VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

A Project Report on

"SMART BUS TRACKING SYSTEM"

Submitted in the partial fulfillment for the requirements for the conferment of Degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE AND ENGINEERING

By

Miss. M MRUDULA	**USN: 1BY14IS027**
Miss. SANGEETA JAIPRAKASH HEGDE	**USN: 1BY14IS045**
Miss. GEETHA J	**USN: 1BY15IS402**
Miss. LAKSHMI	**USN: 1BY15IS406**

Under the guidance of

Mrs. S. MAHALAKSHMI
Assistant Professor

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
YELAHANKA, BENGALURU-560064
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

2017-2018

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI, KARNATAKA

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
YELAHANKA, BENGALURU-560064

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **"SMART BUS TRACKING SYSTEM"** is a bonafide work carried out by **Miss. M MRUDULA (1BY14IS027), Miss. SANGEETA JAIPRAKASH HEGDE (1BY14IS045), Miss. GEETHA J (1BY15IS402), Miss. LAKSHMI (1BY15IS406),** in partial fulfillment for the award of **Bachelor of Engineering Degree in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2017-18. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The project report has been approved as it satisfies the academic requirements with respect to project work for the B.E Degree.

Signature of the Guide Mrs. S. Mahalakshmi	**Signature of the Coordinator** Dr. Pushpa S. K
Signature of the HOD Dr. Manjunath T. N	**Signature of the Principal** Dr. Mohan Babu G. N

EXTERNAL EXAMINERS

Name of the Examiners	Signature with Date
1.	
2.	

ACKNOWLEDGEMENT

We are happy to present this project after completing it successfully. This project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and everyone who has helped us make this project a success.

We heartily thank our **Principal, Dr. Mohan Babu G. N, BMS Institute of Technology & Management** for his constant encouragement and inspiration in taking up this project.

We heartily thank our **Head of Department Dr. Manjunath T. N, Dept. of Information Science and Engineering, BMS Institute of Technology& Management** for his constant encouragement and inspiration in taking up this project.

We heartily thank our Project coordinator **Dr. Pushpa S. K, Associate Professor, Dept. of Information science and Engineering,** for her constant follow up and advice throughout the course of the Project work.

We gracefully thank our Project guide**, Mrs. S. Mahalakshmi, Assistant Professor, Dept. of Information Science and Engineering,** for her encouragement and advice throughout the course of the Project work.

Special thanks to all the staff members of Information Science Department for their help and kind co-operation.

Lastly we thank our parents and friends for their encouragement and support given to us in order to finish this precious work.

By,
Mrudula M
Sangeeta Jaiprakash Hegde
Geetha J
Lakshmi

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

YELAHANKA, BANGALORE-64

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

Declaration

We, hereby declare that the project titled "**SMART BUS TRACKING SYSTEM**" is a record of original project work undertaken for the award of the degree Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2017- 18. We have completed this project under the guidance of **Mrs. S. Mahalakshmi, Assistant Professor, Dept. of ISE**.

I also declare that this project report has not been submitted for the award of any degree, diploma, associate ship, fellowship or other title anywhere else.

Student Photo	**Photo1**	**Photo2**	**Photo3**	**Photo4**
				
USN	1BY14IS027	1BY14IS045	1BY15IS402	1BY15IS406
Name	Mrudula M	Sangeeta Hegde	Geetha J	Lakshmi
Signature				

ABSTRACT

Bangalore city faces severe problems of road congestion and associated issues of commuters, which include delays in the arrival of buses at bus stops, lack of information about different bus routes and stops and time. College students/staffs will miss their bus by a fraction of second. This will lead to many problems like being late for the classes and sometimes late for the exams also. To overcome this we have implemented a Smart Bus Tracking System. This system is used to track the BMSIT college buses.

The proposed system uses a Smartphone application. Buses carry Global Positioning System (GPS) devices to track their positions and Google Maps API is used to display the vehicle on the map in the Smartphone application. It shows where exactly the bus is there on the map and provides updated information to the user at different time intervals. This also displays the estimated arrival time, which helps the user to know when exactly the bus is going to reach his/her stop.

Apart from this our system also provides the congestion details for driver. It provides the alternate routes for the driver if there is any congestion in his route. The next major advantage is, if the bus is in any emergency situation it sends an alert message to the transport in charge. The user can get flexibility of planning travel using the app, to decide when to catch the bus. The proposed system is user friendly and ensures safety and surveillance at low maintenance cost.

INDEX

Chapter No.	Chapter Title	Page No.
	ACKNOWLEDGEMENT……	**i**
	DECLARATION……...	**ii**
	ABSTRACT……...	**iii**
	LIST OF FIGURES……..	**iv**
	LIST OF TABLES……	**v**
1	**Introduction**	**[1-8]**
1.1	Background	1
1.2	Android	3
1.3	GPS Tracking unit	5
1.4	Motivation	8
2	**Literature Survey**	**[9-20]**
2.1	The smart bus for a smart city-A real time implementation	9
2.2	GPS supported city bus tracking and smart ticketing system	10
2.3	Smart assistance for public transport system	11
2.4	Smart bus: a tracking system for school buses	12
2.5	IOT based school bus tracking and arrival time prediction	13
2.6	Pune bus guide	14
2.7	M- Indicator-Mumbai	14
2.8	Delhi bus navigator	15
2.9	Bangalore BMTC information	16
2.10	Chennai bus route	16
2.11	Existing system	17
2.12	Problem statement	18
2.13	Proposed system	19
2.14	Objective	20
3	**Software requirement specification**	**[21-24]**
3.1	Functional and non-functional requirements	21
3.2	User requirements	22
3.3	System requirements	22
3.4	Software requirements	23
3.5	Use case diagram	23
4	**Design and analysis**	**[25-35]**
4.1	System architecture	25
4.2	Activity diagram	32

4.3	Dataflow diagram	33
4.4	Algorithm	34
4.5	Class diagram	35
5	**Implementation**	**[36-45]**
5.1	Platform selection	36
5.2	Programming language gist	37
5.3	Module description	37
5.4	Coding Standards	45
6	**Software Testing**	**[46-49]**
6.1	Unit Testing	46
6.2	Integration Testing	47
6.3	System Testing	48
6.4	Functional Testing	48
7	**Results and Discussions**	**[50-55]**
8	**Conclusion and future enhancement**	**[56-57]**
8.1	Conclusions	56
8.2	Limitations	56
8.3	Future enhancement	57
	BIBLIOGRAPHY	
A	**APPENDIX** **PLAGIARISM REPORT**	

LIST OF FIGURES

Figure No.	Figure name	Page No.
1.1	Bus Tracking System Architecture …….....	2
1.2	Android Architecture …….....	4
1.3	Architecture of GPS …….....	7
3.1	Use Case Diagram …….....	23
4.1	Design and Architecture ……..	25
4.2	Activity diagram for bus tracking …….....	32
4.3	Dataflow diagram …….....	33
4.4	Class diagram …….....	35

LIST OF TABLES

TableNo.	Table Name	Page No.
6.1	Test case for Unit Testing …….....	47
6.2	Test case for Integration Testing …….	48

CHAPTER 1

INTRODUCTION

1.1. BACKGROUND

Vehicle tracking systems were first implemented for the shipping industry because people wanted to know where each vehicle was at any given time. These days, however, with technology growing at a fast pace, automated vehicle tracking system is being used in a variety of ways to track and display vehicle locations in real-time.

However, bus transportation service has very poor transportation information system nowadays. Bus users do not know the exact arrival time for a bus, but only know the scheduled approximate arrival time. Bus transportation service does not have a proper system to track all buses position and the actual arrival time in every bus stop. These problems occur because current bus service system did not apply real time tracking technology to track on each buses on the road and also lack of a platform to update latest bus traffic information to bus users.

In order to solve these problems and enhance current bus service system, real time bus tracking system has to develop and implement. With real time bus tracking system, bus position data is connected real time and transmitted to a central server for processing and extracting transit information. The main technology used to develop this system is Global Positioning System (GPS). GPS technology able to receives the position of an object from space-based satellite navigation system through a GPS receiver.

For wireless data transmission, GSM and SMS technology are commonly used. The SMS technology through GSM network and GSM modem provide a user with vehicle location information. Instead of using SMS, the bus tracking system uses the smart phone application to track and monitor a bus location obtained from the in-vehicle tracking device.

The bus location is automatically placed on Google maps, which makes it easier for tracking a vehicle and provides users with more accurate vehicle location information.

The developed bus tracking system will be able to provide bus users a real time platform to check on updated bus traffic information, for example bus arrival or departure time. Besides, this system also reduces workload for bus management team and provides an immediate platform to update latest and accurate bus traffic information to bus users.

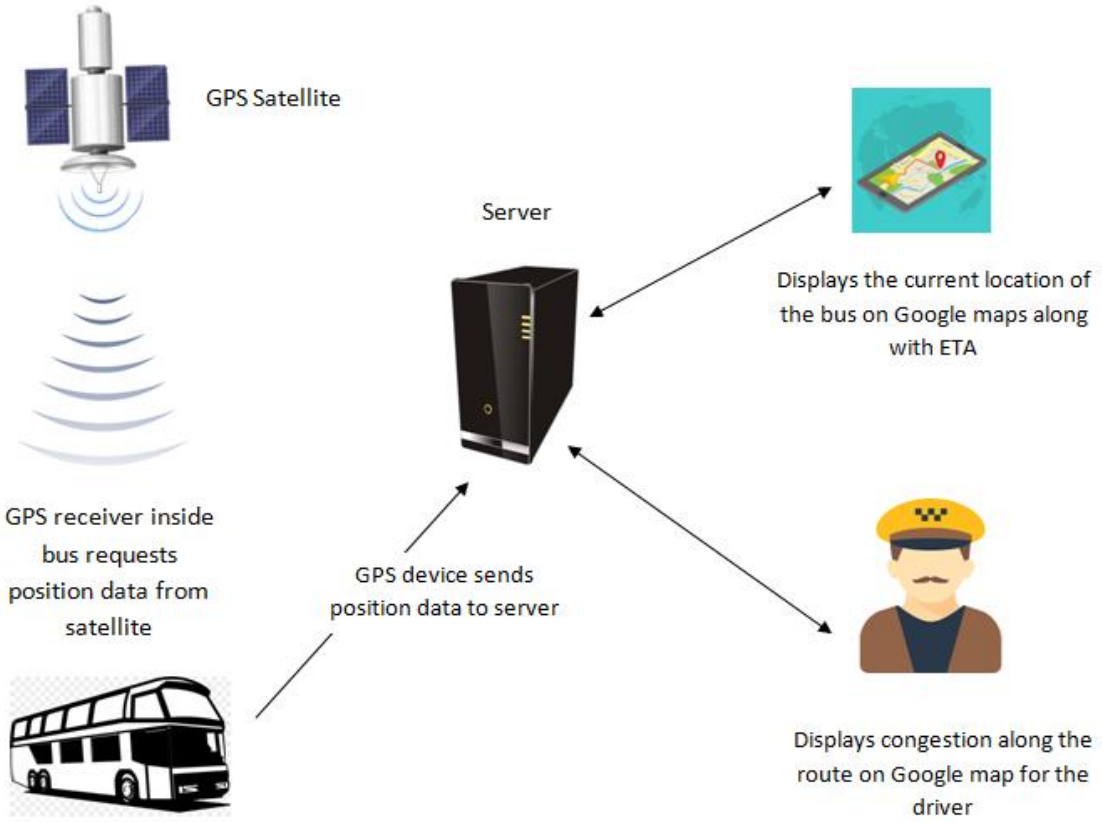


Figure 1.1. Bus Tracking System Architecture

1.2. ANDROID

1.2.1. Overview

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics. Now we have phones which can even access GPS, GPRS, Wi-Fi, NFC and lot of other cool and advanced features which you cannot even imagine. So in this Mobile world of this complication, android is one of those operating system platforms which made it easy for manufacturers to design top class phones.

Stored and copied to a stream where the actual data is transferred. Due to its native characteristic, it runs in the background and waits for a possible transfer operation. They are called right after the related button is pressed. Services retrieve values from static class called FinalValues.java through intent's extended data. They mainly include Final Values .EXTRAS_GROUP_OWNER_ADDRESS and Android OS.

As it is widely known, android is a Linux-based operating system led by Google. It is mostly developed for mobile devices to bring simplicity, functionality and efficiency to the market. Android is an open source project and it has a large number of developers writing applications. Developers write applications primarily in Java (Stephen Shankland, 2007) and applications can be downloaded mostly through official online store called Google Play. Currently there is 600,000 applications available on Google Play and so far 20 billion applications downloaded from this store (engaged, 2012).

Android runs on Linux with libraries and libraries written in C. Dan Morrill, Android Engineer in Google, explained that Android is not a specification, or a distribution in the traditional Linux sense. It's not a collection of replaceable components. Android is a chunk of software that you port to a device. (Dan Morrill, 2010)

Android uses the Dalvik Virtual Machine to run Dalvik Executable code translated from Java byte code. All standard APIs are defined in terms of classes, interfaces, methods and objects. In terms of hardware platform, ARM architecture is main platform for Android. However, there is also support for x86 architecture.

1.2.2. Architecture

Architecture Android runs on Linux under Dalvik VM. Dalvik has a just-in-time compiler where the byte code stored in memory is compiled to a machine code. Byte code can be defined as ‘intermediate level'. JIT compiler reads the byte code in many sections and compiles dynamically in order to run the program faster. Java performs checks on dependent portions of the code and thus the code is compiled only before it is executed. When it is compiled once, it is cached and set to be ready for later uses.

