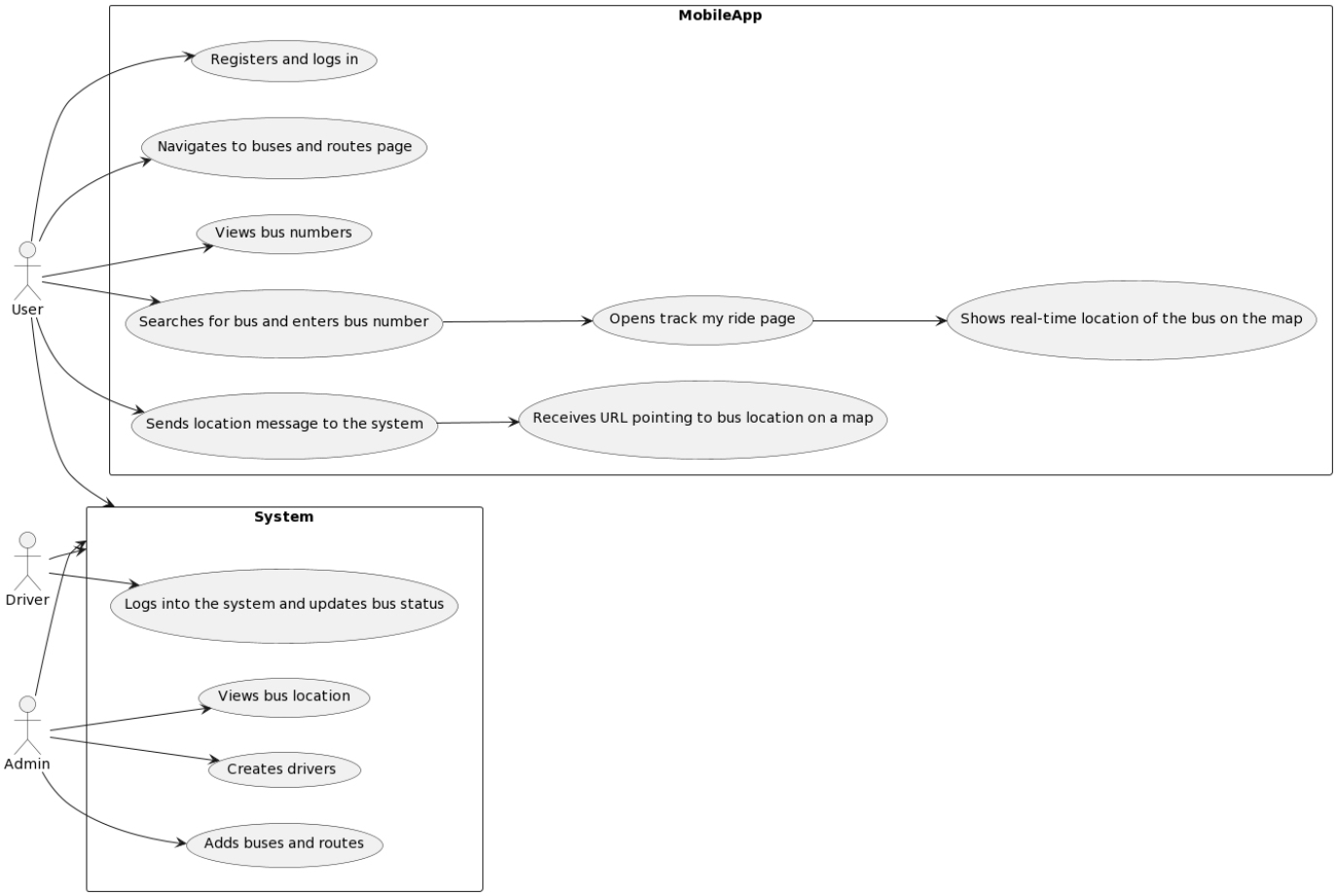


Figure 4 Use case of proposed system

1.10.1 Commuter side

The commuter will signup creating an account by filling in the required information after installing it on their mobile phone.Upon successful login, the user can go to the routes page and select their specific route to see the list of buses which commute the specific route.The user will then click on a selected bus and view it on the real time map to determine its location and approximate time of arrival

1.10.2 Driver side

The driver logs into the system and updates bus status for example trip to town has started, bus is at the garage, bus had an accident and any other information he or she may deem necessary.

1.10.3 Admin side

The admin is the overall overseer of the system. He or she is able to see any bus he/she chooses and add the bus and route, update route or bus information .

1.11 Activity diagram of proposed system

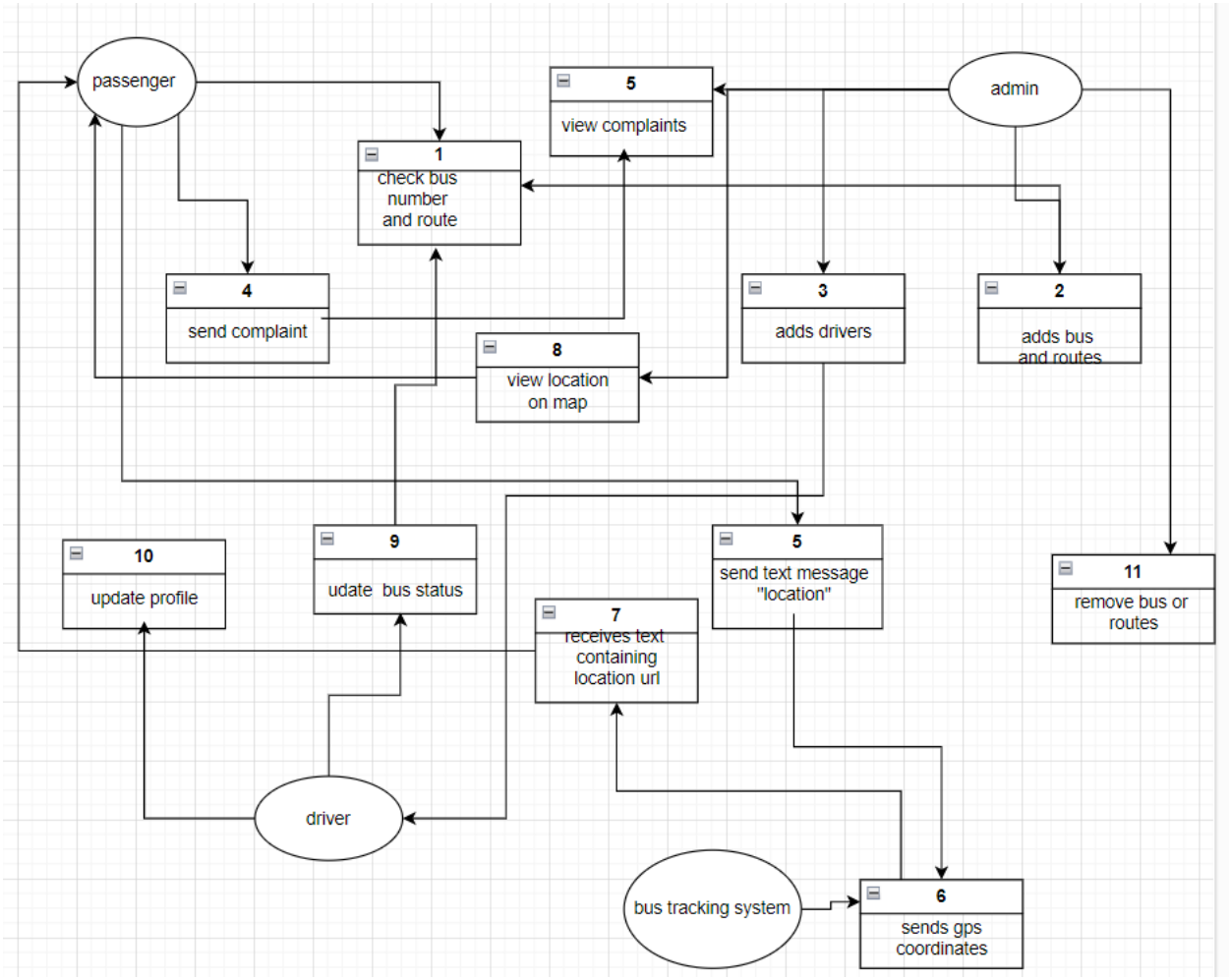


Figure 5 Activity diagram of proposed system

1.11.1 Functional Requirements

Functional requirements refer to the functionalities that must apply to a system and for this proposed system, they are as follows

- the system must be able to show information to user in real time
- the system must be able to process the position data received from the bus positioning module.
- the system must allow user to retrieve information from their mobile device

1.11.2 Non functional requirements

1. Performance: The system should be able to handle a large number of concurrent users and provide real-time updates without significant delays or lag.

2. Reliability: The system should be reliable and available 24/7, with minimal downtime for maintenance or upgrades.

3. Security: The system should be secure, with appropriate measures in place to protect user data and prevent unauthorized access to the system.

4. Scalability: The system should be scalable to accommodate future growth and expansion, such as adding new buses or routes.

5. Usability: The system should be easy to use and intuitive, with clear interface design and navigation.

6. Accessibility: The system should be accessible to users with disabilities, such as providing support for screen readers and other assistive technologies.

7. Compatibility: The system should be compatible with a range of devices and platforms, including desktops, laptops, tablets, and smartphones.

8. Maintainability: The system should be easy to maintain, with clear documentation and well-structured code that is easy to modify or update.

9. Compliance: The system should comply with relevant industry standards and regulations, such as data privacy laws and security standards.

10. Performance under load testing: The system should be able to handle the expected load of concurrent users, and the response time should not degrade as the number of users increases.

.

1.12 User requirements

The user must be able to retrieve the real time estimated bus arrival time of the bus. While waiting in the bus stop or anywhere else, users should be able to access the bus tracking system with their mobile devices. This is the main purpose of the bus tracking system mobile application. The system should provide real time bus tracking with a mapping feature, which means users should be able to view the bus position on a map. The system should be able to send text messages upon requests to customers with positional information with GPS coordinates.With this mapping technique, users will be able to know where the bus is based on the map in real time.

1.13 Evaluation of Alternative Systems

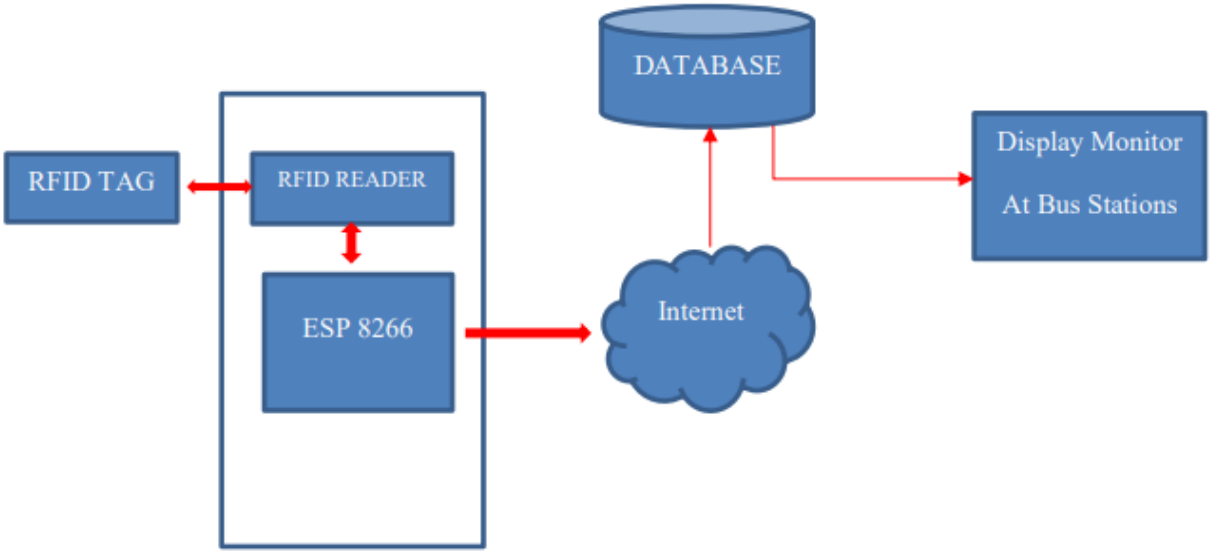


Figure 6 Alternative system

The Alternative Design was not implemented because:

1. The overall system is very expensive to implement because RFID readers and control system will have to be installed at every bus station.
2. The bus station monitors are subject to vandalism by human beings especially in Zimbabwe where the streets are not monitored by CCTVs.
3. The system does not show the real time location of the bus, it only gives approximate bus location e.g. the bus has crossed bus station A.
4. Bus information can only be accessed at bus stations hence Passengers cannot make travelling decisions away from the bus stations.

CHAPTER 4: DESIGN

1.1 System inputs

The bus tracking system will have a few inputs:

- GPS data
- User queries
- User feedback
- Routes information

1.2 System processing

System processing will be hidden from the user and processing includes getting bus coordinates and displaying them on a map

1.3 System output

The output of a real-time bus tracking system can include various types of information that help passengers, transit agencies, and other stakeholders to better manage and optimize the transportation network. Here are some examples of system outputs:

- Real time bus location : The system can display the real-time location of each bus on a map, allowing passengers to track the bus's progress and plan their journey accordingly.
- Estimated time of arrival : The system can provide an estimated time of arrival for each bus at each stop based on its current location and speed. This helps passengers to plan their journeys more effectively and reduces wait times.
- Historical Data analysis : The system can analyze historical data to identify trends in passenger demand, bus utilization, and route performance. This information can be used to optimize the transportation network and improve service delivery.

1.4 Hardware design

This section describes the steps that were taken to design the bus tracking device. The main hardware elements of the tracking device are the GPS module, GSM Module and the microcontroller. The diagram shows the simplified block diagram for the tracking device hardware implementation and how it communicates to satellites and the GSM/GPRS network.

1.4.1 Block diagram of tracking device

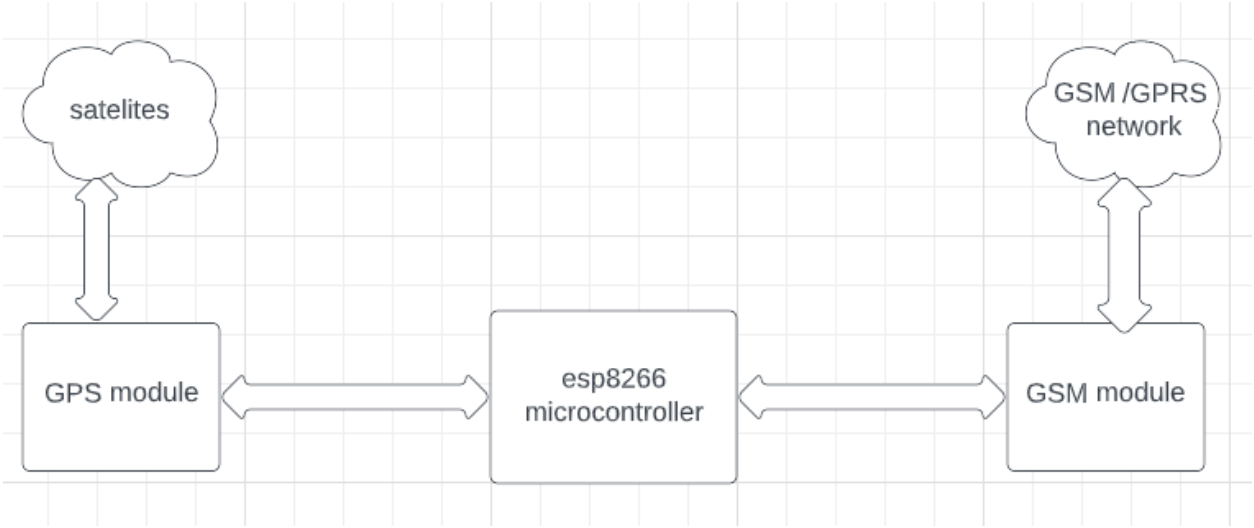


Figure 7 Block diagram of tracking device

1.4.2 Esp8266 Microcontroller

The Esp8266 microcontroller was used to control the hardware components of the hardware design. The main functions of the microcontroller are:

- It powers the NEO-6M GPS module
- It powers the GSM module
- Retrieving longitude and latitude coordinates from the GPS module
- Supplying the GSM module with coordinates to upload

1.4.3 Serial communication

Serial communication was used to exchange data between the Esp8266 Board, GPS Module and the GSM Module. In the design the esp8266 pins 0 and 1 which are the inbuilt pins for serial communication were not used because they were reserved for use by the serial monitor. As a result pins 8,9,10 and 11 were GPS module esp8266 Microcontroller GSM module Tracking device Satellites GSM/GPRS network 42 configured for serial communication. Pins 8and 10 were configured as esp8266 receiver (Rx) while pins 9 and 11 were configured as esp8266 transmitter (Tx). The Figure 3.10 below illustrates serial communication.

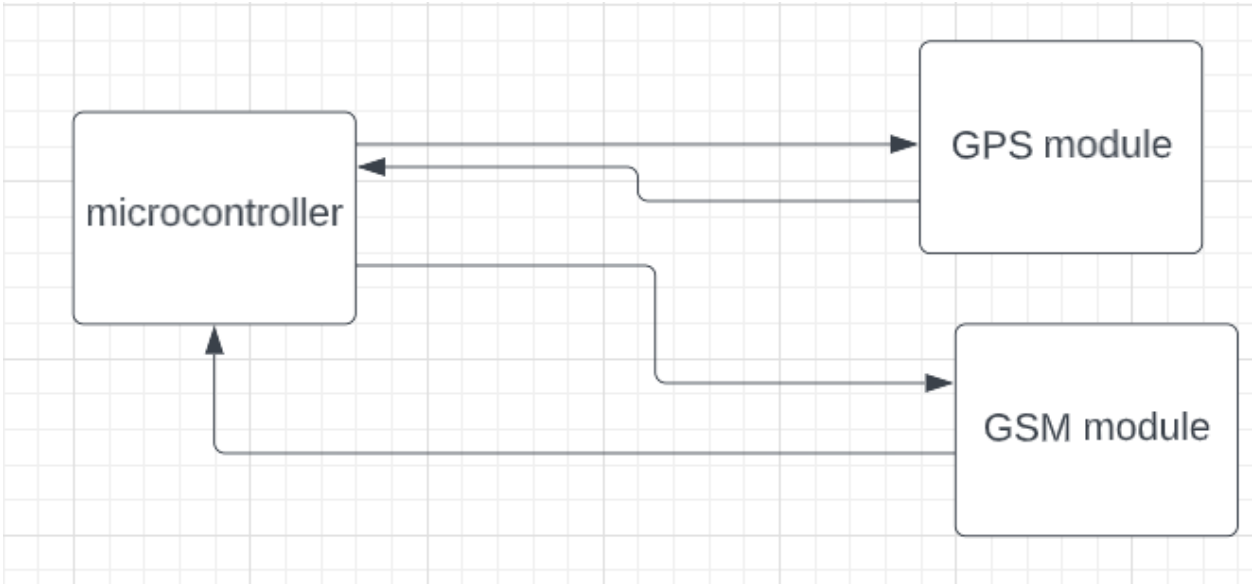


Figure 8 Block diagram of tracking device

1.4.4 GSM SIM800L

GSM SIM 800L is a 12 pin quad-band GSM/GPRS module that can be used to make and receive voice calls, send and receive short message service and upload data to a webserver via GPRS. In this design the SIM 800L module was used to upload the latitude and longitude location coordinates to the website server using the GPRS network feature. The uploaded location coordinates are then stored into the database by a PHP code . This module is relatively power hungry and draw a maximum current of 2A during transmission bursts. The SIM800L was designed without an on board voltage regulator hence the VCC power supply should be strictly between 3.4V to 4.4V. When the module was powered with a 3.3V Arduino power, the module kept shutting down as the Arduino could not match the 2A current requirement of the module. As result the SIM800L module was powered using the 3.7V 4800mAh rechargeable Lithium battery.

1.4.5 Integration of the hardware components

The figure below shows the integration of the GPS, GSM and esp8266 components into the full tracking device powered by a USB cable

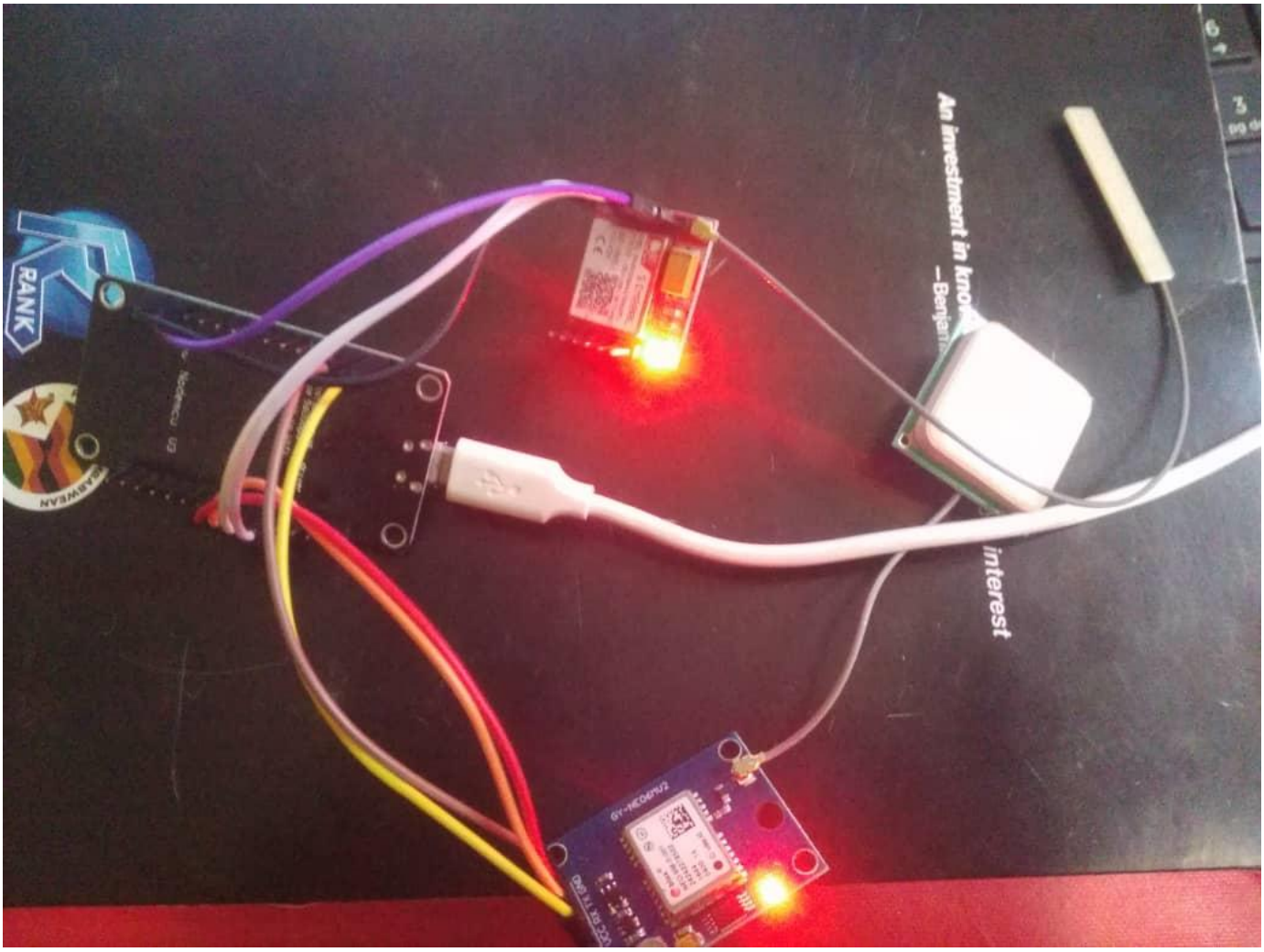


Figure 9 Fully integrated components

1.5 Data flow diagram

Data flow diagrams maps out the movement (flow) of information in a system. It uses defined symbols like rectangle, circles, arrows and short text.

- Circles represent processes
- Rectangles represent external entities
- Arrows show the movement of information
- Short text describe the nature of information being moved