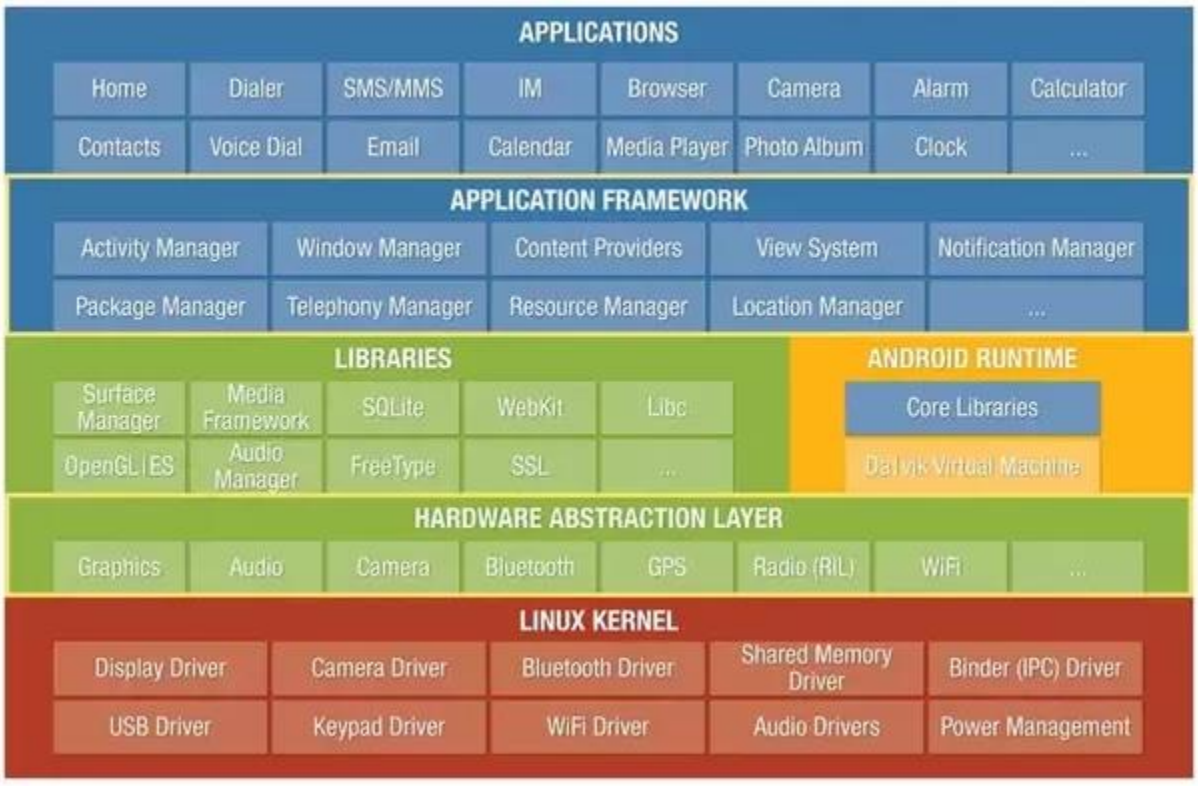


Figure 1.2. Android Architecture

1.2.3. Android Studio and Android SDK

Studio is the official IDE for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- Build variants and multiple APK file generation
- Code templates to help you build common app features
- A rich layout editor with support for drag and drop theme editing
- Lint tools to catch performance, usability, version compatibility, and other problems
- Code shrinking with ProGuard and resource shrinking with Gradle
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine Android provides a custom plug-in for Android development called Android Development Tool (ADT). It is designed to build Android applications. It lets the developer to establish new Android projects, build and debug applications, and export APKs.

1.3. GPS Tracking unit

A GPS tracking unit is a device, normally carried by a moving vehicle or person, that uses the Global Positioning System to determine and track its precise location, and hence that of its carrier, at intervals. The recorded location data can be stored within the tracking unit, or it may be transmitted to a central location data base, or Internet-connected computer, using a cellular (GPRS or SMS), radio, or satellite modem embedded in the unit. This allows the asset's location to be displayed against a map backdrop either in real time or when analyzing the track later, using GPS tracking software. Data tracking software is available for smartphones with GPS capability.

1.3.1. GPS Architecture

A GPS tracker essentially contains a GPS module to receive the GPS signal and calculate the coordinates. For data loggers it contains large memory to store the coordinates,

data pushers additionally contains the GSM/GPRS modem to transmit this information to a central computer either via SMS or via GPRS in form of IP packets.

- **Fundamentals:** The GPS concept is based on time and the known position of specialized satellites. The satellites carry very stable atomic clocks that are synchronized to each other and to ground clocks. Any drift from true time maintained on the ground is corrected daily. Likewise, the satellite locations are known with great precision. GPS receivers have clocks as well; however, they are not synchronized with true time, and are less stable. GPS satellites continuously transmit their current time and position. A GPS receiver monitors multiple satellites and solves equations to determine the precise position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver for it to compute four unknown quantities (three position coordinates and clock deviation from satellite time).

- **Mobile Phones with GPS Capability:** Due in part to regulations encouraging mobile phone tracking, including E911, the majority of GPS receivers are built into mobile telephones, with varying degrees of coverage and user accessibility. Commercial navigation software is available for most 21st- century smartphones as well as some Java-enabled phones that allow them to use an internal or external GPS receiver (in the latter case, connecting via serial or Bluetooth). Some phones using assisted GPS (A-GPS) function poorly when out of range of their carrier's cell towers. Others can navigate worldwide with satellite GPS signals as well as a dedicated portable GPS receiver does, upgrading their operation to A-GPS mode when in range. Still others have a hybrid positioning system that can use other signals when GPS signals are inadequate.

- **Mobile Messaging:** Mobile messaging plays an essential role in LBS. Messaging, especially SMS, has been used in combination with various LBS applications, such as location-based mobile advertising. SMS is still the main technology carrying mobile advertising / marketing campaigns to mobile phones. A classic example of LBS applications using SMS is the delivery of mobile coupons or discounts to mobile

subscribers who are near to advertising restaurants, cafes, movie theatres. The Singaporean mobile operator Mobile One carried out such an initiative in 2007 that involved many local marketers, what was reported to be a huge success in terms of subscriber acceptance. Companies offering location-based messaging (sometimes referred to as "geo-messaging") include The Coupons App (US), Central (International), Zhiing (international), BluePont (US), Dodgeball (US) and Beamster (Austria).

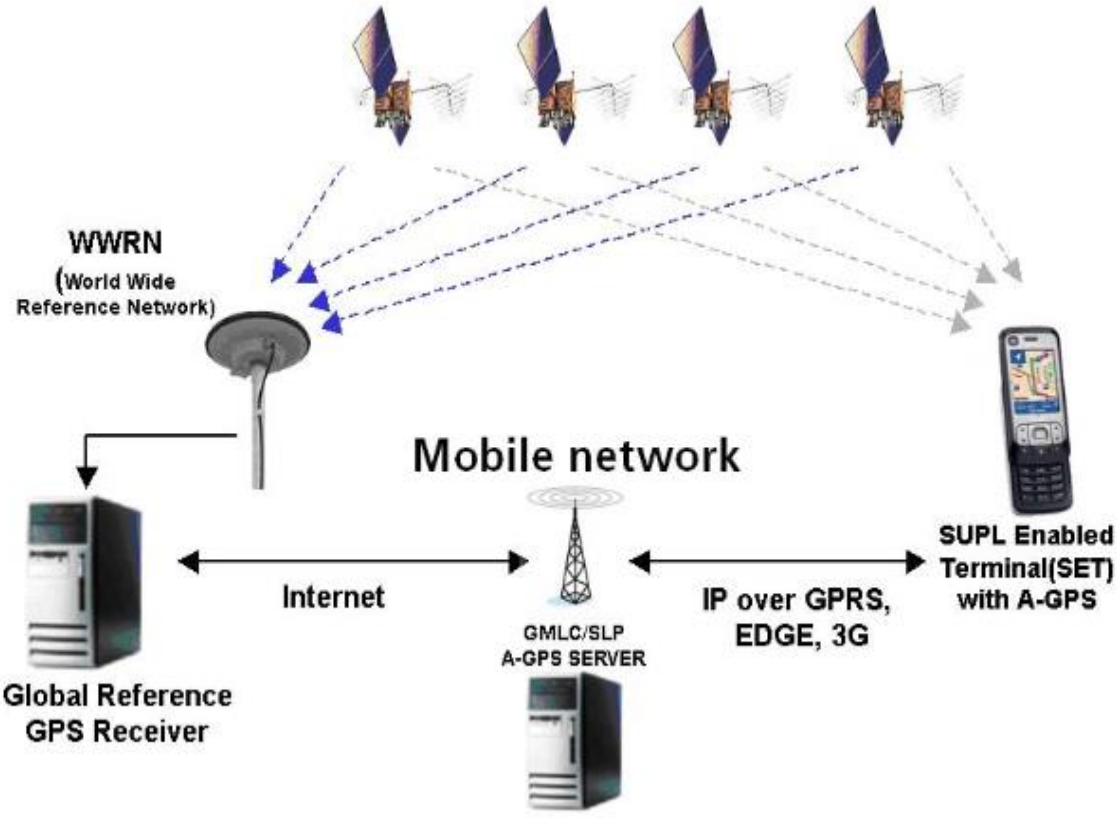


Figure 1.3. Architecture of GPS

1.4. MOTIVATION

Many cities have found that GPS tracking system not only improve the efficiency of city bus operation, but also encourage commuters to take the advantage of city bus system. Many city bus systems have discovered that GPS tracking system allows monitoring the location and arrival time of their bus actually increase the number of people using city buses for routine communing. The application is a user friendly one that anyone can access for free of cost. The basic idea for this project was to guide the bus travelers with the routes, all the possible stops that come on their way to the destination and moreover, display maps and track their locations and show the estimate remaining time required to reach. The aim is to overcome all the drawbacks faced in all the previous applications and generate fast and accurate results.

CHAPTER 2

LITERATURE SURVEY

2.1. Shared S, Bagavathi Sivakumar P, Anantha Narayanan V, " The Smart Bus for a Smart City - A real-time implementation", 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS).

The need for a real-time public transport information system is growing steadily. People want to plan their city commutes and do not like waiting for long hours, nor take a long route to reach their destination. The proposed hardware solution in this paper computes the shortest path to reach the destination in real time and gives that information to the bus driver. Artificial Neural Networks (ANN) is used to give an accurate estimate of the arrival time (ETA) to the commuter by means of an application. ETA to the next stop is communicated to the commuter using the MQTT (Message Queuing Telemetry Transport) protocol, by the hardware mounted on the bus. The proposed solution also adds a fleet management console to the administrators, making them manage and monitor the fleet of buses in real time. The prototype thus developed makes sure the commuting in cities is pleasant, and hassle free.

The existing platforms and applications that are used to assist commuters plan their travel uses mobile data for the connectivity and communication and GPS to get the real-time location of the bus (or other means of transport) relative to the commuter. There are solutions that offer a limited accuracy in metropolitan cities. However, these solutions are not available to the other cities and also, they rely on historical data to provide information.

The Intelligent Transport System (ITS)s based solutions can be studied to overcome these pitfall, would help the commuter to effectively utilize the public transport which includes lower waiting time.

There are many implementations of Intelligent Transport System all around the world, each solution designed to address a specific demographic region. There are existing solutions like tram TRACKER by Yarra Technologies in Melbourne, Australia and Google Maps is always there to cater the needs of the metropolitan commuters. The components of ITS Technologies are the wireless communication like Wi-Fi, WiMAX, RFID, etc. and computational technologies like AI, Real time data processing, etc.

2.2. Ajay Shingare, Ankita Pendole, Nikita Chaudhari and Parikshit Deshpande, Prof. Samadhan Sonavane," GPS Supported City Bus Tracking & Smart Ticketing System",2015 International Conference on Green Computing and Internet of Things (ICGCIoT).

Now-a-days increasing density of vehicles on road is becoming the problem for the traffic control. Ultimately arising obstacle in the managing and tracking of the vehicle. Because of the problem state, it is necessary for every organizations and individuals to track the vehicle. People will monitor and track their vehicles for the safety concerns with the help of our Android app. Public transport and private buses tracked to citizens with traffic and transportation details like location, crowd, etc. The proposed system will be used for the positioning of the bus from remote location. The Smart Card based ticketing module which swaps the card to the smart hand held device for the transaction purpose. The smart ticketing device will also contain the dynamic routes as per the bus depot. The smart device has enhanced with the GSM and GPS technology and made available with required necessary configurations which makes it very efficient than that of the existing system.

The location of the bus can be observed continuously using GPS system. The GPS satellites transmit signals to a GPS receiver. These receivers statically receive signals. GPS satellite transmits data that indicates the location and current time of the vehicle. The Smart Card provides, identification, application processing along with data storage. Every passenger will carry the smart card. The Smart Card holds information of the user such as available balance, identification number, user's information.

These smart cards are capable of recharge. By integrating both GPS technology and smart cards we are going to design a whole bus ticketing system. Whenever the passengers will enter in the bus he/she will be asked by the conductor whether he/she wants to buy ticket by using smart card or money. If smart card is used, then the conductor will swipe smart card. Then validity and of smart card will be checked with server and then the ticket will be issued. According to Source and destination the distance covered by passenger is get calculated and according to that bus fare amount will be reduced from smart card. The smart cards will also useful for conductor for fast issuing the tickets to the passengers.

2.3. ReshmaRathod, "Smart assistance for public transport system" 2016 International Conference on Inventive Computation Technologies (ICICT), Volume: 3.

In this paper we have provided public Smart Assistance in Public Transport System. The project is to be implemented for public bus (for ex: PMTs in Pune). It has the entire smart assistance system required for public security and safety. The smart system includes safety for women as well. It has accident detection and monitoring facility. It also has user friendly application for user to track bus on smart their phones. The smart system can be designed for both online (GPS) and offline (GSM) for user friendly service. Here, GPS system is used to get real-time co-ordinates for offline (GSM) system. It allows user to save its time by acknowledging no. of persons present in the bus as well as no. of seats available in the bus along with the current and next stop acknowledgement with its arrival timings. It also has ramp for handicap people to provide them ease to use the bus service. It also has driver authentication system using RFID tag.

The system also has many additions feature to make public transport system an intelligent and easy to use system, so that public can take smart advantage of it. The system is specially designed for Smart Cities as it is trending now-a-days. The basic idea of producing smart public transport system is to be developed on ARM system using GPS/GSM technology. The basic need to develop this system is to minimize public time issues related to public transport system. Features to be implemented in the system are vehicle tracking

(online/offline), availability of no. of seats in the vehicle (bus), engine heat monitoring in the bus, women safety, accident detection, and various other features.

A Smart Assistance for Public Transport System is to be designed. The Public transport selected is Public Bus. The issues related with public transport bus are taken into consideration. The issues such as bus arrival time prediction, no. of persons available in the bus, safety for women's, accident detection and safety, alcohol detection for driver, speed limiter and indicator, ramp for handicap people, driver authentication using RFID tag and bus report to public through online/offline options are available. In this paper we are focusing on offline system. The basic methodology used is GPS/GSM.

2.4. Majd Ghareeb, Athar Ghamlous, Hawraa Hamdan, Ali Bazzi,Samih Abdul-Nabi, "Smart bus: a tracking system for school buses", 2017 Sensors Networks Smart And Emerging Technologies (SENSET).

An increased concern for parents is the safety of their children on the way back home from school and the timing of their arrival. Waiting school buses in the morning and then in the afternoon to return kids back is a time wasting daily mission on parents, especially with the increasing traffic jams at these hours. In this paper we present a mobile and web application that is designed to address this issue. The system will help parents, the school and the bus to communicate automatically and easily via the application in order to detect kids' arrival time. The bus application side will notify parents few minutes before its approaching to their home. Furthermore, the system will allow parents to inform the school and hence the bus application side about the absence of their kid. The system has been efficiently and dynamically designed and implemented so it can be hosted and used by any school administration without the need to any major modifications.

The objective of this project is to facilitate this task on parents and save their time by automatically notifying them few minutes before the arrival of their kid, so they can go out to receive him from the bus. This is also applied in the morning tour of the bus to inform parents that the bus is approaching and their kid should go out to take the bus to school. This system can be used by any Lebanese school to increase the safety measures for its students and to relieve parents from the responsibility of waiting school buses each day.

2.5. R.C.Jisha, Aiswarya Jyothindranath, L Sajitha Kumary, "IoT based school bus tracking and arrival time prediction", 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

While opting for the public-transport, time and patience are of more concern. We can also say, passengers travelling on public-transport found their loss of time due to waiting at the bus-stops. This system is providing a real-time vehicle tracking via Global-Positioning System (GPS) technology to detect the location of the bus and to use internet by a general-packet-radio service (GPRS) technology to display live images on the Google Map or website app for tracking location of buses anytime. We are using the GPS and GPRS modules, the GPS module will locate the buses via the satellite, and the GPRS module will collect all data and send it to the website.

The buses will be monitored live using coordinates with this system. Also by implementing geofence, user get notify once bus entered in his/her predefined area. We are developing an Android-application which will give the real-time schedule of buses. Also, it provides quick and real-time replay for inquiry, via the server. Also in case of any unexpected activities or breakdown, the alert will be sent to the system, with Bus location.

The transportation system provide as the heart in the social and economic-growth of the country. As the population in India is increasing, rapid explode in rate of vehicles which results in an overload on traffic management. Public transport is becoming an important part of transport system in urban areas, advance in easily available technology Can be enforced which help the passenger who recalculate between a rural and urban to-get the travelling information and it helps the passengers to comfort them with the final real time location. Public transport mainly the bus sluice has been properly developed in many parts of the world.

For reducing the fuel-usage, snobby usage of car and comfort traffic crowding we can use the bus services. Passengers require the exact schedule of bus. The anxiety of passengers increases while waiting for a longtime at the bus stop and changes their mind to opt for the

buses. Many passengers are usually on-time for office and many students restarted to their classes as they determine to stop for the buses instead-of taking another mode of transportation. Goal of system is to decrease the complexity and cost of content these services by implementing Easy-Tracker, an auto system for the transit-tracking and advent time prediction.

2.6. PUNE BUS GUIDE

- An application has been implemented in Pune, named "Pune Bus Guide".
- This application is a quick reference guide for latest updated PMPML Bus Routes. This will help people in Pune or visiting Pune while traveling by PMPML buses.
  This application allows users to
  - Search buses by route number and/or Start/Destination stop names.
  - View the route details showing list of stops in either direction and approximate distance (in km) & approximate duration (in minutes) to complete the route.
- This application gives the way to the destination correctly, but the number of drawbacks that it has is greater than the number of advantages.
- It does not show the passengers current location even if he/she is connected to the GPS. Also, this application has been proven useless as it does not display the bus numbers, so the passengers find it very hard to know the number and time of arrival of the respective buses.
- It does not have a real time bus tracking service or does not even generate maps for the users ease. This application has never been updated ever since its development. Moreover, this application has bugs which makes it all the more difficult for the user to use it.

2.7. M-INDICATOR – MUMBAI

- Another application that was implemented in Mumbai, named "M-Indicator – Mumbai

- Mumbai's travel portal has details about Harbour, Central and Western railway local trains. You can find the schedule of trains between any two stations with time table; maps, single journey return journey, and monthly season pass fare details.
- Route information and timetable of Mumbai Monorail and Metro train is also available on Go4Mumbai website.
- Go4Mumbai website also lists the options to travel in Mumbai using various cab services like Meru, Tabcab, EasyCabs, Uber, Ola or private cabs.
- In case if three or less than 3 people are traveling, options for Auto rickshaw is also provided. There details on Toll Plazas with Toll amount for private and public vehicles.
- Its latest updates have given issues on every Android mobile supporting even the most recent device version. The "A to B" module of buses has given problems.
- Whenever an option for the source to destination is selected, the field still remains blank, i.e. no bus routes are displayed.

2.8. DELHI BUS NAVIGATOR

- The application built in Delhi named "Delhi Bus Navigator".
- It is very useful for the people in metro cities as people are very much dependent on the local bus services to commute and trust them more than any other transport medium. The capability of the Transit to show the routes, transit times and comparison between two trip options makes the trips for the people all the more easier.
- It also helps us in find a trip which involves fewer transfers and lesser walking. With the availability of this service, there is a scope of improvement in providing passengers the accurate information for traveling within Delhi.
- Installation of the GPS (global positioning system- a satellite navigation aid) in the buses is also being planned, which will help the commuters plan their trips in a better way with real-time monitoring bus movements.

- It has drawbacks like: The application works smoothly when offline, but works very badly when connected to the Internet. The application gives information about direct routes only.
- It does not give information about the alternate routes. This application has bugs due to which it lags all the time. Most of the time the application crashes when requested for specific bus routes.

2.9. BANGALORE BMTC INFORMATION

- The application developed in Bangalore named “Bangalore BMTC Info”.
- Finds out the details of the BMTC Bus numbers. Selects the BMTC numbers and the route details (source, destination), route length and the number of trips information is shown.
- The Application Has The Source And Destination Added. Users Can Search Bus Numbers By Specifying Source To Destination.
- The Applications Has The Bus Route/Stops Displayed From Source To Destination. The Stops Are Available for a bus or for a route when a source/destination is selected.
- It has drawbacks like: The application is never in an updated condition.
- The application has fed in wrong routes on several buses and given no updates to fix them. After the minimization and restoration of the application, it cannot search anything.
- This application crashes almost always. The application is not user friendly with a complicated User Interface (UI).

2.10. CHENNAI BUS ROUTE

- The application developed in Chennai named “Chennai Bus Route”.

- The bus commuter can update the current location of the bus during their travel by just updating “spot” and the app will use the current location retrieved through the GPS of the user’s phone as the current location of the bus.

- The “MY MTC” app also allows users to tag their favorite buses and get regular and real-time updates about the location of that bus. The commuters can even check the bus location by using “find”. The app makes use of efficient algorithms to make this tedious process look easy and fast.

- The app comes packed with very interesting and impressive features. It provides the commuters with real-time notifications and information about the bus. It makes spotting and finding of a bus a real easy task.

- The app even lets a user find the nearest bus stop from their current location through their phone’s GPS. In order to make travelling by buses easier and worry free, the app provides the users with the route maps of the bus route. It even provides them with them information like distance, bus fare and number of stops about the bus.

- “Chennai Bus Route” has the following drawbacks: The application works fine, but the bus timings have not been mentioned. Not all bus stops are updated. The application does not display maps.

2.11. EXISTING SYSTEM

There are many existing platforms and applications that are used to assist commuters plan their travel uses mobile data for the connectivity and communication and GPS to get the real-time location of the bus (or other means of transport) relative to the commuter. There are solutions that offer a limited accuracy in metropolitan cities.

However, these solutions are not available to the other cities and also, they rely on historical data to provide information. The Intelligent Transport System (ITS)s based solutions can be studied to overcome these pitfall, would help the commuter to effectively utilize the public transport which includes lower waiting time. There are many implementations of Intelligent Transport System all around the world, each solution designed to address a specific demographic region. There are existing solutions like tram TRACKER by Yarra Technologies in Melbourne, Australia and Google Maps is always there to cater the needs of the metropolitan commuters.

The components of ITS Technologies are the wireless communication like WiFi, WiMAX, RFID, etc. and computational technologies like AI, Realtime data processing, etc. The solution is also loosely based on the Floating car data, which is based on the collection of position, speed and direction of travel of the vehicles from mobile phones with a two-way communication of GPS enabled vehicles with Smartphone based applications. The task of computing the shortest path of travel is computed with Dijkstras Algorithm the optimizations done are discussed in section III. Communication protocol used to develop this prototype uses 3G for internet connectivity with MQTT[5,6] for message passing from the bus to the commuters. MQTT enables the bus to interact with the commuter directly.

2.12. PROBLEM STATEMENT

1. Possessing own transportation has become more common nowadays. The number of vehicles on the road keep on increasing and most of us are eager to own personal vehicle as we can go anywhere without limitation.
2. Undoubtedly, the existence of bus has reduced road traffic and taking bus is a good starting to inculcate the car-pooling value. Besides, it provides a low-cost transportation which means to the low-income family for traveling to another destination. However, things always don't come perfect.
3. The main drawback of traveling with bus is the inconsistent arrival time which may due to unforeseen circumstances. Even when we know the bus schedule well, there are number of reasons that bus as may not arrive as expected.
    - Traffic congestions
    - Heavy downpour
    - Bus breakdowns

    It is particularly annoying when a person has urgent appointment, but we are late due to the time-consuming of bus trip.

4. Bangalore city faced severe problems of road congestion and associated issues of commuters, which include delays in the arrival of buses at bus stops, lack of information about different bus routes and stops and time. College students/staffs will miss their bus

by a fraction of second. This will lead to many problems like being late for the classes and sometimes late for the exams also.

5. Students also cannot check on the updated bus schedule if there is a bus delay happens. For example, student can choose walk to the building he or she want to go instead of waiting for a delay bus if there is a real time platform for student to know about the bus is delay.

2.13. PROPOSED SYSTEM

The real time tracking of bus can be done and this information is then given to remote user who wants to know the real time bus information. Our system provides the relevant information regarding

- ✓ Real time location:

  Here the current location of the bus will be displayed.
- ✓ Route details:

  The system will display the route details with stops name.
- ✓ Driver's Contact Number:

  The system will display the driver's name with contact details.
- ✓ Average waiting time and expected time to reach:

  The system will display the ETA(estimated time of arrival) on the Google maps between two stops.
- ✓ Real time traffic to diverse route in case of heavy congestion:

  The system will display the traffic details for the driver in case of congestion.
- ✓ Emergency module:

  The system will send the notifications to the management in case of emergency.

2.14. OBJECTIVES

The objective of bus tracking system is,

- The mobile phone industry is one of the fastest and most dynamic business sectors today. The need to communicate efficiently and instantaneously is always an undying necessity. The market sector and the ever-growing and demanding consumers always want to have more, and they want it better than ever. Having a mobile phone for us makes life easier.
- Communication is always a part of daily life, and we cannot avoid it. The invention of mobile phone has gone tremendous leaps in innovation and new applications. Originally, it was intended to be a telephone that can be carried wirelessly at greater distances.
- Advances in communications, upgrades in radio frequency and developments on the internet had given mobile phones more sophisticated but easy to use in applications.
- The main goal of the proposed work is to improve the Bus system by adding the necessary additional features into the application, like accurate bus timings, correct bus numbers and moreover adding a GPS tracker into it.
- This study accepts input in the form of selection of the bus, to track the location of the respective bus and give the map for the same. It can also show the expected time required for the bus to reach the user.
- To inform drivers/user regarding diversion of route in case of heavy traffic or congestion.
- Emergency situations can also be informed to the transport in charger.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1. FUNCTIONAL REQUIREMENTS

Functional requirements refer to the functionalities that must apply to a system. The functional requirements of bus tracking system are stated below.

- ✓ 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 bus positioning module, calculate the estimated time to user and display the position on maps.
- ✓ The system must be able to show estimated arrival time for every bus in every bus stop.
- ✓ The system must be able to allow user retrieve information from mobile device and computer.
- ✓ The system must be able to show the traffic information on maps.
- ✓ The system must be able to send the SMS when in emergency.