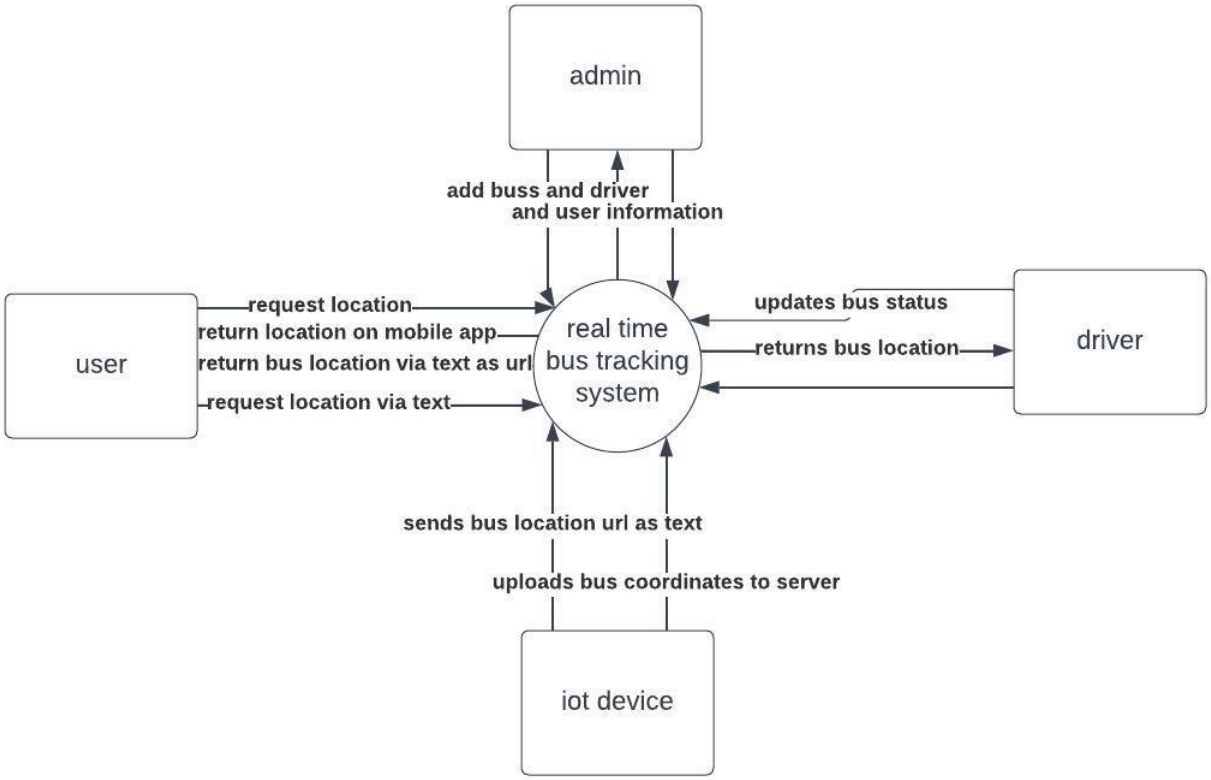


Figure 10 DFD of proposed system

1.6 Context diagram of proposed system

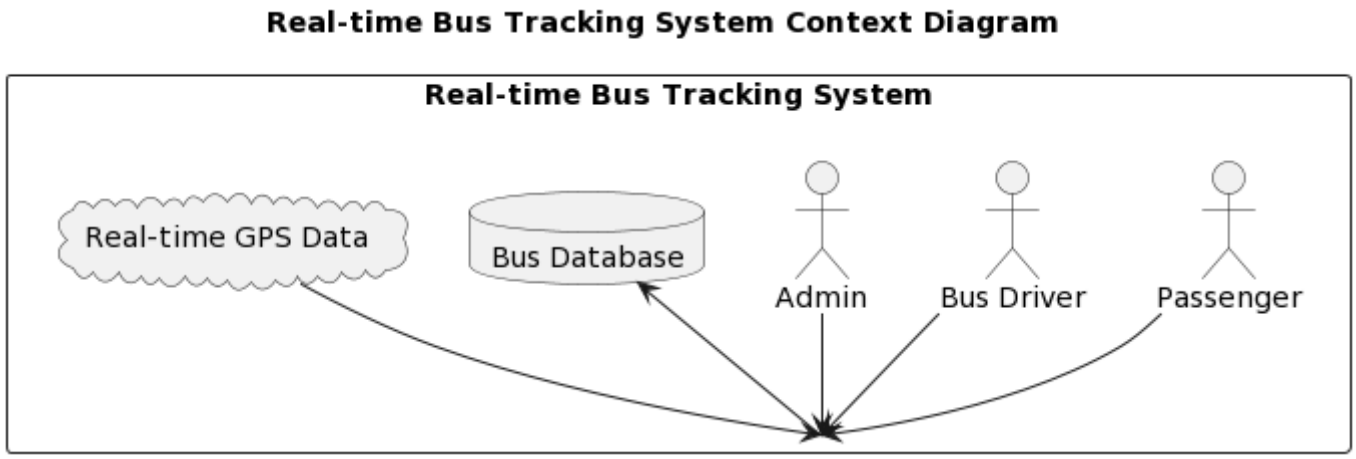


Figure 11 Context diagram of proposed system

1.7 Level one data flow diagram of proposed system

Level one data flow diagram provides much more details on the processes of the system. It breaks down the main processes into smaller sub processes that can be analyzed and improved on a deeper level

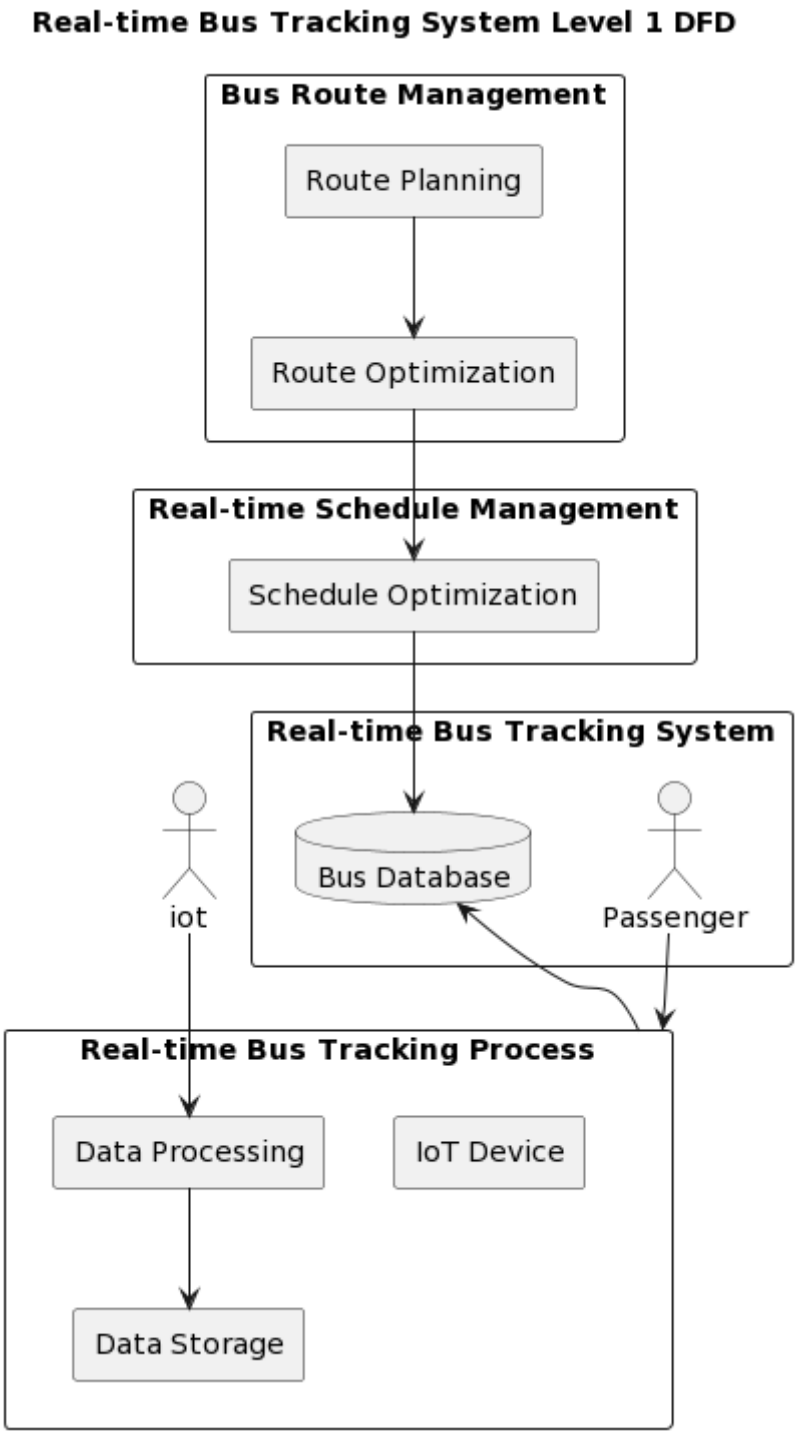


Figure 12 Data flow diagram of proposed system

1.8 Activity diagram

An activity diagram is a graphical notation of flow chart which shows flow of one activity to another. Below activity diagram shows the activities, associations and conditions involved in information inquiry