NON-FUNCTIONAL REQUIREMENTS

- ✓ The system should provide the accurate estimated bus arrival time to user.
- ✓ The system should reduce the paper work done by bus management team.
- ✓ The system should be able to increase the efficiency and performance of bus service.
- ✓ The system should reduce work done by bus management team by automated calculation of estimated bus arrival time and showing real time bus position to user.
- ✓ The system should allow user to access information in anywhere with anytime.

3.2. USER REQUIREMENTS

There are two main target users for the proposed system, bus user (student/staff) and bus management team. The requirement from different user is stated below.

3.2.1. Bus user

- Student/Staff is the main user of the proposed system because the main objective of bus tracking system is to provide estimated bus arrival time for student.
- The student/staff must able to retrieve real time estimated bus arrival time for every bus stop. While waiting in bus stop, student able to access bus tracking system with mobile device instead of using computer to access.
- This is the main purpose of bus tracking system in mobile application is developed. The system is provided real time bus tracking system with mapping feature, which mean student/staff able to view the bus position with a map.
- With this mapping technique, student able to know where is a bus position based on the map in real time.

3.2.2. Bus Driver

- Bus driver is the second important user of this system. Bus driver is able to update bus status accordingly in order to inform bus users about immediate situation.

3.3. SYSTEM REQUIREMENTS

- **System Processor:** Pentium P4
- **Mobile Processor:** 1GHz or higher
- **Motherboard:** Genuine Intel
- **Memory:** 512 MB of RAM, 1GB recommended.
- **Display:**1024x 768 or higher-resolution display with 16 bits colors of android mobile phone.

3.4. SOFTWARE REQUIREMENTS

- **Operating system:** Windows XP or higher.
- **Technology Used:** Android 4.1 or higher
- **IDE:** Android Studio 3.0.1
- **Plug-in:** ADT plug-in(Android Development Tool)
- **Language:** Android, JAVA, SOAP or Restful web service.

3.5. USECASE DIAGRAM

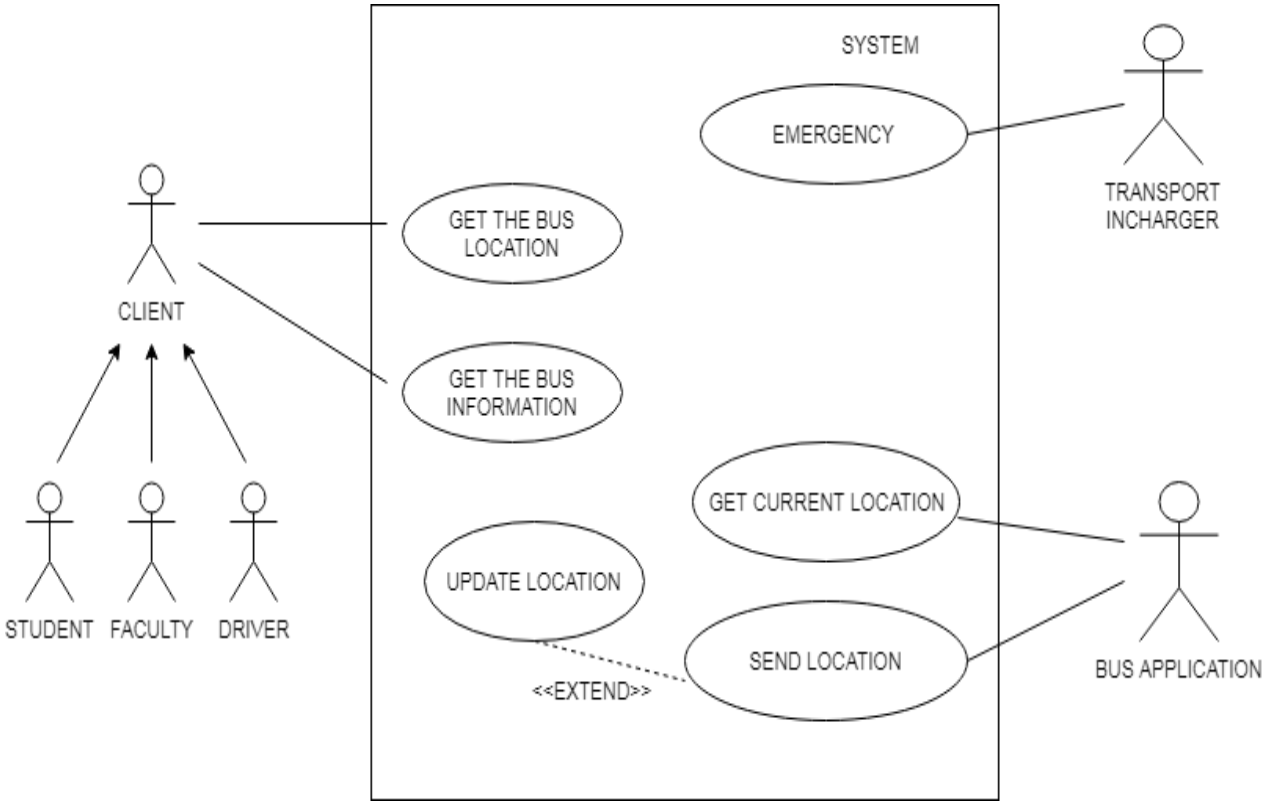


Figure 3.1. Use Case Diagram

Actor:

- Student
- Faculty
- Driver
- Transport in charge
- Bus application

Use-case:

- Get the bus location
- Get the bus information
- Get current location
- Send location
- Update location
- Emergency

Pre-condition:

- GPS in bus should be working properly and should send the coordinates to the server

Post-condition:

- Mobile application should receive location of bus from server

CHAPTER 4

DESIGN & ANALYSIS

4.1. SYSTEM ARCHITECTURE

This chapter describes features, fragments, classes, architecture and the application itself by providing necessary information of major components. First, overall information is given along with project's components and classes. Subsequently, the architecture details of the application are discussed.

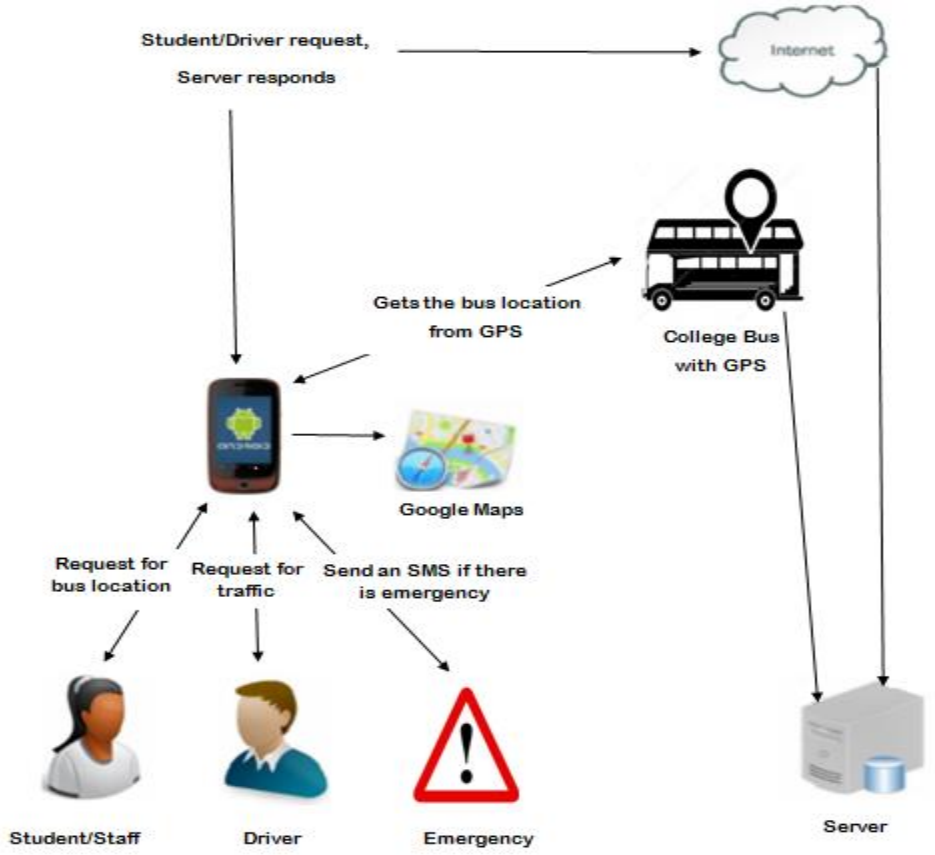


Figure 4.1. Design and Architecture

4.1.1. User interface design

The interface design of bus information must be as simple as possible so that bus users able to get information easily. The application is divided into three main parts, which is map with bus position, congestion details and emergency.

The first segment is the extraction of the information from the API, second is navigation using Google maps for traffic and third is getting the current location and sending SMS. Core operations of the entire system is carried out with these two classes, namely as MainActivity.java and Routes.java

The first module in the app is the commuter module. The commuter module contains two different buttons, one is for bus 1 and the other is for bus 2. Users can click on the respective buttons. By clicking so, they can get the latitude and longitude information of where exactly the bus is located in the form of a red marker along with the Estimated Arrival Time (ETA).User location is also highlighted in the app in the form of a blue circle .It also shows the name of the location. When the user clicks on the directions button, it shows the exact distance between the user and the bus.

The second module is the driver module. This module also has two buttons which correspond to the two routes. When the driver clicks on the respective routes, he can get the traffic details on his routes. Red color indicates high traffic; orange color indicates medium traffic whereas blue color indicates no traffic.

The last module is the Emergency module. If there is any emergency situation like breakdown of bus, accident and so on, the user can click on the emergency button. When he/she clicks on the button, an alert box will be displayed to confirm whether he/she needs the help or not. By clicking on the yes option, a SMS will be sent to the transport in charge of the college along with the exact location of the bus.

Apart from this, the app contains a navigation drawer which includes three different options: Route map, Know your driver and Feedback. Route map specifies each of the stops along the respective route. Two radio buttons have been incorporated for this purpose. If the user

wants to get to know any information regarding the driver, he can click on the Know Your Driver option.

4.1.2.XmlPullParser Factory

Android recommends usingXMLPullParser to parse the xml file than SAX and DOM because it is fast. The org.xmlpull.v1.XmlPullParser interface provides the functionality to parse the XML document using XMLPullParser.

xmlPullParserFactory = XmlPullParserFactory.newInstance();

newInstance(): Creates a new instance of a PullParserFactory that can be used to create XML pull parsers.

4.1.3. Background asynctask(async task)

Android AsyncTask is an abstract class provided by Android which gives us the liberty to perform heavy tasks in the background and keeps the UI thread light thus making the application more responsive.

Android application runs on a single thread when launched. Due to this single thread model tasks that take longer time to fetch the response can make the application non-responsive. To avoid this we use android AsyncTask to perform the heavy tasks in background on a dedicated thread and passing the results back to the UI thread. Hence use of AsyncTask in android application keeps the UI thread responsive at all times.

The basic methods used in an android AsyncTask class are defined below :

1. **doInBackground() :** This method contains the code which needs to be executed in background. In this method we can send results multiple times to the UI thread by publishProgress() method. To notify that the background processing has been completed we just need to use the return statements.

2. **onPreExecute() :** This method contains the code which is executed before the background processing starts
3. **onPostExecute() :** This method is called after doInBackground method completes processing. Result from doInBackground is passed to this method
4. **onProgressUpdate() :** This method receives progress updates from doInBackground method, which is published via publishProgress method, and this method can use this progress update to update the UI thread

The three generic types used in an android AsyncTask class are given below:

1. **Params :** The type of the parameters sent to the task upon execution
2. **Progress :** The type of the progress units published during the background computation
3. **Result :** The type of the result of the background computation.

4.1.4. Http URL connection

A URL Connection with support for HTTP-specific features. Uses of this class follow a pattern:

1. Obtain a new HttpURLConnection by calling **URL.openConnection()** and casting the result to HttpURLConnection.
2. Prepare the request. The primary property of a request is its URI. Request headers may also include metadata such as credentials, preferred content types, and session cookies.
3. Optionally upload a request body. Instances must be configured with setDoOutput(true) if they include a request body. Transmit data by writing to the stream returned by **getOutputStream().**
4. Read the response. Response headers typically include metadata such as the response body's content type and length, modified dates and session cookies. The response body may be read from the stream returned by **getInputStream().** If the response has no body, that method returns an empty stream.

5. Disconnect. Once the response body has been read, the HttpURLConnection should be closed by calling **disconnect()**. Disconnecting releases the resources held by a connection so they may be closed or reused.

4.1.5. Uri and "geo:0,0?q="

URI(Uniform resource identifier) as its name suggests is used to identify resource(whether it be a page of text, a video or sound clip, a still or animated image, or a program).The most common form of URI is the Web page address, which is a particular form or subset of URI called a Uniform Resource Locator (URL).

Android uses URI string as the basis for requesting data in a content provider (i.e. to retrieve a list of contacts) and for requesting actions (i.e. opening a webpage in a browser). Intents let you start an activity in another app by describing a simple action you'd like to perform (such as "display a map" or "show directions to the airport") in an Intent object. The Google Maps app for Android supports several different intents, allowing you to launch the Google Maps app and perform one of four actions:

- ✓ **Display a map at a specified location and zoom level:** Search for locations or places, and display them on a map. Request directions from one location to another. Directions can be returned for three modes of transportation: driving, walking, and bicycling.
- ✓ **Use of geo**: Intent to display a map at a specified location and zoom level. q defines the place(s) to highlight on the map. The q parameter is required for all Search requests. It accepts a location as either a place name or address.

4.1.6. LocationManager

This class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to fire an application-specified Intent when the device enters the proximity of a given geographical location.