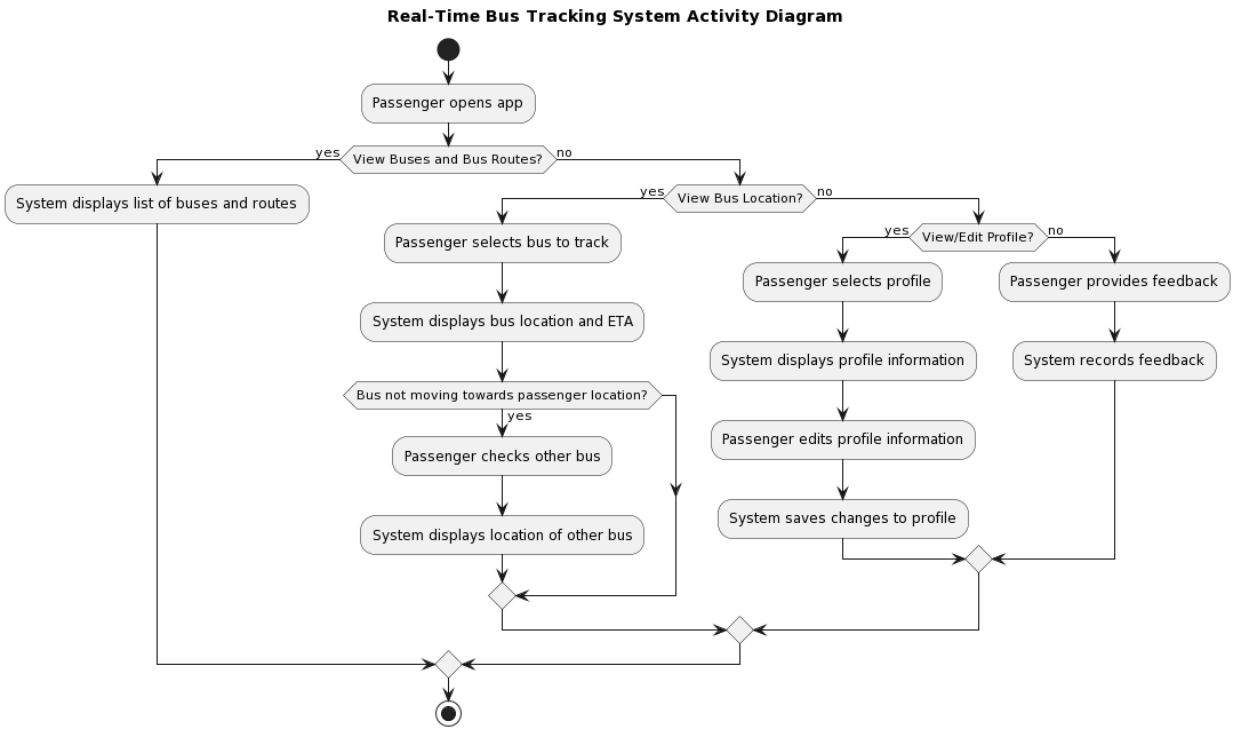


Figure 13 Activity diagram of proposed system

1.9 Architectural design of proposed system

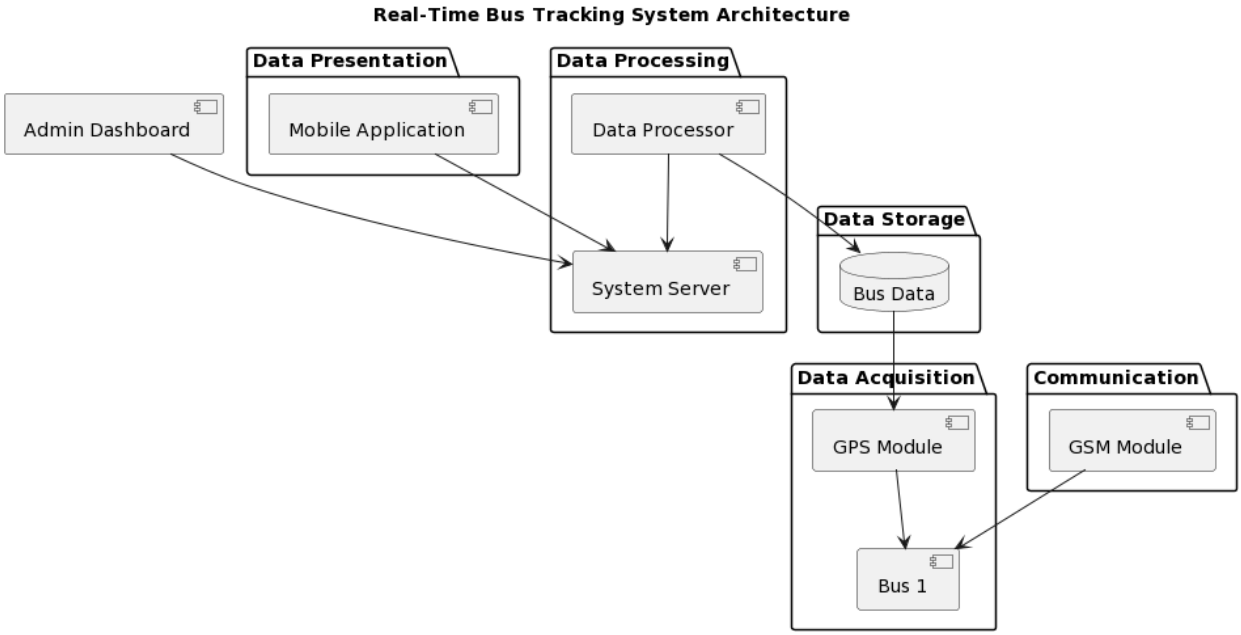


Figure 14 Architectural design of proposed system

1.10 Normalized database

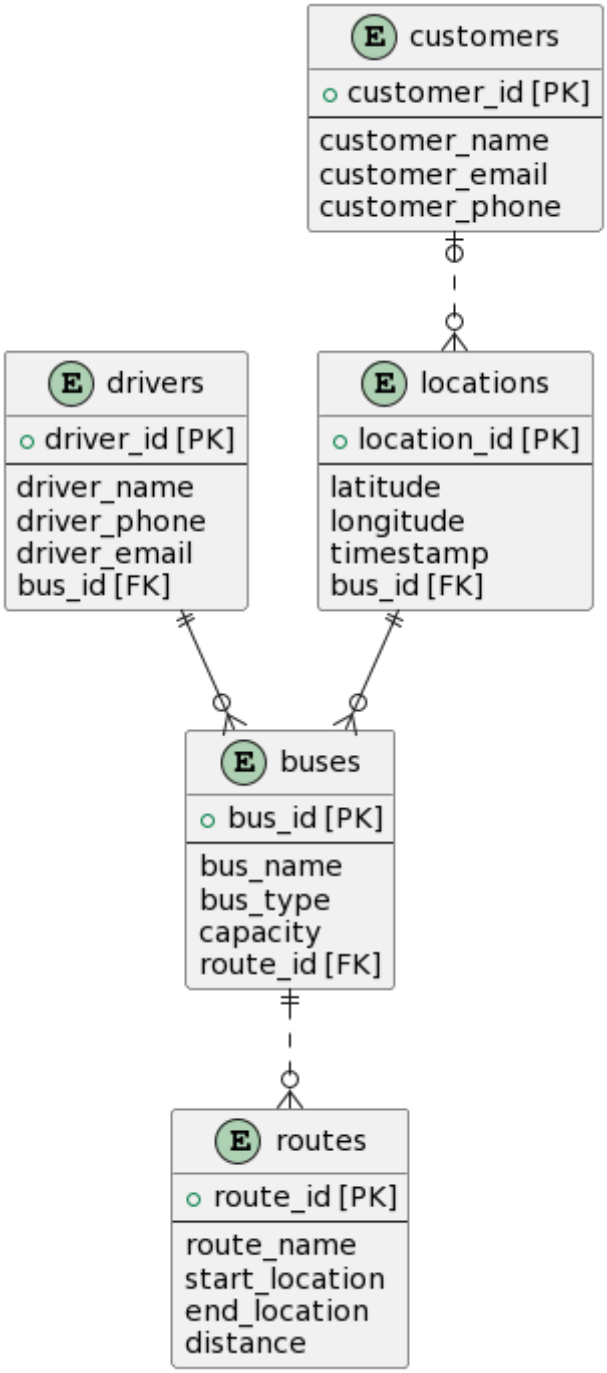


Figure 15 Normalized database of proposed system

1.11 Program design

1.11.1 Class diagram

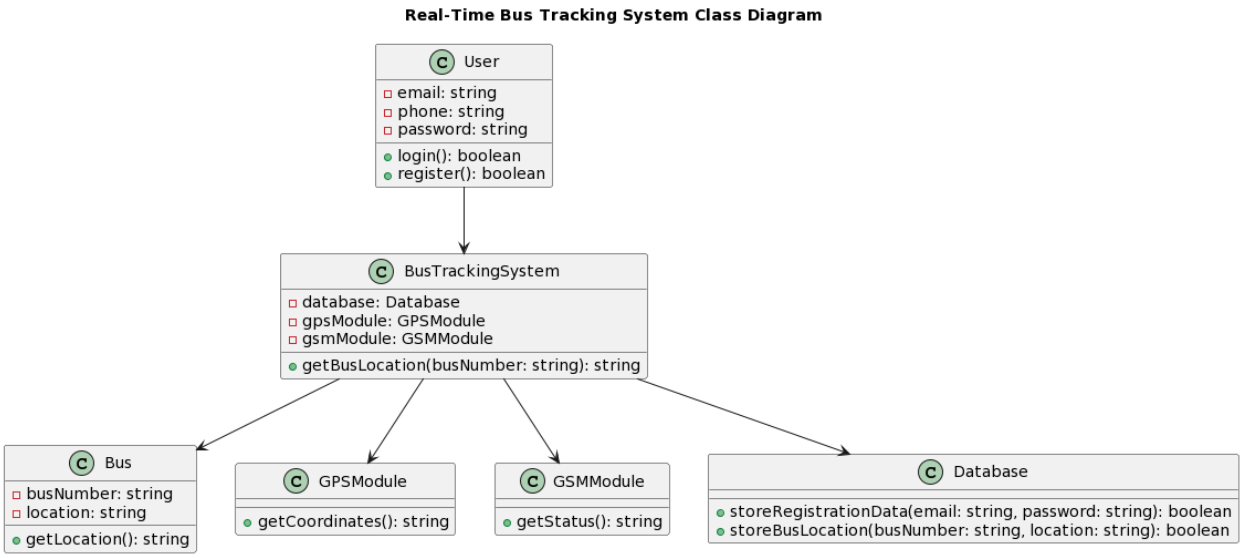


Figure 16 Class diagram

1.11.2 Sequence diagram

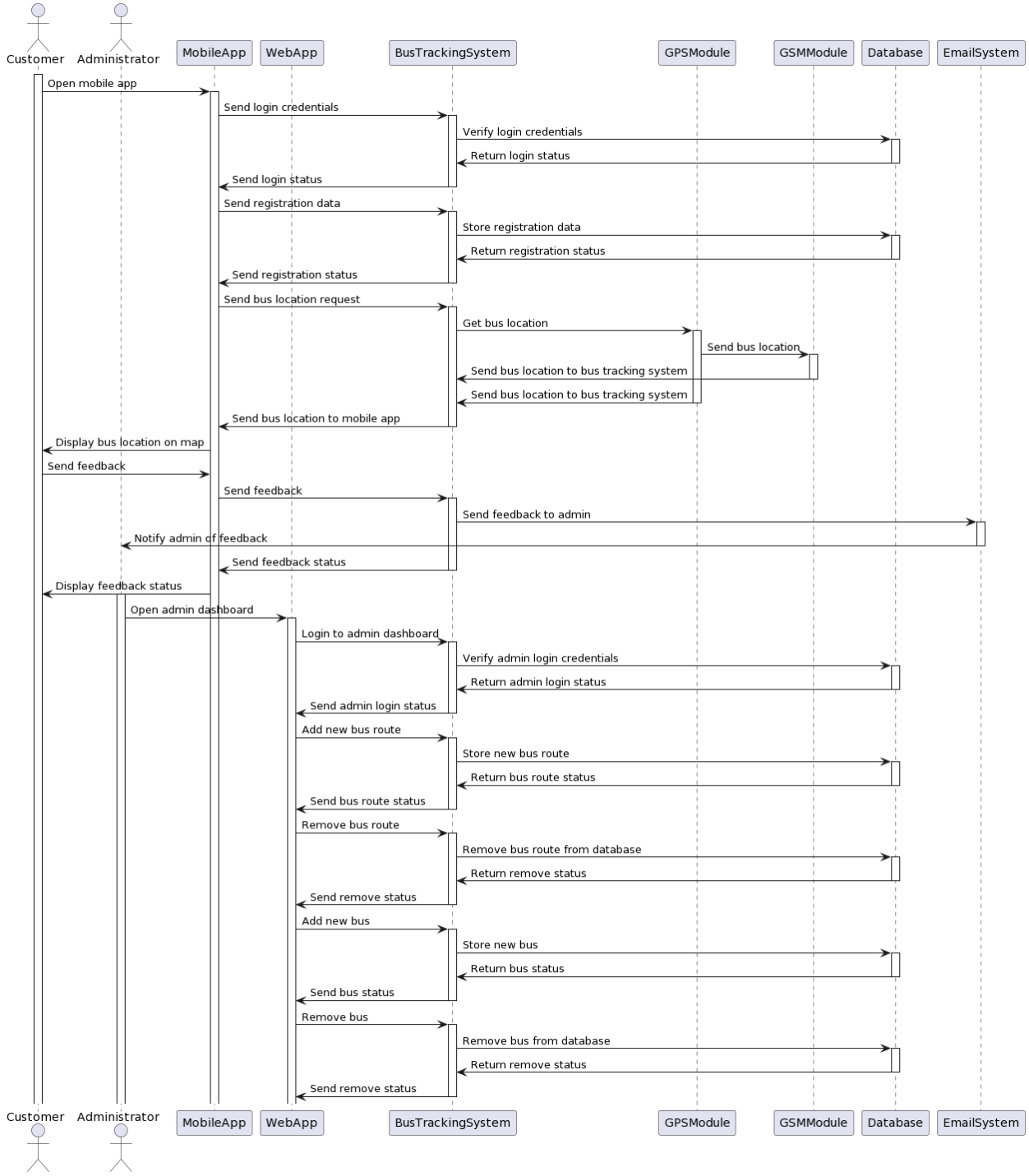


Figure 17 Sequence diagram of proposed system

1.12 Interface design

Sending location text message to receive URL with GPS coordinates

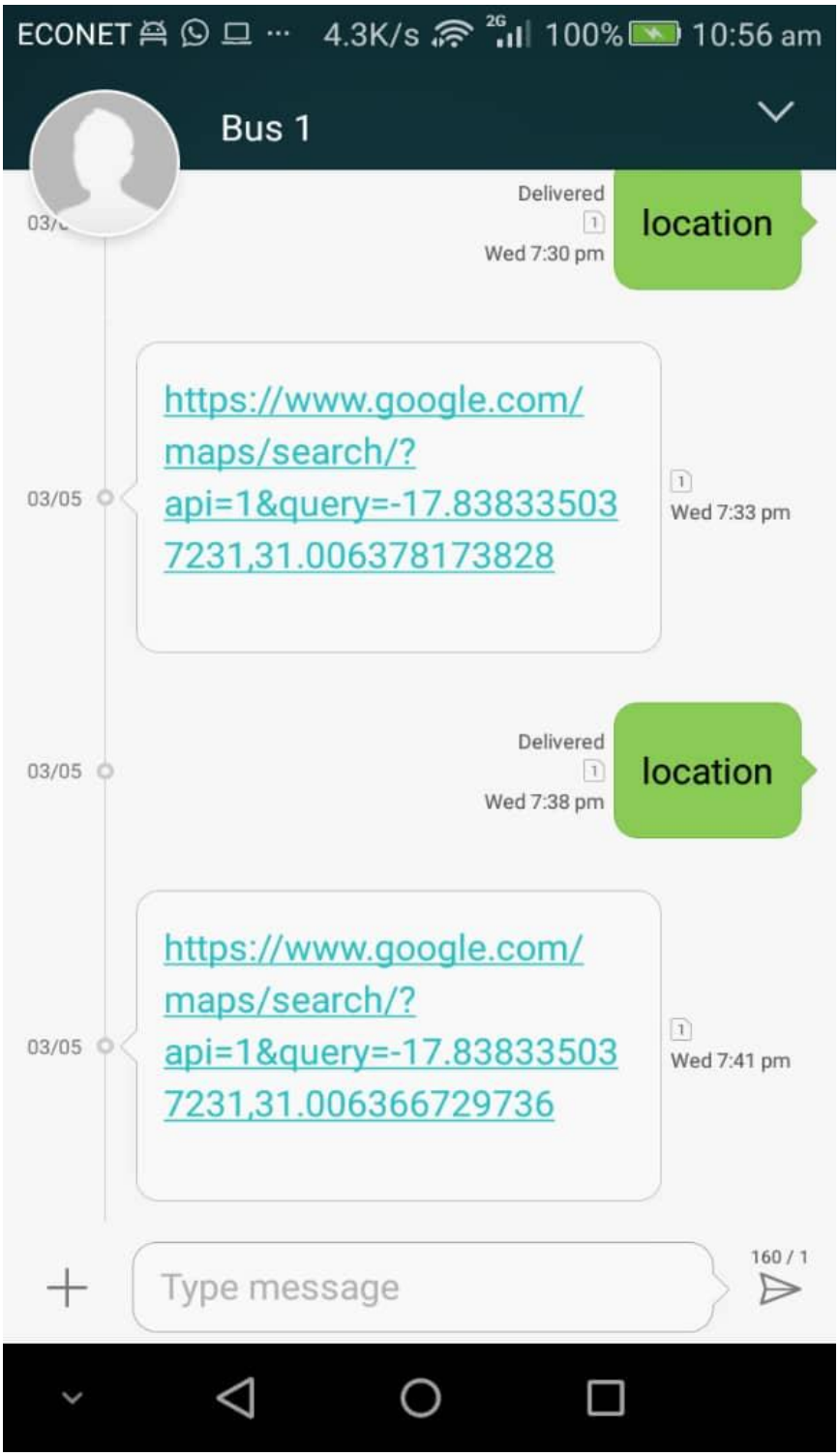


Figure 18 Receiving GPS coordinates via texts

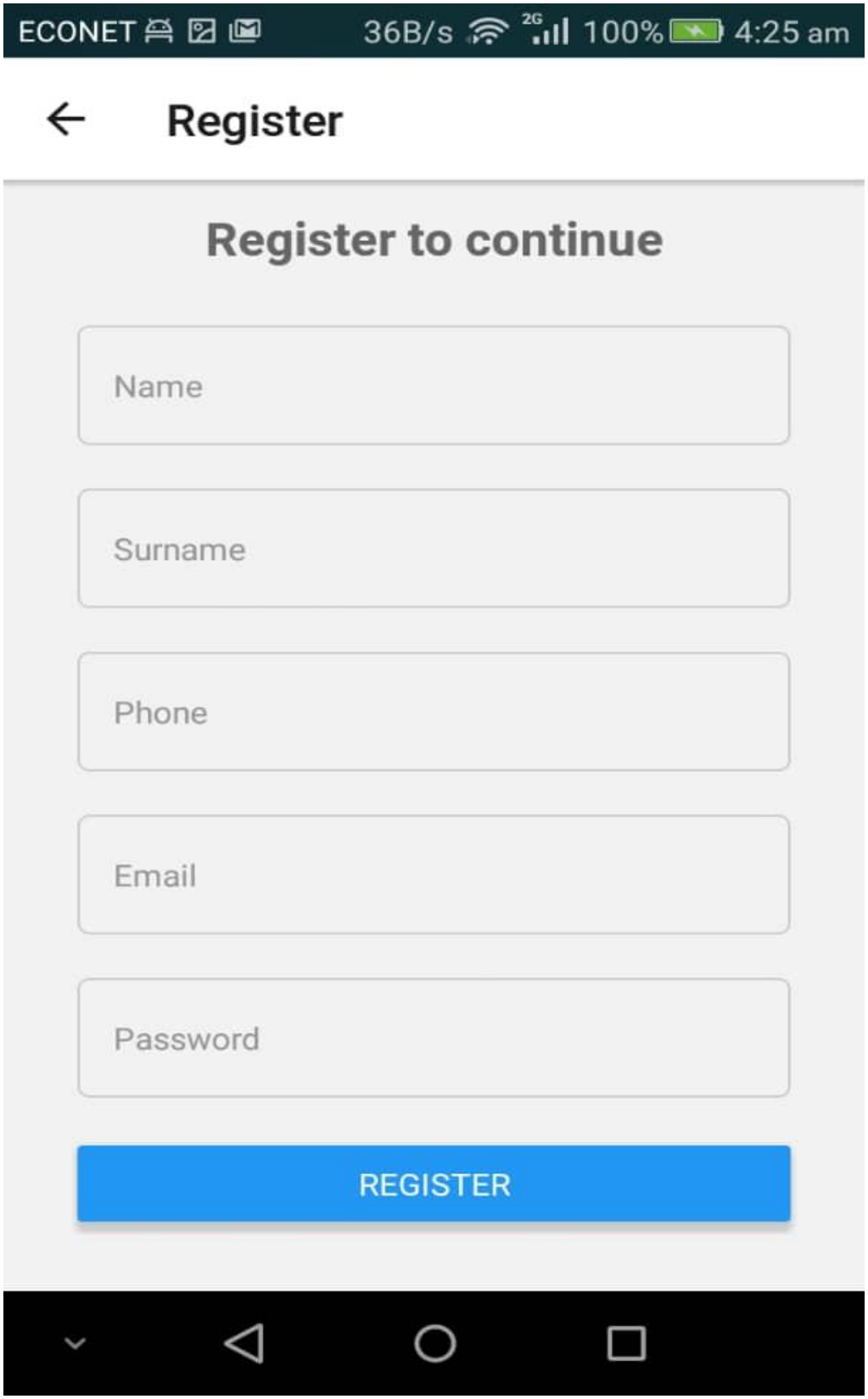


Figure 19User Registration

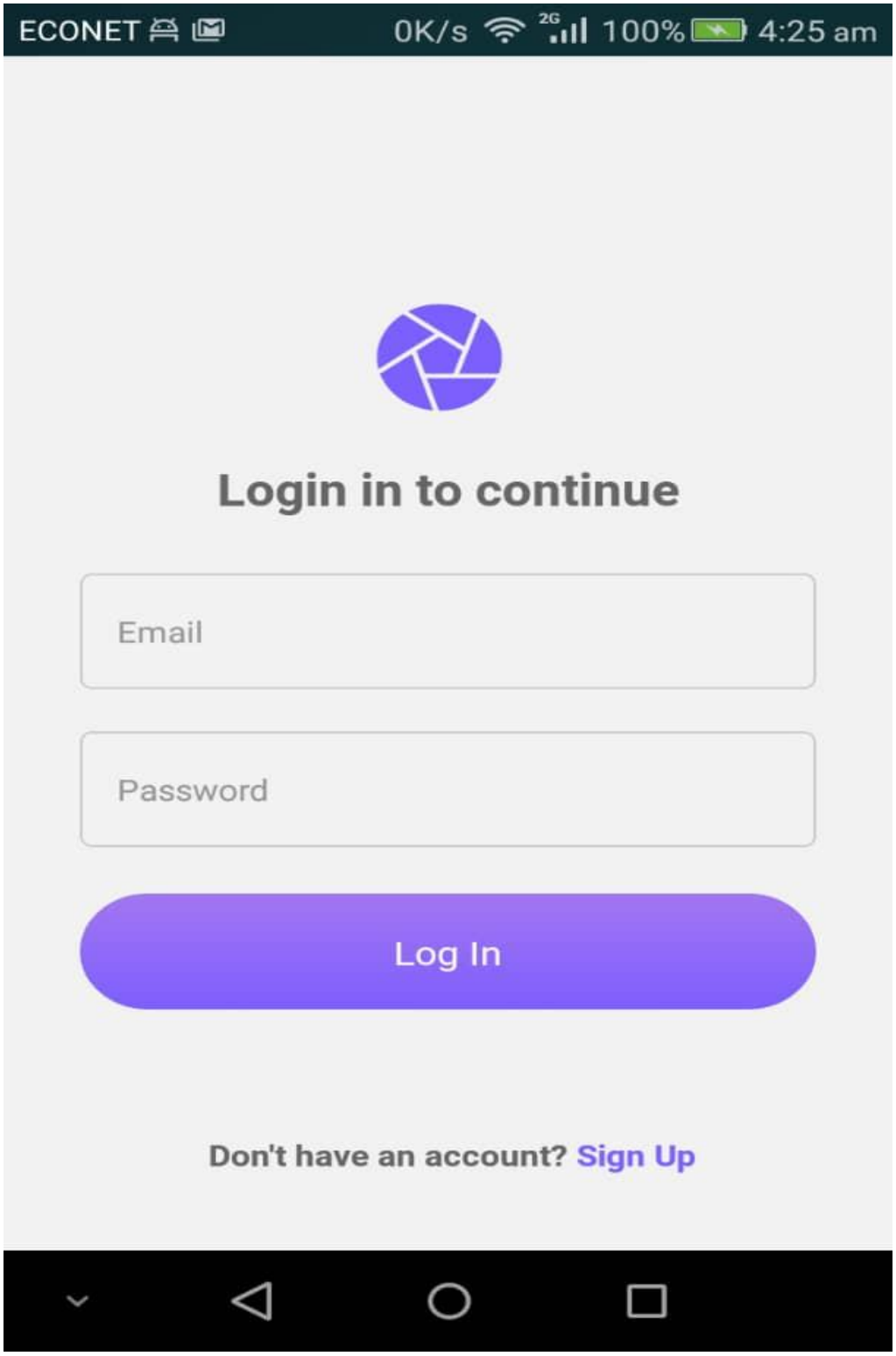


Figure 20User Login

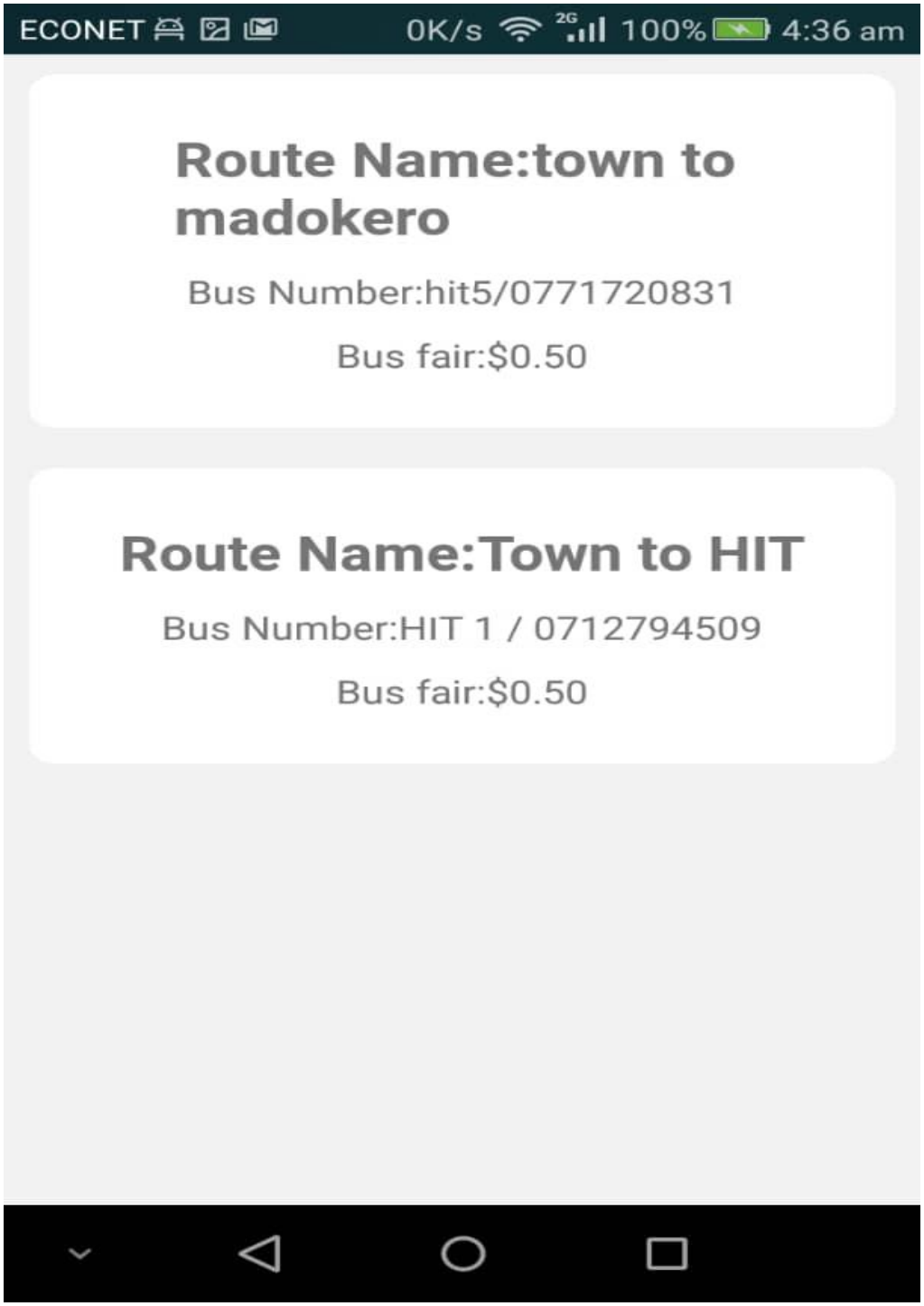


Figure 21View routes , bus numbers and fair

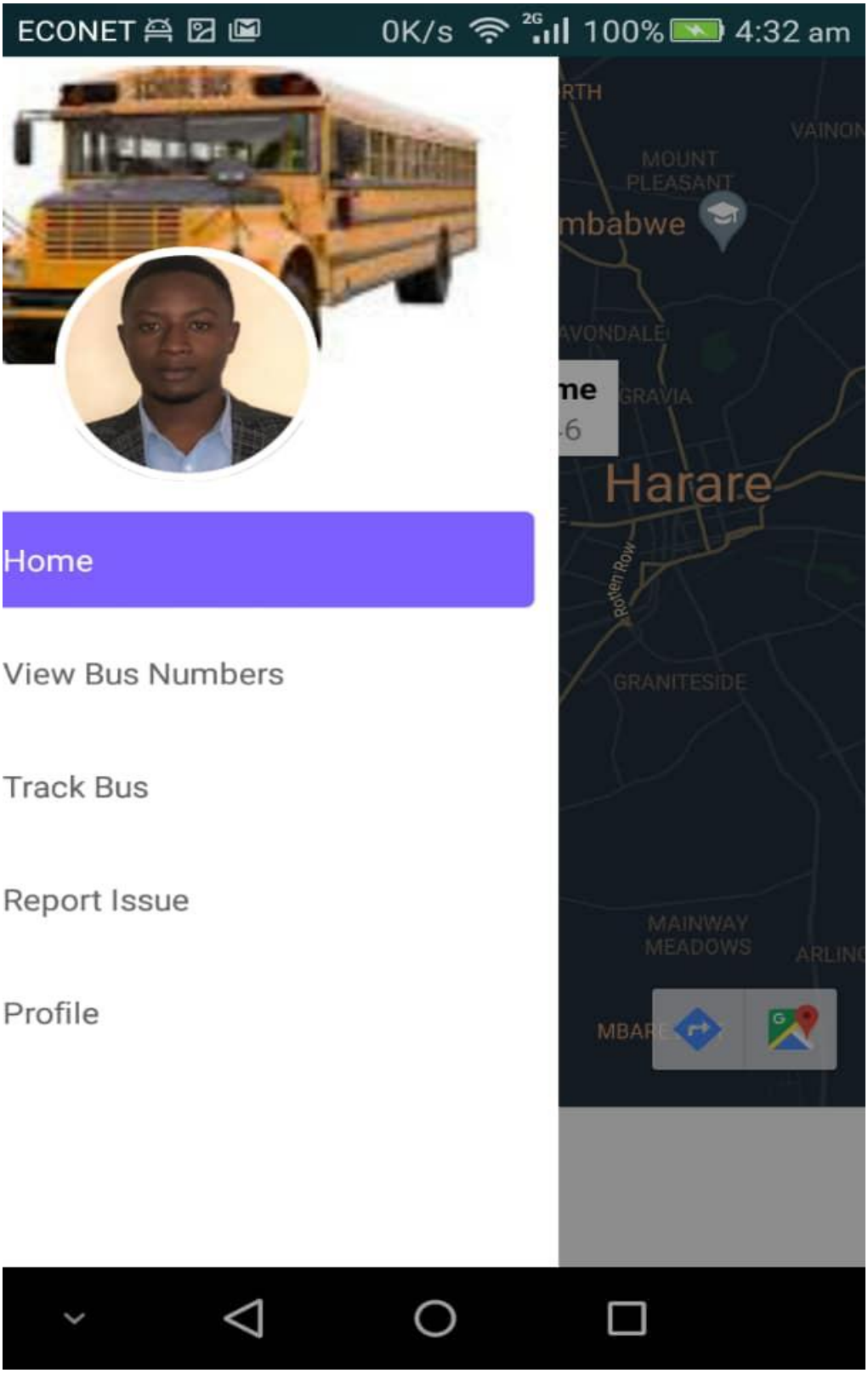


Figure 22User interface on mobile app

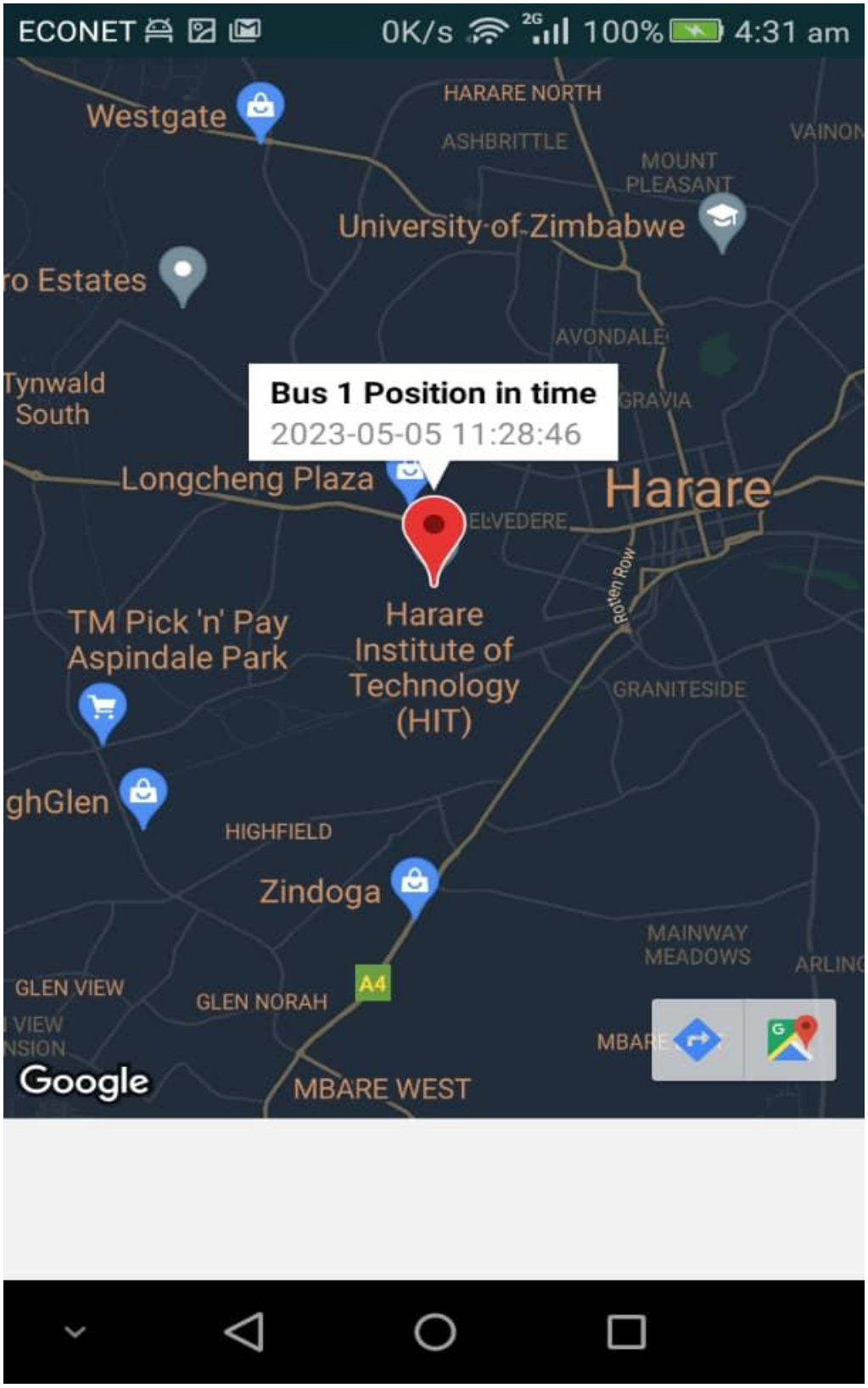


Figure 23View bus position on mobile app

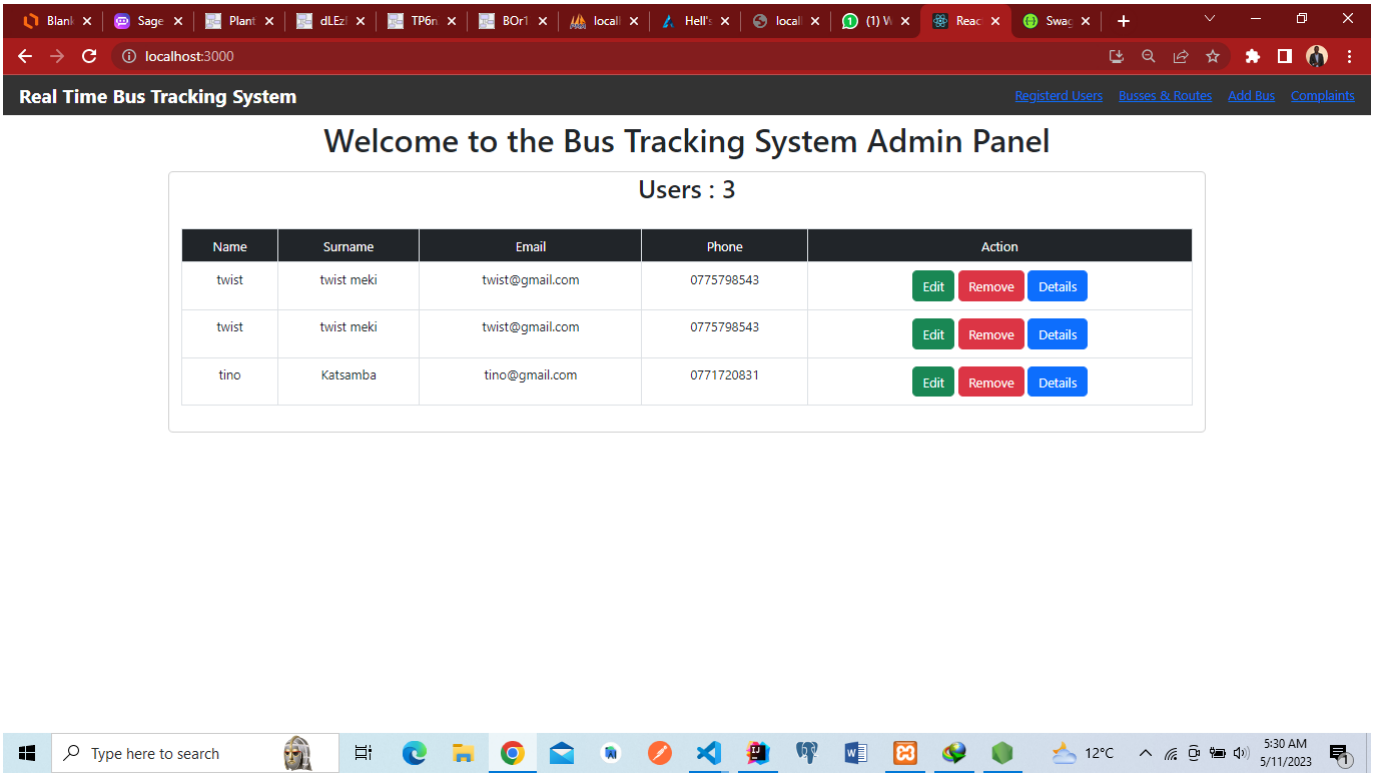


Figure 24 Admin views all users

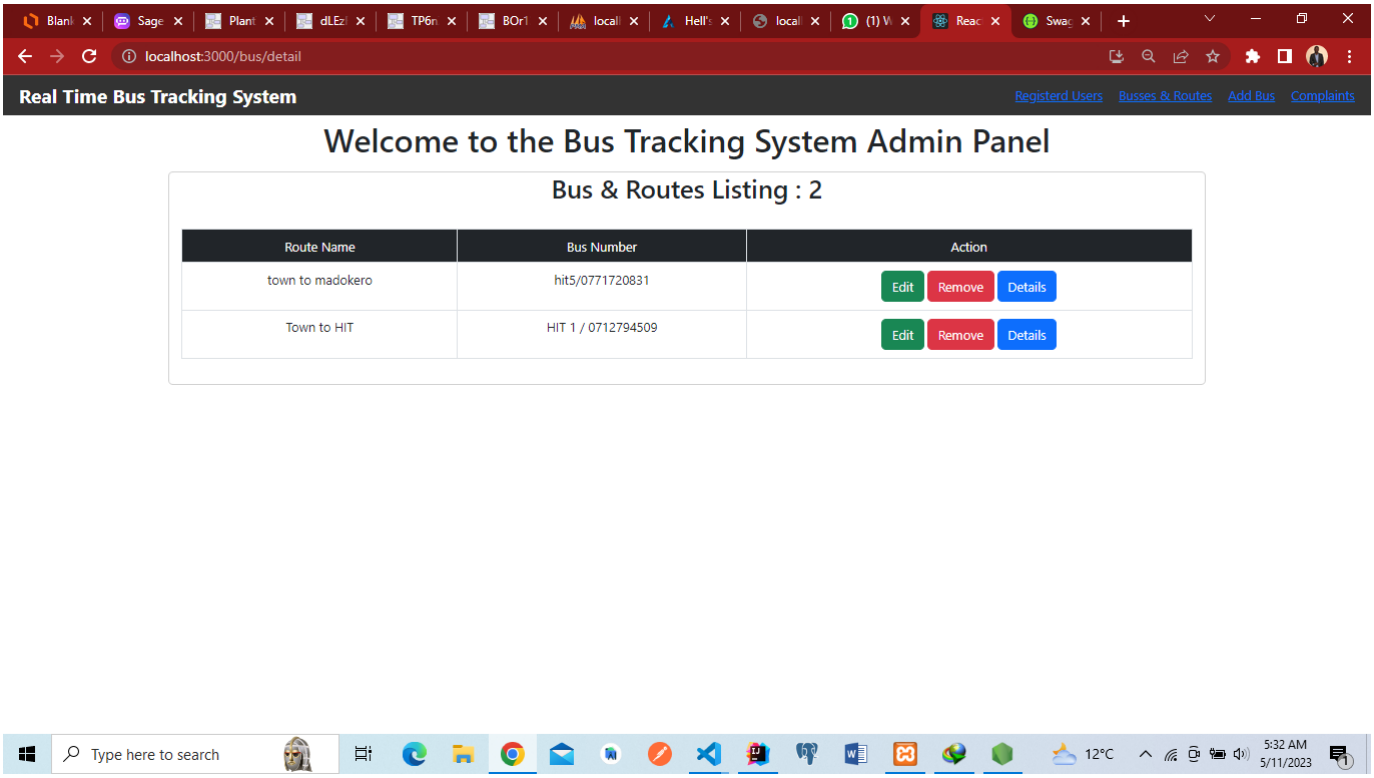
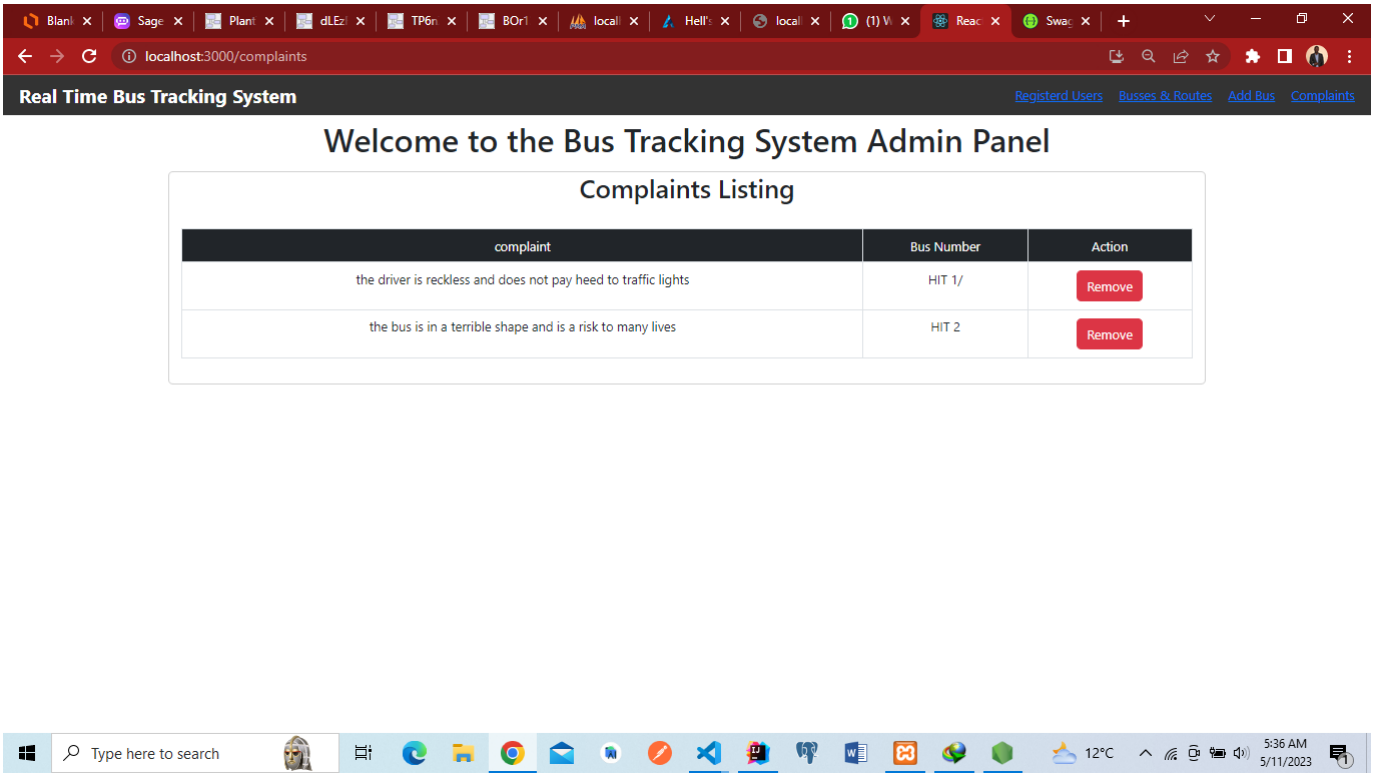


Figure 25 Admin views all buses and routes


localhost:3000/complaints
Real Time Bus Tracking System
Registerd Users
Busses & Routes
Add Bus
Complaints
Welcome to the Bus Tracking System Admin Panel
Complaints Listing
complaint
Bus Number
Action
the driver is reckless and does not pay heed to traffic lights
HIT 1/
Remove
the bus is in a terrible shape and is a risk to many lives
HIT 2
Remove

Type here to search
12°C
5:36 AM
5/11/2023

CHAPTER 5: IMPLEMENTATION AND TESTING

This chapter focuses on the implementation and results from the tests carried out in the project. It encompasses unit, integration and system testing. It includes test cases for functional and non-functional requirements as outlined in chapter 3 of this document

1.1 Sample code of major modules

Table 5 Language tools and platform

Platform	Windows 10
Languages	Java (Springboot) , Javascript(React & react native), C , PHP
Tools	Visual Studio Code, Google chrome, Arduino IDE, Intelij
Methodology	Build and fix

1.1.1 Configuring IOT device in Arduino IDE pseudo code

Import the necessary libraries

(ESP8266WiFi, ESP8266HTTPClient, TinyGPS++, and SoftwareSerial)