- ✓ **LocationListener:** Used for receiving notifications from the LocationManager when the location has changed. These methods are called if the LocationListener has been registered with the location manager service using the requestLocationUpdates(String, long, float, LocationListener) method. The methods of LocationListener are:
    1. **onLocationChanged:**
       voidonLocationChanged (Location location)
       Called when the location has changed.
    2. **onProviderDisabled:**
       voidonProviderDisabled (String provider)
       Called when the provider is disabled by the user. If requestLocationUpdates is called on an already disabled provider, this method is called immediately.
    3. **onProviderEnabled:**
       voidonProviderEnabled (String provider)
       Called when the provider is enabled by the user.
    4. **onStatusChanged:**
       voidonStatusChanged (String provider, int status, Bundle extras)
       Called when the provider status changes. This method is called when a provider is unable to fetch a location or if the provider has recently become available after a period of unavailability.
- ✓ **getLatitude():** Get the latitude in degrees.
- ✓ **getLongitude():** Get the longitude in degrees.

4.1.7. Fine Location

Provides better and accurate locations.

<uses-permissionandroid:name="android.permission.ACCESS_FINE_LOCATION"/>

It gives permissions to work with GPS Provider and Network Provider.

4.1.8. Coarse Location

Provides less accurate locations.

<uses-permissionandroid:name="android.permission.ACCESS_COARSE_LOCATION"/

It gives permissions to work with only Network Provider.

4.1.9. Geocoder

A geocoder is either a piece of software or a service that implements a geocoding process. The Android API contains a Geocoder class that can use either a location name or a location's latitude and longitude values to get further details about an address (it can perform both forward and reverse geocoding).

Public constructors are:

- **Geocoder(Context context, Locale locale):**Constructs a Geocoder whose responses will be localized for the given Locale.
- **Geocoder(Context context):**Constructs a Geocoder whose responses will be localized for the default system Locale.

4.1.10. SmsManager

Manages SMS operations such as sending data, text, and pdu SMS messages. Get this object by calling the static method getDefault().SmsManager class is responsible for sending SMS from one emulator to another or device.SmsManager class is responsible for sending SMS from one emulator to another or device.

SEND_SMS: Allows an application to send SMS messages.

Constant Value: "android.permission.SEND_SMS".

4.2. ACTIVITY DIAGRAM

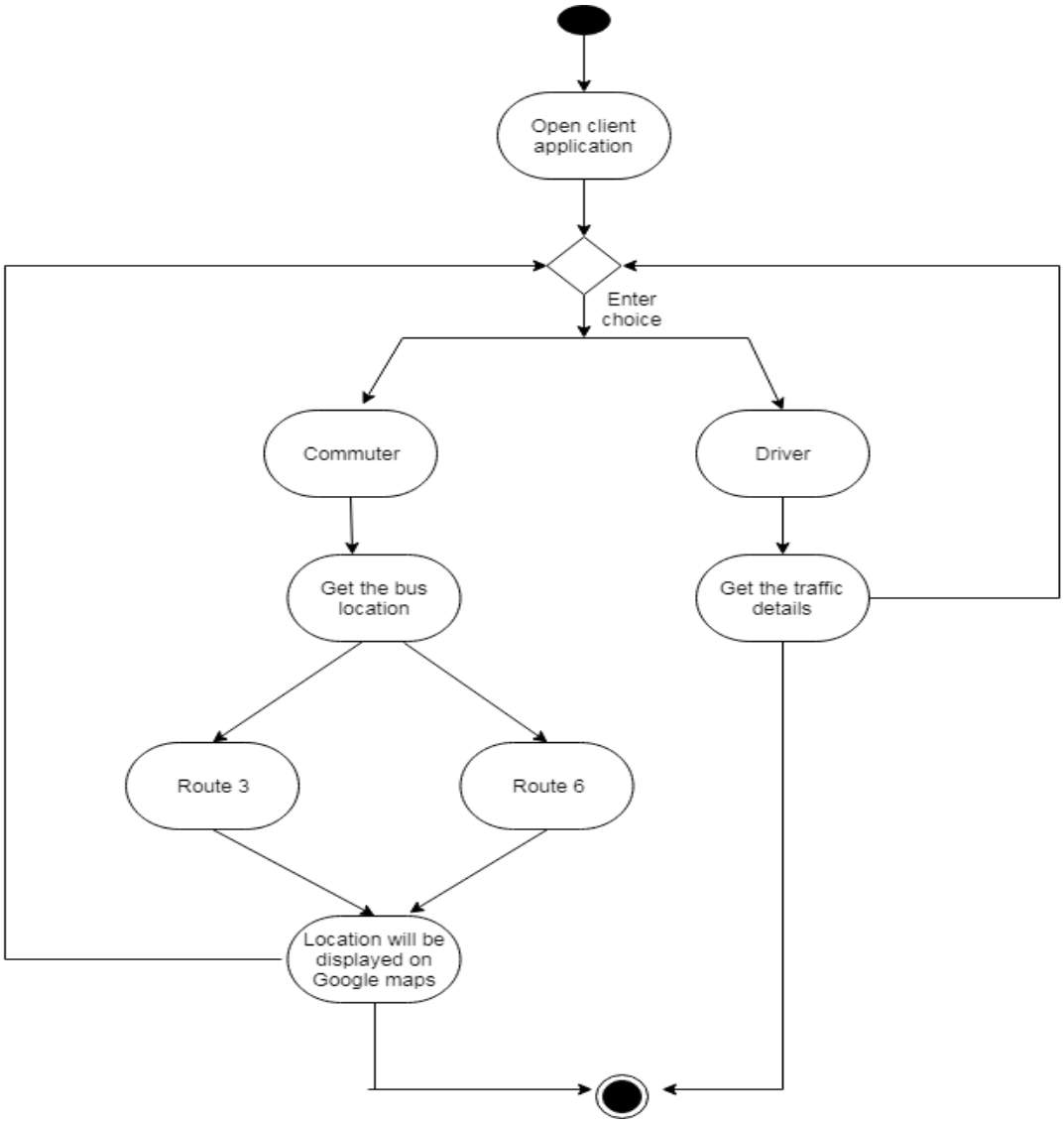


Figure 4.2. Activity diagram for bus tracking

4.3. DATAFLOW DIAGRAM

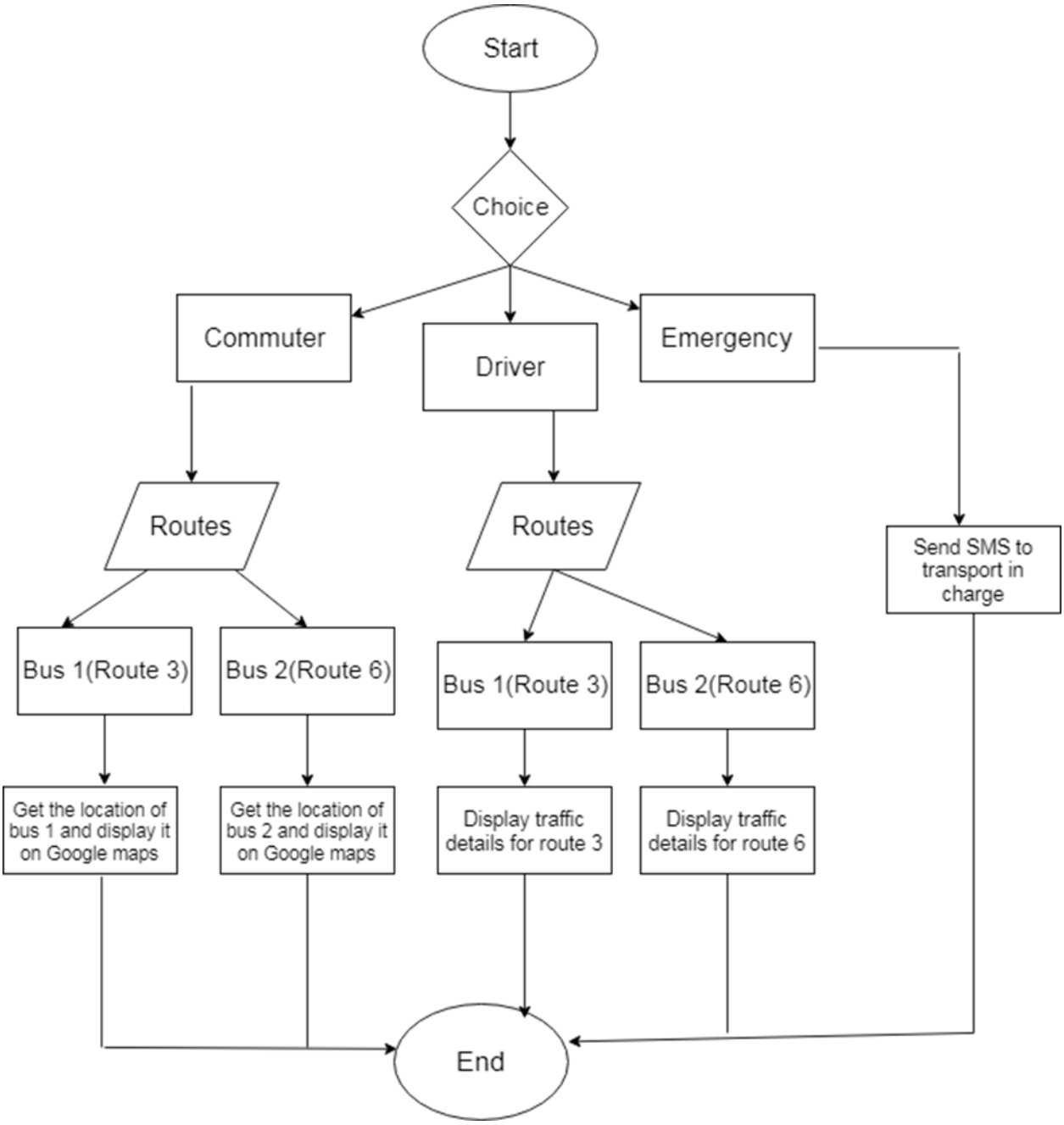


Figure 4.3. Dataflow diagram.

4.4. ALGORITHM

The following steps describe the algorithm

Step1. There are three different option's commuter, driver and emergency. The students/staff can click on the commuter option to get the bus location.

Step2. In step 2, the respective students/staff can click on their routes. By clicking on their routes they get the information of where exactly there bus is located on the Google maps.

Step3. In the same way, the driver can click on the driver option to get congestion details. When the bus 1 driver clicks on his route he gets the traffic details. In the same way Bus 2 driver can click on his route.

Step4. The last option is emergency, the students/staff and driver can click on emergency button if there is any bus break down or accident and so on. By clicking on that option an SMS will be sent to transport in charge along with the location.

Step5. Click on end option to terminate.

4.5. CLASS DIAGRAM

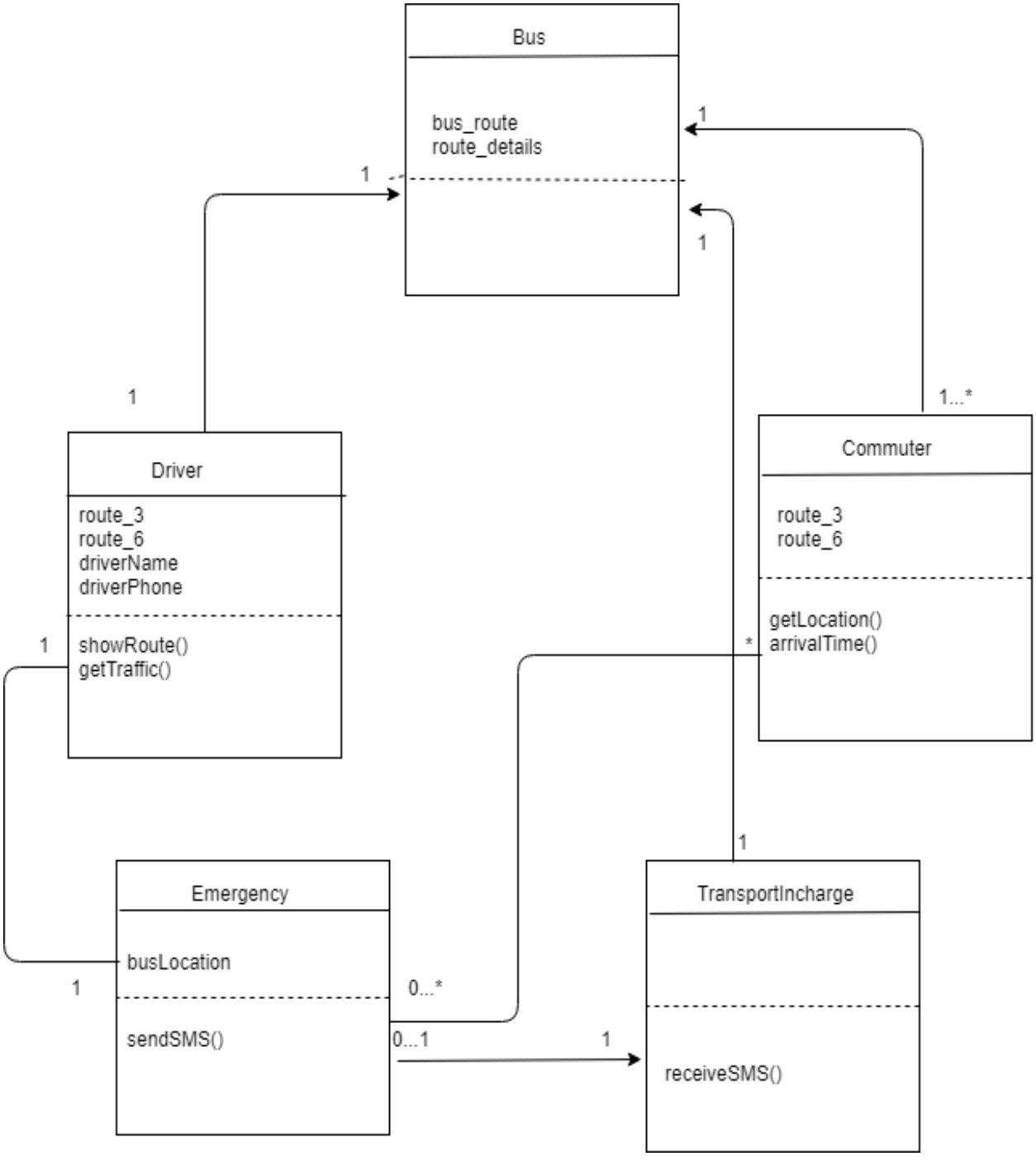


Figure 4.4. Class diagram

CHAPTER 5

IMPLEMENTATION

5.1. PLATFORM SELECTION

Android Studio

It is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards 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.
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ, and PyCharm e.g. Python, and Kotlin and Android Studio 3.0 supports "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects back port some Java 9 features.

5.2. PROGRAMMING LANGUAGE GIST

JAVA

- Android applications are developed using the Java language
- Some of the Java's important core features are:

  - ✓ It's easy to learn and understand
  - ✓ It's designed to be platform-independent and secure, using virtual machines
  - ✓ It's object-oriented

- Android relies heavily on these Java fundamentals.
- The Android SDK includes many standard Java libraries (data structure libraries, math libraries, graphics libraries, networking libraries and everything else you could want) as well as special Android libraries that will help you develop awesome Android applications.

5.3. MODULE DESCRIPTION

5.3.1. Commuter module

Project is designed in four parts where each part is responsible for different aspects. Essentially, main activity handles instantiate methods and pre-configuration process for Location Manager class such as manage the previous location, current location and status check whether GPS is on or off.

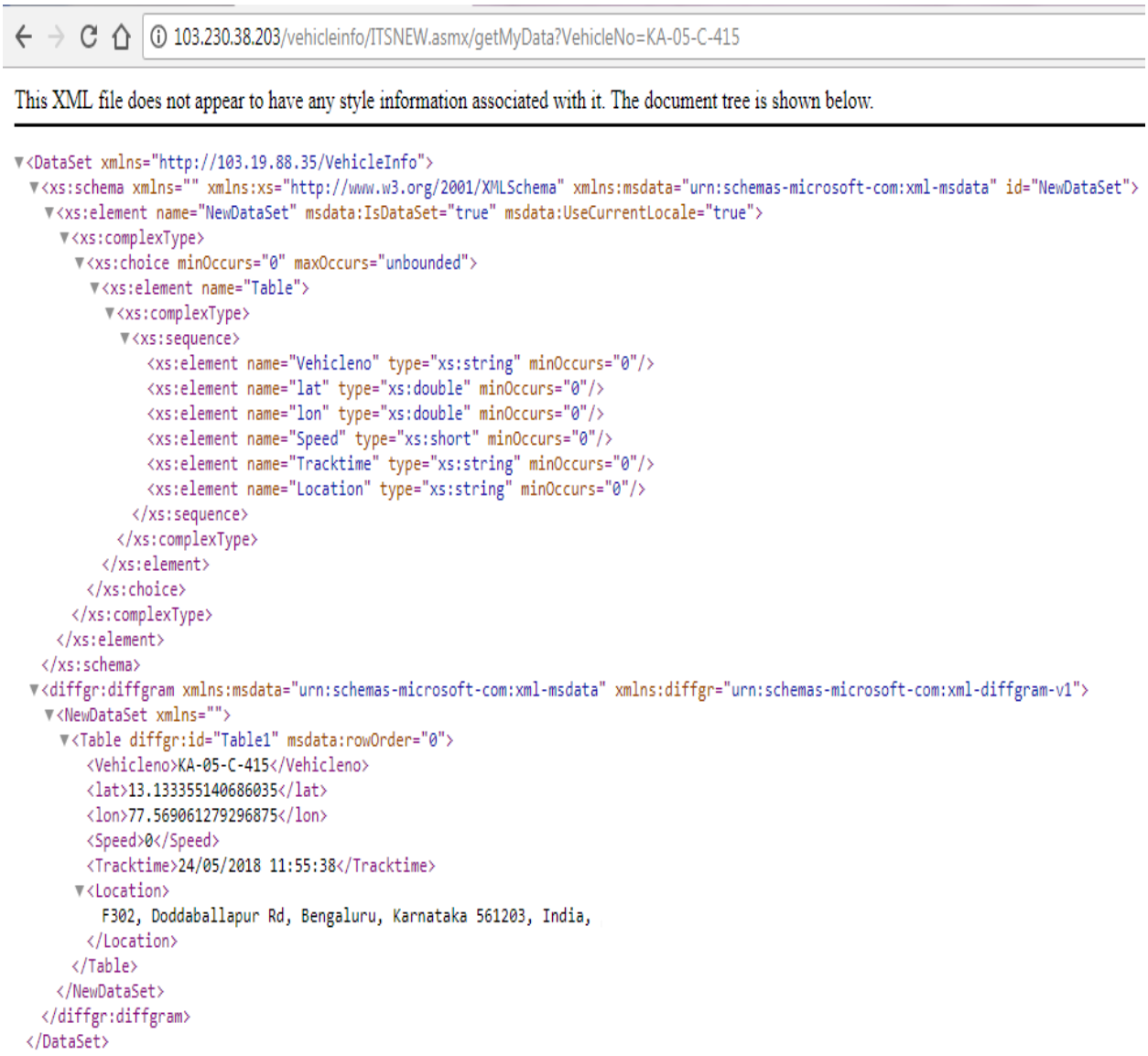
5.3.1.1. Extraction of information from remote XML

The first step involves the extraction of the GPS coordinates from the server. The two URL's , one for Route 3 and the other one for Route 6 is as follows:-

Route 3: http://103.230.38.203/vehicleinfo/ITSNEW.asmx/getMyData?VehicleNo=KA-05-C-415

Route 6: http://103.230.38.203/vehicleinfo/ITSNEW.asmx/getMyData?VehicleNo=KA-50-806

These are the screenshots of the XML documents



XML document for bus route 3

XML document for bus route 6

In order to extract the latitude and longitude from the XML , we use the XmlPullParser interface. XML Pull Parser is an interface that defines parsing functionality provided in XMLPULL V1 API. This is a simple interface – parser consists of one interface, one exception and one factory to create parser.

An XML file consists of 4 major components:

- **Prolog :** The first line that contains the information about a file is prolog. Typically this is the line:
- **Events :** Events in an XML file include simple start and end tags and more
- **Text :** It's simple text in between two tags. Example: My Text</RandomTag
- **Attributes :** Attributes are the additional properties of a tag that are present within the tag. Example : Some Text or nested tags.

Steps required to parse a XML using XML pull parser are,

- ✓ Get instance of XMLPULL factory
- ✓ By default factory will produce parsers that are not namespace aware; to change setNamespaceAware() function must be called
- ✓ Create an instance of the parser
- ✓ Then set parser input
- ✓ Start parsing. Typical XMLPULL application will repeatedly call next() function to retrieve next event, process event until the event is END_DOCUMENT. We will be able to get the latitude and longitude and store them in the variables lati and longi.

```
try {
xmlPullParserFactory= XmlPullParserFactory.newInstance();
xmlPullParserFactory.setNamespaceAware(false);
parser = xmlPullParserFactory.newPullParser();
    } catch (XmlPullParserException e) {
e.printStackTrace();
    }
```

```
privateString getLoadedXmlValues(XmlPullParser parser) throws XmlPullParserException,
IOException {
int eventType = parser.getEventType();
        String name = null;
        Entity mEntity = new Entity();
while (eventType != XmlPullParser.END_DOCUMENT){
```

```
if(eventType == XmlPullParser.START_TAG){
                name = parser.getName();
if(name.equals("lat")){
mEntity.lati= parser.nextText();
                }
else if(name.equals("lon")){
mEntity.longi= parser.nextText();
             }
            }
eventType = parser.next();
        }
return mEntity.lati+ ", " + mEntity.longi;
    }

public class Entity{
public String lati;
public String longi;
    }
}
```

5.3.1.2. Set up connection to internet

Android AsyncTask is an abstract class provided by Android which gives us the liberty to perform heavy tasks in the background and keeps the UI thread light thus making the application more responsive. Here we use the BackgroundAsyncTask function to make a connection to the internet as a background task.

```
private class BackgroundAsyncTaskextendsAsyncTask<String, Void, String> {
@Override
protected String doInBackground(String ...params) {
        URL url = null;
        String returnedResult = "";
try {
url = new URL(params[0]);
        } catch (MalformedURLException e) {
e.printStackTrace();
        }
HttpURLConnection conn = null;
try {
            conn = (HttpURLConnection)url.openConnection();
conn.setReadTimeout(10000);
conn.setConnectTimeout(20000);
conn.setRequestMethod("GET");
conn.setDoInput(true);
conn.connect();
InputStream is = conn.getInputStream();
parser.setInput(is, null);
returnedResult = getLoadedXmlValues(parser);
        } catch (IOException e) {
e.printStackTrace();
        } catch (XmlPullParserException e) {
e.printStackTrace();
        }
return returnedResult;
    }
```

5.3.1.3. Display the coordinates of bus on Google maps

Android uses URI string as the basis for requesting data in a content provider and for requesting actions .Intents let you start an activity in another app by describing a simple action you'd like to perform (such as "display a map" or "show directions to the airport") in an Intent object.

The Google Maps app for Android supports several different intents, allowing you to launch the Google Maps app and perform actions

Use of geo: Intent to display a map at a specified location and zoom level. q defines the place(s) to highlight on the map.

```
public void displayMap(double clat, double clong){
try{
        String geoCode = "geo:0,0?q=" + clat + ","
+ clong ;
        Intent sendLocationToMap = new Intent(Intent.ACTION_VIEW,
Uri.parse(geoCode));
startActivity(sendLocationToMap);
    }
catch(Exception e){
    }

}
```

Here geo?q= takes the latitude and longitude of the bus and displays the location of the bus using a marker on Google maps.

5.3.1.4. Use LocationManager to get current location

This class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to fire an application-specified Intent when the device enters the proximity of a given geographical location.

It checks for permission for access of location using the permission.ACCESS_COARSE_LOCATION and uses getLatitude and getLongitude to get the user's current location.

```
private void get_lat_long_details() {
// turnGPSOn();
mlocManager= (LocationManager) getSystemService(Context.LOCATION_SERVICE);
```

```
mlocListener= new MyLocationListener();
if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {

Toast.makeText(getApplicationContext(), "in first if",
Toast.LENGTH_LONG).show();
if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.ACCESS_COARSE_LOCATION)) {

Toast.makeText(getApplicationContext(), "in second if",
Toast.LENGTH_LONG).show();
        } else {
// permission is already granted
Toast.makeText(getApplicationContext(), "in else",
Toast.LENGTH_LONG).show();
ActivityCompat.requestPermissions(this,
new String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
MY_PERMISSIONS_REQUEST_NETWORK_PROVIDER);
        }


    } else {
mlocManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0,
0, mlocListener);

    }
}


/* Class My Location Listener */

public class MyLocationListenerimplementsLocationListener

{

public void onLocationChanged(Location loc)

    {
c_lat = loc.getLatitude();

c_long= loc.getLongitude();


Toast.makeText( getApplicationContext(),

"latitude->" + c_lat + "\n" + "longitude->" + c_long+ "\n" ,

Toast.LENGTH_SHORT).show();

mlocManager.removeUpdates(mlocListener);

getAddress();


    }
```

5.3.2. Driver Module

To get traffic details using navigation:

```
String geoCode =
"https://www.google.com/maps/dir/?api=1&origin=BMSCE,Bengaluru&destination=BMSIT,Bengaluru&waypoints=Lalbagh+West+Gate%2CBengaluru%7CMallya+Hospital%2CBengaluru%7CHalasuru+Police+Station%2CBengaluru%7CColes+Park%2CBengaluru%7CTV+Tower%2CJayamahal+Main+Road%2CBengaluru%7CRT+Nagar+Police+Station%2CBengaluru&travelmode=driving&dir_action=navigate";
```

This API lets us use Google maps in order to display the traffic details. Here we specify the origin and destination and the mode of travelling.

5.3.3. Emergency Module

To send SMS :

Sms Manager class is responsible for sending SMS from one emulator to another or device. MY_PERMISSION_REQUEST_SMS and SEND_SMS checks for permission and allows the application to send the SMS.

A Geocoder class uses either a location name or a location's latitude and longitude values to get further details about an address.

```
Geocodergeocoder;
List<Address>addresses;
geocoder = new Geocoder(this, Locale.getDefault());

addresses = geocoder.getFromLocation(c_lat, c_long, 1);

String address = addresses.get(0).getAddressLine(0);
String city = addresses.get(0).getLocality();
String state = addresses.get(0).getAdminArea();
String country = addresses.get(0).getCountryName();
String postalCode = addresses.get(0).getPostalCode();
String knownName = addresses.get(0).getFeatureName();

address = address + city + state + country + postalCode + knownName;
Toast.makeText( getApplicationContext(),

"Address is: " +  address,Toast.LENGTH_SHORT).show();
if (ContextCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS)
            != PackageManager.PERMISSION_GRANTED) {
if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.SEND_SMS)) {
```

```
        } else {
// permission is already granted
ActivityCompat.requestPermissions(this,
new String[]{Manifest.permission.SEND_SMS},
MY_PERMISSIONS_REQUEST_SEND_SMS);
    }

  } else {
SmsManagersmsManager = SmsManager.getDefault();
smsManager.sendTextMessage("+918105730024", null, "Help Needed. Location: " + address ,null, null);

Toast.makeText(getApplicationContext(),
"SMS Sent Successfully",
Toast.LENGTH_LONG).show();

  }
```

5.4. CODING STANDARDS

- ✓ This project is developed under the coding standard of the Java and user library used in it.
- ✓ Naming conventions of java language is followed for method and variable names.
- ✓ Begin method names with a strong action verb (for example, location).
- ✓ If the verb is not descriptive enough by itself, include a noun (for example, getLatitudegetLongitude).
- ✓ Add adjectives if necessary to clarify the noun (for example,getLastLocationRecord)
- ✓ Use the prefixes get and set for getter and setter methods. Getter methods merely return the value of a instance variable; setter methods change the value of a instance variable.
- ✓ For example:
- ✓ use the method names getAddress and setAddress to access or change the instance variable address
- ✓ Abbreviations are not used in declaring the name of method, class, activity, adapter, fragment and variables.
- ✓ Control structures, conditions, loops etc. are used wisely.
- ✓ Repetitive code must be written is methods.
- ✓ Avoid using same name of inbuilt keywords.