Define the variables and objects needed:

- TinyGPSPlus gps
- SoftwareSerial ss(D5, D6)
- SoftwareSerial SIM900(D7, D8)
- String postData
- String link
- char ssid
- char password
- int count
- double SPEED
- float latitude
- float longitude
- int year
- int month

- int date
- int hour
- int minute
- int second
- String date_str
- String time_str
- String lat_str
- String lng_str
- int pm
- String SMS
- String txtmessage
- boolean msgsend = false

Initialize the serial communication:

- Serial.begin(9600)
- SIM900.begin(9600)
- ss.begin(9600)
- delay(2000)

Connect to WiFi:

- WiFi.begin(ssid,password)
- while (WiFi.status() != WL_CONNECTED)

```
  {
    delay(500)
    Serial.print(".")
  }
```

- Serial.println("")
- Serial.println("WiFi connected")
- Serial.println(WiFi.localIP())

Handshake with SIM900:

- SIM900.println("AT")
- updateSerial()
- SIM900.println("AT+CMGF=1")
- updateSerial()
- SIM900.println("AT+CNMI=1,2,0,0,0")
- updateSerial()

In the loop, read GPS data:

- while (ss.available() > 0)
  if (gps.encode(ss.read()))
    if (gps.location.isValid())
      Get the latitude, longitude, and speed data from the GPS object
      Format the SMS message with the Google Maps URL
      Send the SMS message using the SIM900 module
      Send the location data to the server using an HTTP POST request
      Check for successful connection and response from the server

Define the sendSMS function that sends an SMS message using the SIM900 module:

- SIM900.println("AT")
- updateSerial()
- SIM900.println("AT+CMGF=1")
- updateSerial()
- SIM900.println("AT+CMGS=\"+263771720831\"")
- updateSerial()
- SIM900.print(message)
- updateSerial()
- SIM900.write(26)
- delay(5000)

Define the updateSerial function that forwards data received by the SIM900 module to the serial port:

- delay(500)
- while(SIM900.available())
  {
    Serial.write(SIM900.read())
  }

Define the handleSerial function that receives and handles serial data from the SIM900 module:

- Initialize a buffer with a size of 32 characters
- Read the incoming serial data character by character
- If a new line character is received, check if the buffer contains a valid message, and handle it accordingly
- Otherwise, append the received character to the buffer
- If the buffer is full, discard the received character

Define the handle Received Message function that handles the received message:

- If the message is "location", send the SMS message

1.1.1.1 PHP code to upload coordinates to server

```
<?php

if(isset($_POST['lat']) && isset($_POST['lng']))
{
$latt= $_POST['lat'];
$long = $_POST['lng'];

mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
$mysqli = new mysqli("localhost", "root", "", "bus_tracker");


    /* create a prepared statement */
```

```
    $stmt = $mysqli->prepare("INSERT INTO `records` (`lattitude`,`longitude`) VALUES (?,?)");
    /* bind parameters for markers */
    $stmt->bind_param("ss", $latt, $long);
    /* execute query */
    $stmt->execute();

    echo "records updated";

}

else{
    echo "Connection failed";

}
```

1.1.1.2 Real time bus location display on a map

```
// Import required modules
import React, { useEffect, useState } from 'react';
import { SafeAreaView, StyleSheet, View, Dimensions } from 'react-native';
import MapView, { Marker } from 'react-native-maps';
import axios from 'axios';

const Maps = () => {
  // Set initial state for latitude, longitude, and date
  const [lattitude, setLattitude] = useState(-67.842929840088);
  const [longitude, setLongitude] = useState(31.010303497314);
  const [date, setDate] = useState(null)

  useEffect(() => {
```

```
  // Make an API call to get the latest coordinates
  axios.get('http://192.168.32.99:8080/coordinate/all-coordinates')
    .then(response => {
      let a = response.data
      console.log(a[a.length -1])
      let b = a[a.length -1];
      // Update the state with the latest coordinates
      setLattitude(parseFloat(b.lattitude));
      setLongitude(parseFloat(b.longitude));
      setDate(b.date);
    })
    .catch(error => {
      console.log(response.data)
    });
}, []);

return (
  // Set up the SafeAreaView and View components for the map
  <SafeAreaView style={{flex: 1}}>
    <View style={ {height: Dimensions.get('window').height - 100}}>
      <MapView
        style={styles.mapStyle}
        initialRegion={{
          latitude: -17.842929840088,
          longitude: 31.010303497314,
          latitudeDelta: 0.0922,
          longitudeDelta: 0.0421,
        }}
        customMapStyle={mapStyle}>
        {/* Add a marker for the latest coordinates */}
```

```
          <Marker
            draggable
            coordinate={{
              latitude: lattitude,
              longitude: longitude,
            }}
            onDragEnd={e => alert(JSON.stringify(e.nativeEvent.coordinate))}
            title={'Position in time'}
            description={date}
          />
        </MapView>
      </View>
    </SafeAreaView>
  );
};
```

```
package com.example.capstone.controller;

import com.example.capstone.model.Records;
import com.example.capstone.service.RecordsService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

@RestController
@RequiredArgsConstructor
@RequestMapping("/coordinate")
public class RecordsController {
    private final RecordsService recordsService;

    //find by id
    @GetMapping("/get-coordinates/{id}")
    public ResponseEntity<Records> findById(@PathVariable("id")String recordsId){
        return new ResponseEntity<>(recordsService.findById(recordsId), HttpStatus.OK);
```

```
    }
    //list all coordinates
    @GetMapping("/all-coordinates")
    public ResponseEntity<List<Records>> allRecords(){
        return new ResponseEntity<>(recordsService.listAllRecords(), HttpStatus.OK);
    }
}
```

```
package com.example.capstone.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@AllArgsConstructor
@NoArgsConstructor
@Data
@Builder
public class Coordinates {
    @Id
    @GeneratedValue(strategy = GenerationType.UUID)
    private String id;
    private String lattitude;
    private String longitude;
    private String Date;
}
```

```
package com.example.capstone.controller;

import com.example.capstone.model.UserDetails;
import com.example.capstone.service.UserService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
```

```
import java.util.Optional;

@RestController
@RequiredArgsConstructor
@RequestMapping("user")
@CrossOrigin(origins = "*")
public class UserController {
    private final UserService userService;

    //save users
    @PostMapping("save")
    public ResponseEntity<UserDetails> saveUser(@RequestBody UserDetails user){
        return new ResponseEntity<>(userService.saveUser(user), HttpStatus.CREATED);
    }

    //LOGIN
    @PostMapping("/authenticate")
    @ResponseStatus(HttpStatus.OK)
    public String authenticate(@RequestParam String email, @RequestParam String password) {
        return userService.authenticate(email, password);
    }

    @GetMapping("/login/{email}/{password}")
    public Optional<UserDetails> login(@PathVariable String email, String password) throws Exception {
        UserDetails email1 = userService.login(email,password);
        System.out.println(email1);
        return Optional.ofNullable(email1);
    }

    //get user by id
    @GetMapping("get-by-id/{id}")
    public ResponseEntity<Optional<UserDetails>> findbyId(@PathVariable("id") String userId){
        return new ResponseEntity<>(userService.findById(userId),HttpStatus.FOUND);
    }

    //get all users
    @GetMapping("all-users")
    public ResponseEntity<List<UserDetails>> allUsers(){
        return new ResponseEntity<>(userService.listAllUsers(),HttpStatus.OK);
    }

    //delete
    @DeleteMapping("delete-by-id")
    public ResponseEntity<String> deleteById(@PathVariable String id){
        return new ResponseEntity<>(userService.deleteById(id),HttpStatus.OK);
```

```
    }
}
```

```
import { useEffect, useState } from "react";
import { Link, useNavigate } from "react-router-dom";

const BusListing = () => {
    const [empdata, empdatachange] = useState(null);
    const navigate = useNavigate();

    const LoadDetail = (id) => {
        navigate("/bus/detail/" + id);
    }
    const LoadEdit = (id) => {
        navigate("/employee/edit/" + id);
    }
    const Removefunction = (id) => {
        if (window.confirm('Do you want to remove?')) {
            fetch("http://localhost:8080/busses/delete-bus" + id, {
                method: "DELETE"
            }).then((res) => {
                alert('Removed successfully.')
                window.location.reload();
            }).catch((err) => {
                console.log(err.message)
            })
        }
    }
```

```
useEffect(() => {
    fetch("http://localhost:8080/busses/all-busses").then((res) => {
        return res.json();
    }).then((resp) => {
        empdatachange(resp);
        console.log(empdata)
    }).catch((err) => {
        console.log(err.message);
    })
}, [])
return (
    <div className="container">
        <div className="card">
            <div className="card-title">
                <h2>Bus & Routes Listing</h2>
            </div>
            <div className="card-body">
                {/* <div className="divbtn">
                    <Link to="create" className="btn btn-success">Add New (+)</Link>
                </div> */}
                <table className="table table-bordered">
                    <thead className="bg-dark text-white">
                        <tr>
                            <td>Route Name</td>
                            <td>Bus Number</td>
                            {/* <td>Fair</td> */}
                        </tr>
```

```
                </thead>
                <tbody>

                  {empdata &&
                    empdata.map(item => (
                      <tr key={item.id}>
                        <td>{item.bus_route_name}</td>
                        <td>{item.bus_number}</td>
                        {/* <td>{item.fair}</td> */}

                      </tr>
                    ))
                  }

                </tbody>

              </table>
            </div>
          </div>
        </div>
    );
}

export default BusListing;


import React, { useState, useEffect } from 'react';
import { StyleSheet, View, Text } from 'react-native';
import axios from 'axios';
```

```
const Card = ({ item }) => {
  return (
    <View style={styles.card}>
      <Text style={styles.title}>{item.name}</Text>
      <Text style={styles.description}>{item.description}</Text>
      {/* other card content */}
    </View>
  );
};

const ViewComplaints = () => {
  const [items, setItems] = useState([]);

  const fetchItems = async () => {
    try {
      const response = await axios.get('http://example.com/api/items');
      setItems(response.data);
    } catch (error) {
      console.log(error);
    }
  };

  useEffect(() => {
    fetchItems();
  }, []);

  return (
    <View style={styles.container}>
      {items.map((item) => (
        <Card key={item.id} item={item} />
```

```
      ))}
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  card: {
    backgroundColor: '#fff',
    borderRadius: 10,
    padding: 10,
    margin: 10,
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.2,
    shadowRadius: 2,
    elevation: 2,
  },
  title: {
    fontSize: 18,
    fontWeight: 'bold',
  },
  description: {
    fontSize: 16,
  },
  // other styles
```

});

export default ViewComplaints;

1.2 Software testing

Software testing is a critical process that ensures that the software product meets the intended requirements and specifications. It involves executing the software components and systems to identify any errors, bugs, or defects that may impact its functionality, usability, and performance. Software testing is performed throughout the software development lifecycle to ensure that the product is of high quality and meets the user's needs.

1.2.1 White Box Testing

White box testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

White box testing for the Real Time Bus Tracking system involves

1. Unit testing: This involves testing individual components like the GPS, GSM and esp8266 microcontroller on their own
2. Integration testing: this involves testing whether the GPS, GSM and microcontroller are compatible with each other
3. Code coverage testing: This involves measuring the percentage of code that is executed during testing to ensure that all code paths are covered and that there are no unreachable code segments.
4. Security testing: This involves testing whether authentication is working in order to safe guard important information

1.2.2 Black box testing

Black box testing is also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

Black box testing Real Time Bus Tracking system involves:

1. Usability Testing: This involves testing the system's usability, such as the ease of use, user interface, and user experience, to ensure that it is user-friendly and accessible.

2. Compatibility Testing: This involves testing the system's compatibility with different devices, platforms, and software versions, to ensure that it works seamlessly across a range of environments.

3. Acceptance Testing: This involves testing the system's compliance with its acceptance criteria and ensuring that it meets the user's expectations and requirements.

4. Regression Testing: This involves testing the system's functionality after making changes or updates, to ensure that it has not introduced any new defects or issues.

1.3 Testing levels and Results

There are many different levels of software testing .We are going to be describing tests conducted on each level

1.3.1 Unit Testing

In this level of testing, individual units or components of the application are tested in isolation to ensure they function correctly. For the Real Time Bus Tracking system, examples of units to be tested include the GSM module and the GPS module. The results of unit testing helps in ensuring that individual components work as expected and can be integrated successfully.

1.3.2 Unit testing results

Table 6 unit testing

Function	Test case 1	Test case 2
GSM module	Success	Failed due to poor network for messaging from netone to econet
GPS module	Success	Success

1.3.3 Integration testing

This is a level of testing done after unit testing and it involves with testing how the individual components of the application work together to form a complete system. Test drivers and test stubs are used to assist in Integration Testing.in the case of the Real Time Bus Tracking system ,integration testing involves checking whether the GSM module is sending GPS coordinates to customers via SMS. Also are the GPS coordinates being sent to the server correct

1.3.4 Integration Testing Cases

Table 7 integration testing

Test case objective	Test case description	Expected outcome	Result
Check if GSM module is receiving GPS coordinates through microcontroller	Send the word location to the number registerd on the GSM module	A text message containing a url to point location on a local map is expected	Success

Check if the GPS module is working properly	Open the map integrated in the mobile app	A marker containeing date and time should be seen marking location and date time	Success

1.3.5 Validation Testing

Table 8 validation testing

Domain	**Expected Results**	**Actual Results**
Functional Testing	• The system modules should function as expected. • The system should be easily accessible and user friendly. • Error messages should be displayed on the system.	As expected
Integration Testing	All the integrated modules should work together flawlessly.	As expected
System Testing	All the components of the system should function properly.	As expected
Acceptance Testing	The system should meet user requirements and system objectives.	As expected

1.3.6 Functional testing

Table 9 functional testing

Test Case ID	Test Scenario	Test Steps	Expected results	Actual results	Pass/Fail
AT001	User Registration	1.open the mobile app 2.click on the register button 3.enter valid user details 4.click on the submit button	registration success	registration success	Pass
AT002	User Login	1.open the mobile app	The user is logged in to	User logged in	Pass

		2.click on the login button	the system and directed to the main page		
AT003	Bus tracking	1.select bus to track 2.click on the bus number to search it 3.verify that the bus location is shown on the map	The bus location is displayed on a map	Location is shown on the map	Pass
AT004	Tracking using text	Send the word location to bus 1	A URL pointing to the bus location on a map is sent to the user	Location is shown on the map	Pass

1.4 Non-functional testing

Non-Functional Testing is the type of testing done against the non-functional requirements. Most of the criteria are not consider in functional testing so it is used to check the readiness of a system. Non-functional requirements tend to be those that reflect the quality of the product, particularly in the context of the suitability perspective of its users.