CHAPTER 6

SOFTWARE TESTING

Software testing is a process used for verifying the correctness, completeness and quality of the developed software. Software is built out of sub-systems that are composed of modules, which in turn are composed of procedures and functions.

The sequence of testing activities performed for the tracking system is as below:

- Unit Testing
- Integration Testing
- System Testing
- Functional testing

6.1. UNIT TESTING

There exist a number of components in every sub-system. Every component is tested using respective test procedures. Each component is tested individually based on their needs. Unit test focuses verification effort on the smallest unit of the software design component.

Each module can be tested using the following two Strategies:

Black Box Testing: In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program.

White Box testing: In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

No	Test Objective	Test Step	Expected result	Result
1	To ensure that the module is able to receive GPS data by GPS receiver.	1) Turn on GPS feature. 2) Run the bus positioning module installed. 3) Observe the GPS data receive from satellite.	The GPS location data should display on the screen.	Pass
2	To ensure that the module able to receive data from server.	1) Run the module installed in smart phone. 2) Observe the data	The module should be able to receive information from server.	Pass
3	To ensure that the module will continuously send updated GPS location data to server when location is changed.	1) Run the commuter module. 2) Wait for bus to move 3) Observe the data 4) Repeat	The module should continuously receive updated GPS location data.	Pass
4	To ensure that the traffic details are updated on Google maps	1)Run the driver module 2)Check for traffic in different areas	The module must change route as traffic changes	Pass
5	To ensure the SMS is sent properly to the destined number and with the correct location.	1)Run the emergency module 2)Click yes when prompted for confirmation	The transport in charge must get the SMS along with the location	Pass

Table 6.1: Unit Testing

6.2. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

No	Test Objective	Test Step	Expected result	Result
1	To ensure that all modules is able to communicate with server.	1) Run all modules 2) Observe bus icon position on map. 3) Observe the change.	All modules should able to communicate with server and the bus information should display smoothly and correctly.	Pass

Table 6.2: Integration Testing

6.3. SYSTEM TESTING

System testing is the stage of implementation. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding a yet undiscovered error.

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

6.4. FUNCTIONAL TESTING

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

Functional testing means testing the application against business requirements. Functional testing is executed using the functional specifications given by the client or by the design specifications according to use cases given by the design team. Role of functional testing is to validating the behavior of an application.

- Functional testing means testing the application against business requirements.

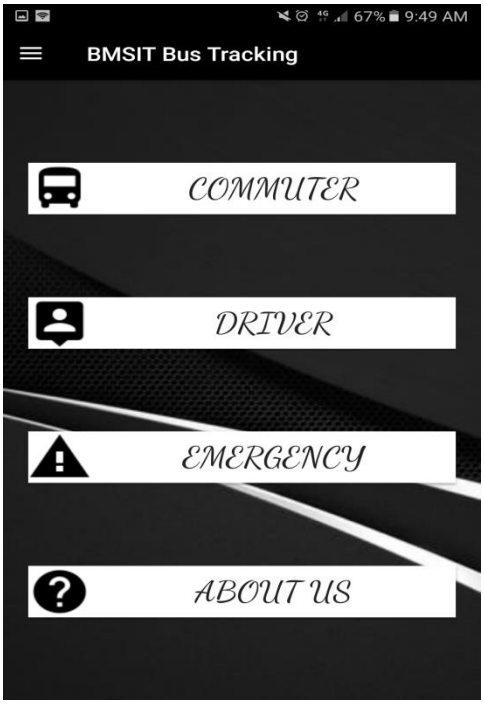
- Functional testing is executed using the functional specifications given by the client or by the design specifications according to use cases given by the design team.
- Role of functional testing is to validating the behavior of an application.

CHAPTER 7

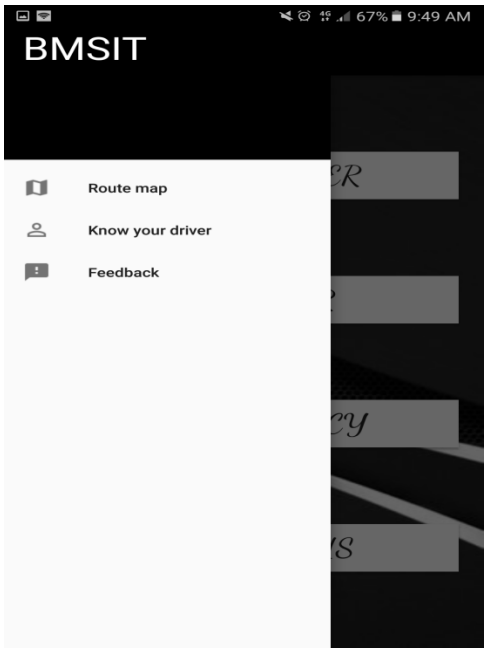
RESULTS AND DISCUSSION

SCREENSHOTS

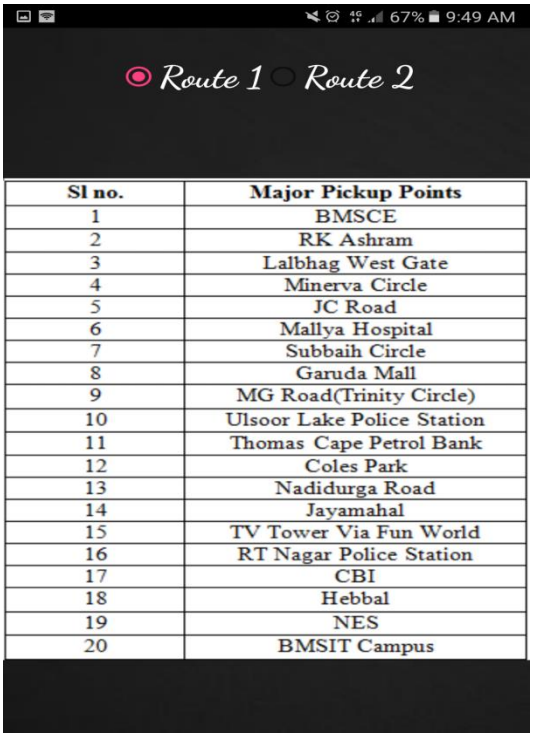
Splash screen



Menu option

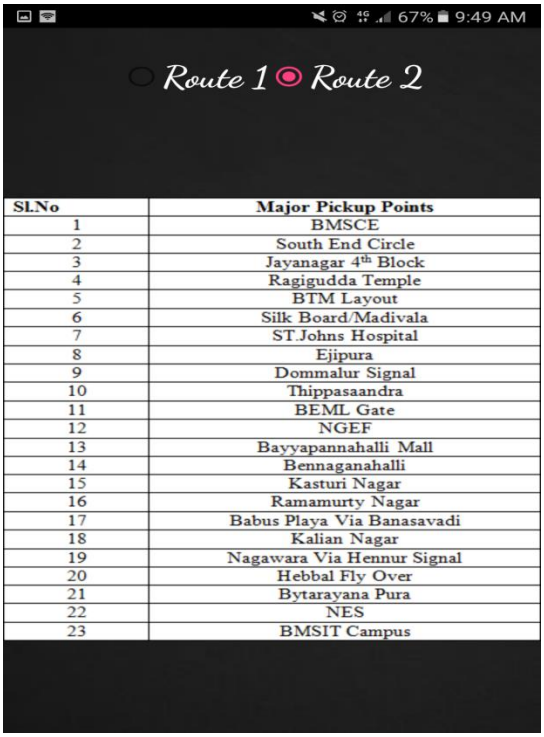


Navigation Drawer



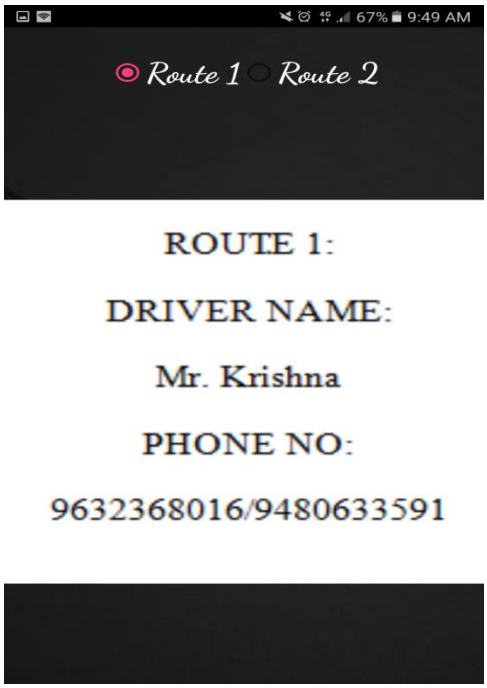
Sl no.	Major Pickup Points
1	BMSCE
2	RK Ashram
3	Lalbhag West Gate
4	Minerva Circle
5	JC Road
6	Mallya Hospital
7	Subbaih Circle
8	Garuda Mall
9	MG Road(Trinity Circle)
10	Ulsoor Lake Police Station
11	Thomas Cape Petrol Bank
12	Coles Park
13	Nadidurga Road
14	Jayamahal
15	TV Tower Via Fun World
16	RT Nagar Police Station
17	CBI
18	Hebbal
19	NES
20	BMSIT Campus

Route 1 details

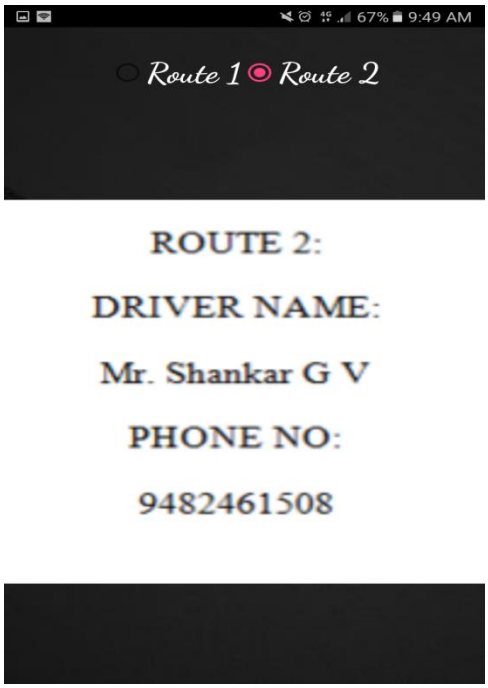


Sl.No	Major Pickup Points
1	BMSCE
2	South End Circle
3	Jayanagar 4th Block
4	Ragigudda Temple
5	BTM Layout
6	Silk Board/Madivala
7	ST.Johns Hospital
8	Ejipura
9	Dommalur Signal
10	Thippasaandra
11	BEML Gate
12	NGEF
13	Bayyapannahalli Mall
14	Bennaganahalli
15	Kasturi Nagar
16	Ramamurty Nagar
17	Babus Playa Via Banasavadi
18	Kalian Nagar
19	Nagawara Via Hennur Signal
20	Hebbal Fly Over
21	Bytarayana Pura
22	NES
23	BMSIT Campus

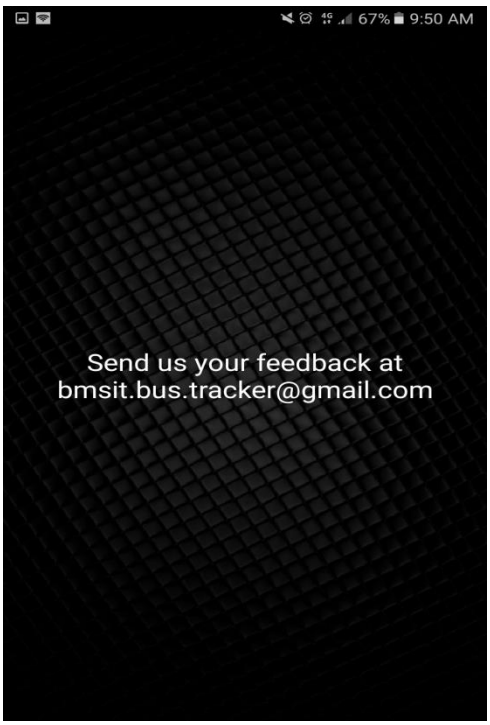
Route 2 details



Route 1 Driver details

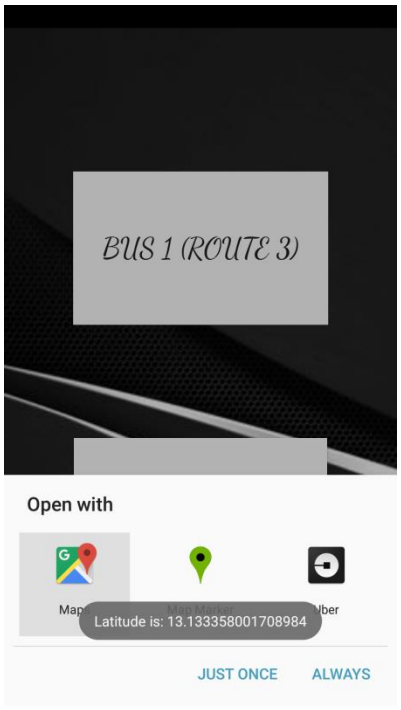


Route 2 Driver details

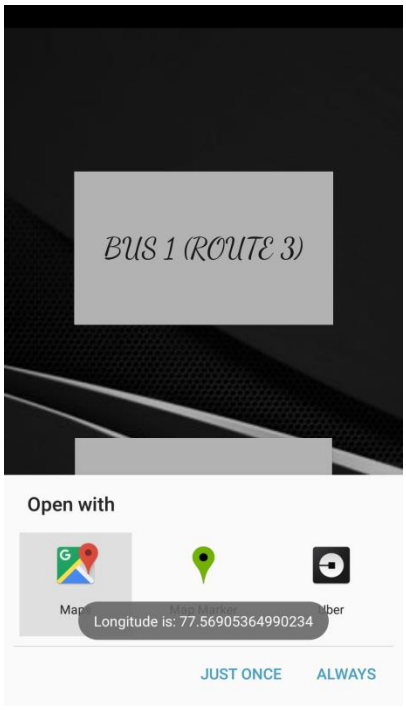


Feedback

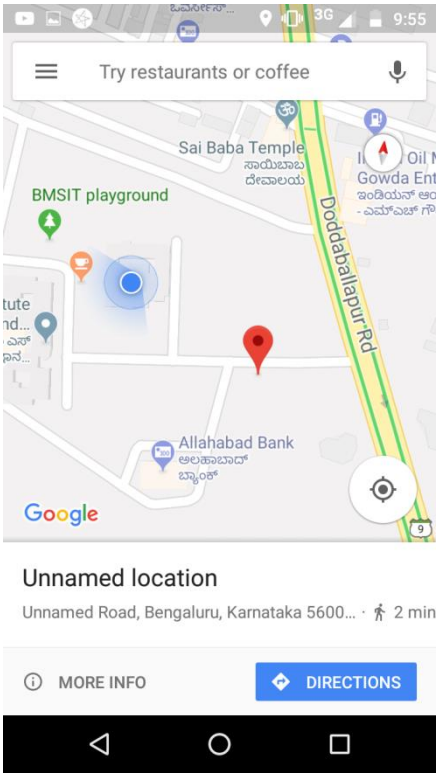
Option to select bus route



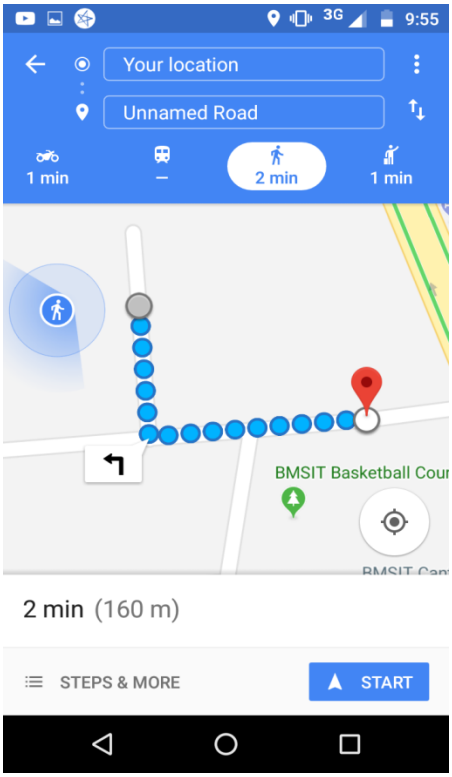
Display latitude



Display longitude

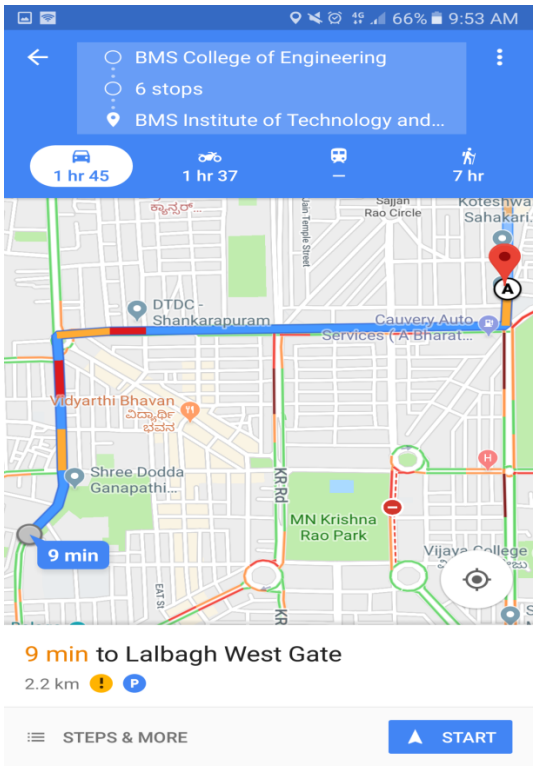


Bus 1 location

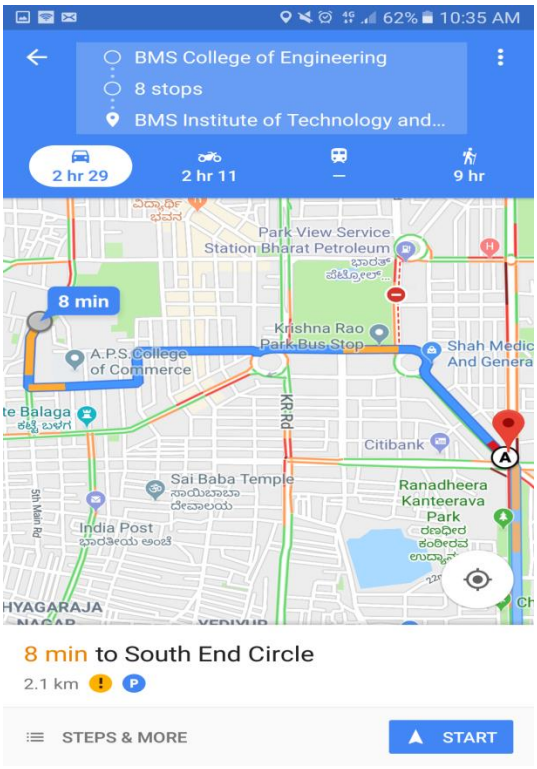


Expected arrival time(ETA)

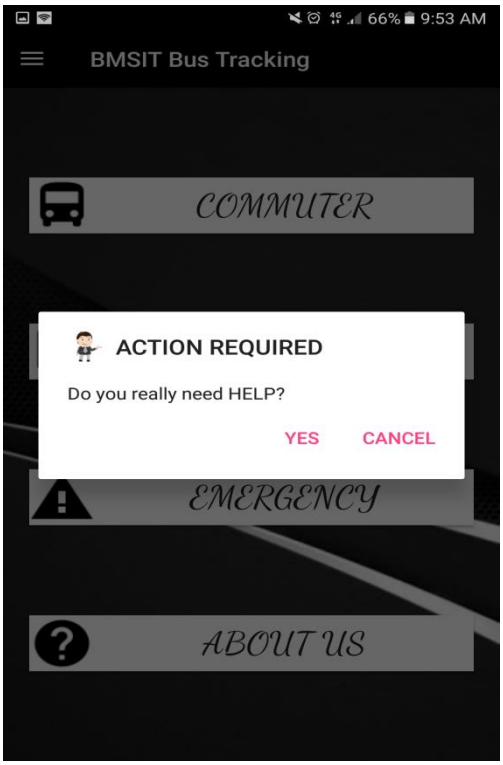
Option to select bus route for driver



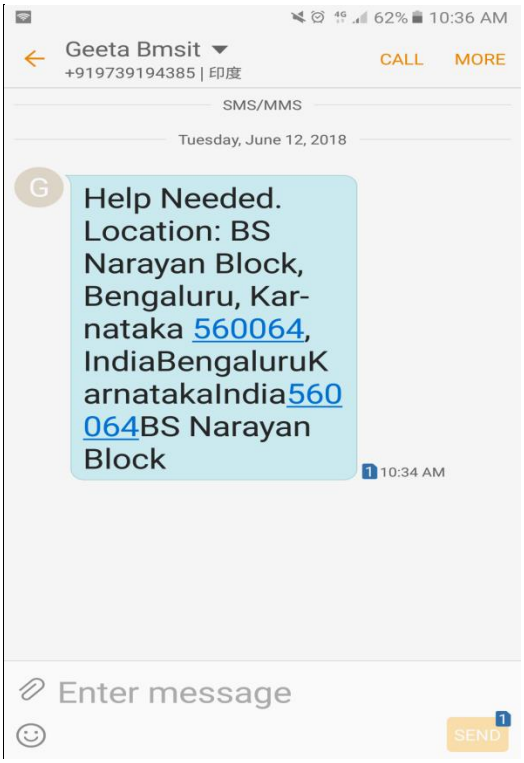
Route 1 traffic details



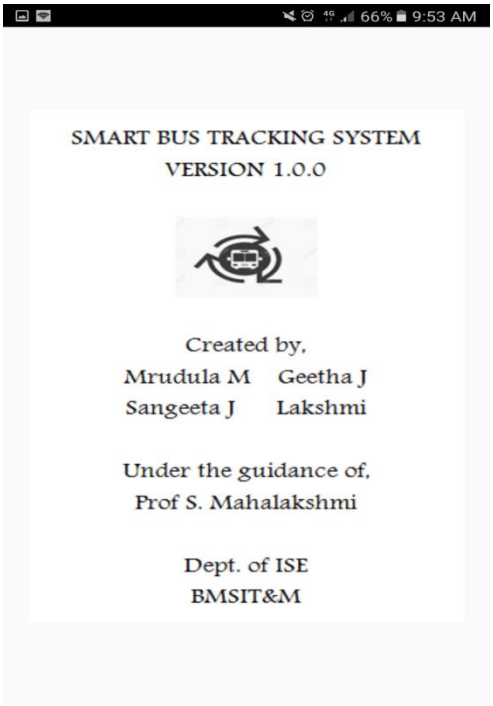
Route 2 traffic details



Emergency option



SMS send to transport in charge



About us

CHAPTER 8

CONCLUSION & FUTURE ENHANCEMENT

8.1. CONCLUSION

In this present situation Bangalore city is facing lots of problem due to traffic and so on. One among the major problem is, the students/staff will miss there buses in fraction of second, which may leads to problem like being late for classes, exams and so on. To overcome this kind of problem, **Smart Bus Tracking System** was introduced. In this system there are three different modules:

- ✓ The first module is the commuter module, when the users wants to know where exactly there bus is located they can click on this module, so that they get the information of their bus on the Google maps along with Estimated Time Of Arrival.
- ✓ The second module is the driver module, if the driver wants know the traffic details in his route. He can click on the module. By clicking so, the traffic will be displayed on the Google maps.
- ✓ The last module is emergency module. If there is any emergency situation like accident, breakdowns etc. the users/driver can send an SMS to the transport in charger.

8.2. LIMITATIONS:

- Our tracking system is not secured because it can be accessed by any unauthorized persons also.
- Our application doesn't work if the GPS is not turned on in the user's phone.
- Only the transport officer receives the bus break down details.
- Delay of the bus is not informed for the commuter in early stage.
- The student or staff will not get any alarm notification regarding the crossing of buses from each stop.

8.3. FUTURE ENHANCEMENT

- ✓ As of now we have implemented our project to track the BMSIT college bus of route 3 and route 6. In future, it can be implemented on remaining buses.

- ✓ Our application is not so secured; it can be accessed by everyone. In future, the login details should be provided so that only the people who are travelling in BMSIT college bus and also who have paid the transportation fee can access the location of the bus.

- ✓ An alert message or notification should be sent to the user whenever the bus arrives to the stop or crosses the stop.

BIBLIOGRAPHY

[1] https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-practicals/content/en/Unit%204/92_p_adding_settings_to_an_app.html

[2] https://stackoverflow.com/questions/32411898/how-to-set-up-settings-activity-in-android-studio

[3] https://www.youtube.com/watch?v=ND6a4V-xdjI

[4] https://www.androidhive.info/2013/07/how-to-implement-android-splash-screen-2/

[5] https://developer.android.com/training/basics/network-ops/xml.html

[6] Shared S, Bagavathi Sivakumar P, Anantha Narayanan V, " The Smart Bus for a Smart City - A real-time implementation", 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS).

[7] Ajay Shingare, Ankita Pendole, Nikita Chaudhari and Parikshit Deshpande, Prof. Samadhan Sonavane," GPS Supported City Bus Tracking & Smart Ticketing System",2015 International Conference on Green Computing and Internet of Things (ICGCIoT).

[8] ReshmaRathod, "Smart assistance for public transport system" 2016 International Conference on Inventive Computation Technologies (ICICT), Volume: 3.

[9] Majd Ghareeb, Athar Ghamlous, Hawraa Hamdan, Ali Bazzi,Samih Abdul-Nabi, "Smart bus: a tracking system for school buses", 2017 Sensors Networks Smart And Emerging Technologies (SENSET).

[10] R.C.Jisha, Aiswarya Jyothindranath, L Sajitha Kumary, "IoT based school bus tracking and arrival time prediction", 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

APPENDIX
PLAGIARISM REPORT

Plagiarism Checker X Originality Report

Similarity Found: 24%

Date: Tuesday, June 12, 2018
Statistics: 2773 words Plagiarized / 11362 Total words
Remarks: Medium Plagiarism Detected - Your Document needs Selective Improvement.