They include:

- Performance testing: This type of testing ensures that the system can handle a large number of requests and provide real-time updates without any significant delays.
- Load testing: This test verifies that the system can handle a high volume of requests simultaneously without crashing or slowing down.
- Stress testing: This test ensures that the system can handle unexpected usage without crashing or suffering from performance degradation.
- Security testing: This type of testing verifies that the system is secure and that user data is protected from unauthorized access, hacking, or data breaches.
- Usability testing: This test ensures that the user interface is intuitive and easy to use and that users can quickly and easily access the information they need.
- Compatibility testing: This test verifies that the system works correctly on different platforms, devices, and operating systems.
- Scalability testing: This test ensures that the system can handle an increasing number of users and requests without suffering from performance degradation.
- Availability testing: This test ensures that the system is available and accessible to users at all times, without any significant downtime or outages.

CHAPTER 6: CONCLUSION AND RECOMENDATIONS

1.1 Introduction

In conclusion, the development of a real-time bus tracking system using an ESP8266, GPS, and GSM module is a significant achievement in the transportation industry. This system provides an efficient and reliable way of tracking buses and providing real-time updates to the users. By integrating a mobile app, customers can access the system easily and get accurate information on the location, route, and timing of the buses. The use of GPS and GSM technology ensures that the system is accurate and reliable, while the ESP8266 provides the necessary connectivity and communication between the system and the mobile app.

1.2 Future works

The real-time bus tracking system can be integrated with other public transit systems, such as trains to provide a more comprehensive and seamless transportation experience for users. Digital payment system can be integrated into the system to enable users to pay for their fare using the mobile app, making the payment process more convenient and secure.

Artificial intelligence can be used to improve the accuracy of the system's tracking and provide more personalized and relevant updates to the users. For example, AI can analyze the user's travel history and preferences to provide customized route suggestions and real-time updates.

The real-time bus tracking system can be integrated with other smart city infrastructure, such as traffic lights and road sensors, to provide a more connected and efficient transportation system.

1.3 Recommendations

Some of the recommendations include:

- User-Centered Design: The system should be designed with the user in mind, providing a user-friendly and intuitive interface that is easy to use and understand. The system should also be accessible to users with disabilities, such as visual impairments, by providing features like text-to-speech and high-contrast modes.
- Accuracy and Reliability: The system's accuracy and reliability are critical to its success. The system should be designed to provide real-time updates on the bus's location, route, and timing accurately and reliably. The system should also be able to handle unexpected situations, such as traffic congestion or road closures, and provide alternative routes and timings.
- Security and Privacy: The system should ensure that user data is protected and that sensitive information is not exposed to unauthorized users. The data should be encrypted and transmitted securely to prevent any data breaches or unauthorized access.
- Scalability and Performance: The system should be designed to handle a large number of users and requests simultaneously without losing performance or reliability. The system should also be scalable, allowing for future growth and expansion.

- Integration with Other Systems: The system should be designed to integrate with other transportation systems, such as trains and subways, and digital payment systems, to provide a more comprehensive and seamless transportation experience for users.
- Continuous Improvement: The system should be continuously improved based on user feedback, market trends, and advancements in technology. Regular updates and maintenance should be provided to ensure that the system is up-to-date and functioning optimally.

1.4 Challenges

Developing a real-time bus tracking system requires technical expertise in hardware, firmware, software, and mobile app development. Integration of these components can be challenging, and any errors or defects in the system can lead to inaccurate tracking and unreliable updates.

The development and implementation of such a system can be expensive. The cost of hardware specifically was a bit high coupled with the fact that the components are delicate and can be rendered useless by static. At one point I had to buy another GPS module.

GPS accuracy can be affected by several factors, such as atmospheric interference, multipath errors, and satellite positioning. The system must be designed to account for these factors to ensure accurate tracking and reliable updates.

APPENDIX 1:User manual

Introduction:

Thank you for using our real-time bus tracking system. This system allows you to track your bus in real-time using your mobile device.

Getting Started:

To use the system, you will need to download the mobile app from the App Store or Google Play Store. Once you have downloaded the app, register your account and log in to begin tracking your bus.

Tracking Your Bus:

To track your bus, navigate to the buses and routes page and select the bus number you want to track. Then, click on the "Track my Ride" button to open the real-time tracking page.

The real-time tracking page will show you the current location of the bus on a map, as well as information about the bus's route and estimated arrival time. You can also set up notifications to receive updates about the bus's location and estimated arrival time.

Sending a Location Message:

If you want to send a location message to the system, you can do so by clicking on the "Send Location Message" button on the real-time tracking page. This will allow you to send your location to the system, which will then send you a URL pointing to the bus's location on a map.

Reporting an Issue:

If you encounter an issue with the bus, such as a delay or mechanical problem, you can report it through the mobile app. Simply click on the "Report an Issue" button on the real-time tracking page and fill out the form with the details of the issue.

Conclusion:

We hope that this user manual has helped you understand how to use our real-time bus tracking system. If you have any further questions or concerns, please don't hesitate to contact our customer support team. Thank you for using our system and have a safe and comfortable ride!

APPENDIX:2 TECHNICAL PAPER

Real Time Bus Tracking System

Tinotenda Katsamba; Simbai Zindove

Department of Software Engineering, School of Information Sciences and Technology

Harare Institute of Technology, Harare, Zimbabwe

tinotendakatsamba2000@gmail.com;

szindove@hit.ac.zw

ABSTRACT

In the busy cities like Harare and Bulawayo , people don't have time to invest in waiting for transport. Waiting time for transport in such crowded cities leads to less productivity on a whole. People face this problem in their daily life where they have no idea about the current status of their transport. So the proposed solution is an android based application that will help the user to check out the current location of the bus and also will help the user to know how much time the bus will take to reach the current location of the user. The system will use GPS as the basis for the application and basic android application will be interfacing with the updated database to provide the real-time data to the user, hence enhancing the user-experience. Text messages will be implemented alongside the mobile app to enable offline tracking

***Keywords*:** GPS,GSM

I. INTRODUCTIONS

Real-time bus tracking systems have become increasingly, popular in recent years as a means of improving public transportation services. These systems use GPS and other sensors to track the movement and location of buses in real time, providing passengers with accurate information about bus arrival times, delays, and other important information. This technology has the potential to significantly improve the efficiency and quality of public transportation services, as well as enhance the passenger experience. However, the implementation of real-time bus tracking systems is not without its challenges, including technical, operational, and regulatory issues. This survey paper is going to provide an overview of the current state of the art in real-time bus tracking systems, including the various technologies, approaches, and challenges associated with implementing such systems.

II. PROBLEM STATEMENT

Due to the inconsistent and unavailable of bus time tables in bus terminus, commuters have been on many occasions been left stranded.Despite government efforts to ease transport problems by offering subsidized ZUPCO buses on different routes to ease

transport costs and provide safe and cheap transport to the commuting public, commuters are still facing a lot of problems in accessing this transport service and in some cases turn to private commuters who overcharge, drive recklessly hence endangering the lives of civilians and in some cases they turn to be criminals who rob ,rape unsuspecting commuters who are desparate to get home especially after a long day work.In most cases commuters spend hours waiting for buses without the slightest hint on when they may come .This in itself is a wasting of time because rather than just checking on your phone to see whether there is a bus nearby to take the commuter home and do something else productive,one would be stuck in a queue adding to stress and other non communicable tensions to an individual.

.

III. RELATED WORKS

" Authors “Süleyman Eken, Ahmet Sayar” have implemented“ have implemented the system “A smart BusTracking System based on location- aware service and QR code.” In this paper, Bus tracking system, any passenger with Smartphone can scan QR code placed at bus stop to view estimated bus arrival times, current location of the bus. The drawback in this project was that the user had to be physically present at the bus stop to scan the QR code.

Author Shubham Jain et al., created an "Application-based bus tracking system",.This paper is based on a bus tracking system, in which a GPS Tracking application is used to track the bus. GPS technology is user-oriented, to receive the navigating instructions at any instant of time. Here, the location of the bus is received from the satellite and then with the help of cellular networks, it is further processed and sent to the web-server. The coordinates received are processed through Google Maps API. Google Maps API helps to collect data like latitudes and longitudes, locations, etc.The data received is processed in the user’s device, to display the real-time information.

Authors “M. A. Hannan, A. M.Mustapha,A.Hussain and H.Basri” have implemented the system “Intelligent BusMonitoring and Management System” The proposed system uses Artificial intelligence with the help of RFID module which is used in-order to reduce the manual work carried out in the Bus-Management & Monitoring System. In this a RFID is used to track a bus when it crosses the bus stop. Hence the exact location of the bus is not shown , only an approximate location is shown based on the bus stops. In today’s world,accuracy is very important and hence this was the limitation of this project.

Authors ``ManiniKumbhar, MeghanaSurvase, Pratibha MAvdhutSalunk” have implemented “Real Time Web Based Bus Tracking System” The proposed system reduces the waiting time of remote users for buses. A system is used to track the bus at any location at any time. All the current information is stored to the server and it is retrieved to remote users via web based application. This System is a web based system but nowadays people mostly tend to use Android apps since they are more portable and smartphones are used more

widely in today's world. Also a web based system is inconvenient for a user to use on a regular basis while waiting for a bus at the bus stop.

Authors "R.Maruthi, C.Jayakumari" implemented the system "SMS based Bus Tracking System using Open Source Technologies." A bus tracker application to track a bus using GPS transceiver has been proposed in this paper. The objective of this work is to develop a system that manages and controls the transport using a tracking device to know the scheduled vehicle and the current location of the vehicle via SMS using a GPS tracking device

Authors "Md. Marufi Rahman, Jannatul Robaiat Mou,Kusum Tara, Md. Ismail Sarkar" have implemented the system "Real Time Google Map and Arduino Based VehicleTracking System" using GSM and Arduino coordinates sent by arduino is shown on google maps

IV. SOLUTIONS

The problem can be solved by a Real Bus Tracking System

Objectives:

- To design a mobile app
- To design a bus tracking device (hardware)
- To send GPS coordinates to customers via text messages
- To link the mobile app with the tracking device

A. *Flow chart*

Figure 1: Architecture Solution

B. *Coding Strategy*

The coding strategy is a set of actions used to complete all of the project's goals. Because of the project's large size, it was separated into many parts. Before the database was constructed, a detailed design of how it would be structured was drawn. Before the classes were established, the structure and connections between them were determined. Some of the features were created by trial and error until the intended outcomes were achieved.

B. *Experimentation and Testing*

geometric matching module

- Unit testing: This involves testing individual components like the GPS,

GSM and esp8266 microcontroller on their own

- Integration testing: this involves testing whether the GPS, GSM and microcontroller are compatible with each other
- Code coverage testing: This involves measuring the percentage of code that is executed during testing to ensure that all code paths are covered and that there are no unreachable code segments.
- Security testing: This involves testing whether authentication is working in order to safe guard important information

V. CONCLUSION

In conclusion, the development of a real-time bus tracking system using an ESP8266, GPS, and GSM module is a significant achievement in the transportation industry. This system provides an efficient and reliable way of tracking buses and providing real-time updates to the users. By integrating a mobile app, customers can access the system easily and get accurate information on the location, route, and timing of the buses. The use of GPS and GSM technology ensures that the system is accurate and reliable, while the ESP8266 provides the necessary connectivity and communication between the system and the mobile app.

.

VI. FUTURE WORKS

The real-time bus tracking system can be integrated with other public transit systems, such as trains to provide a more comprehensive and seamless transportation experience for users. Digital payment system can be integrated into the system to enable users to pay for their fare using the mobile app, making the payment process more convenient and secure.

Artificial intelligence can be used to improve the accuracy of the system's tracking and provide more personalized and relevant updates to the users. For example, AI can analyze the user's travel history and preferences to provide customized route suggestions and real-time updates.

The real-time bus tracking system can be integrated with other smart city infrastructure, such as traffic lights and road sensors, to provide a more connected and efficient transportation system.

.

VII. ACKNOWLEDGEMENTS

First and foremost, I want to express my gratitude to the Almighty God for His constant direction in my life and for providing me with the skills and knowledge that I employed in the development of this system. I would also want to express my gratitude to my family for their help and support during the process. Thank you Christopher Meki, Kudakwashe Koti and everyone who helped me finish the project by providing me with the essential information and direction. Finally, I'd like to express my gratitude to my project supervisor, Miss Simbai Zindove, for her direction and consistent monitoring, which helped me complete this project.

REFERENCES

1. Eddie Chi-Wah Lau, ‘‘Simple Bus Tracking System’’, Journal of
2. f Khondker Shajadul Hasan, Mashiur Rahman, Abul L. Haque, M Abdur
3. Rahman, Tanzil Rahman, and M Mahbubur Rasheed, “Cost Effective
4. GPS-GPRS Based Object Tracking System,” Proceedings of the
5. Kudakwashe koti