

Real Time Online Bus Tracking System

*A B. Tech Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of*

Bachelor of Technology

by

Laxman Kumar Prabhakar
(1401CS22)

under the guidance of

Dr. Jimson Mathew & Dr. Abyayananda Maiti



to the

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY PATNA
PATNA - 801103, BIHAR**

CERTIFICATE

*This is to certify that the work contained in this thesis entitled “**Real Time Online Bus Tracking System**” is a bonafide work of **Laxman Kumar Prabhakar (Roll No. 1401CS22)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Patna under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: Dr. Jimson Mathew & Dr. Abyayananda Maiti

Date: Signature of HoD
..... (Dept. of CSE, IIT Patna)

Acknowledgements

There are many people whom I want to thank for the successful completion of my Project. First of all, I would like thank Dr. Jimson Mathew & Dr. Abyayananda Maiti, my guide for this project for giving me this opportunity. I learnt a lot from his insights, discussions, and guidance. He is always encouraging and it was because of his confidence that I was able to complete the work that was assigned to me. He would always assign me to the right person whenever I needed any assistance or I was stuck at any place, and would also suggest new ideas to implement for the project. I would like to thank my PhD supervisor, Mr. Rakesh Kumar Sanodiya, and my M.Tech supervisor, Mr. Binayak Krishna Swami, who gave me an introduction to the existing technologies which could be used for the project, and helping me out in times I needed help. We would have good discussions regarding the design and functioning of the app which gave me direction to work, and helped me a lot. I cannot forget thanking my brother Santosh Kumar Singh, my parents and friends for their constant support, faith and encouragement.

Contents

List of Figures	vi
1 Introduction	1
1.1 What necessitates this project	1
1.2 Background	2
1.3 Concept	2
1.4 Literature survey	3
2 Tools and technologies used	5
2.1 Software	5
2.1.1 Android Studio	5
2.1.2 Java	6
2.1.3 XML	6
2.1.4 GPS	7
2.1.5 Google Maps	7
2.1.6 Google maps API	8
2.1.7 Firebase database	8
2.1.8 Android (Operating System)	9
2.2 Hardware	9
3 Proposed System	11

3.1	Method	11
3.2	System Architecture	12
3.3	How Our Real Time GPS Tracking System Works?	14
4	Software design	15
4.1	Login credential and Authentication for user	15
4.1.1	Why authentication process is required?	15
4.1.2	How to implement it in Android application	16
4.1.3	Simulation results	17
4.2	Accessing GPS information with android	18
4.2.1	Why accessing of GPS information is required?	18
4.2.2	How to access GPS information?	18
4.2.3	Simulation results	19
4.3	Current Location & Routes	19
4.3.1	Simulation results	20
4.4	Bus Information	21
4.4.1	Simulation results	22
4.5	Server side implementation	22
4.6	DataBase management	23
4.7	Principle of Operation of counting	23
5	Conclusion and Future Work	25
5.1	Future Work	25
5.2	Conclusion	26
References		27

List of Figures

3.1	Basic block diagram	13
3.2	Application Working diagram	14
4.1	Registration page	17
4.2	Login page	17
4.3	bus location in driver app	19
4.4	bus location in user app	19
4.5	Route with current location	20
4.6	Marker info window	22
4.7	Bus info of all buses	22
4.8	Hardware setup	24

Abstract

In this project we have implemented an Online Bus Tracking System that works in real time and is meant for android smartphones. This helps the students as well as public to get information about the buses and save the commuter time. The main aim of implementing this System is to provide real time exact location of plying buses on the Google Maps and also to provide the bus details like bus name, bus routes, bus stops, driver details and contact number of the driver. A large number of people commuting today in buses possess android phones. So this application will be widely used by all, as it provides real time info to the user about desired buses. It is achieved by updating the location and other useful information of buses every moment with the help of a GPS device over the Internet and maintaining of a server and a Database. The application also provides the route of each bus, the expected time to arrive at a particular stop, and the number of seats available in the buses. A hardware implementation for the counting the number of passengers is made with the help of load cells and an Arduino kit. basically in this project we are using client-server technology.

Chapter 1

Introduction

1.1 What necessitates this project

The objective of the project is well elucidated in the abstract page. The main aim of developing such a software is to smartify the existing public transport in an otherwise unorganised sector in India. Bringing up such smart systems would not just help people travelling to work daily, but also people in emergencies, travellers and tourists, and people who have newly immigrated to a place and have difficulty in travel because of language barrier.

Overall this app would be of great convenience to the public and in the future be also great for smart traffic management (as public vehicles will be under the scanner of the traffic police, who would be able to better identify traffic patterns and take action). Also general people would be able to understand the traffic density of different roads real time, and make better informed driving choices.

1.2 Background

Now a days when people move around the way then face lots of problems due to the existing public transportation system because of heavy traffic, construction work, and strikes etc., in general buses are delayed in time. Due to this public have to wait for long time on the bus stops even without knowing the arrival time of the on particular stops. Anybody who want to use the public transportation system, can't find the time of arrival of particular bus at the particular destination even at their homes and plan their departure from home accordingly. But due to unexpected delays in traffic congestion the bus arrival time cannot be guaranteed .

1.3 Concept

Our main focus is to provide such a system to remote user which will reduce waiting time for bus and will provide him with all necessary details regarding the arrival/departure time of the bus, its real location and expected waiting time.

For solving all these above problem we are using android smartphones and IoT. In the present work, IoT is used to solve the common commuting problems in public and private buses existing in our towns and cities. Common issues include heavy crowding in the buses at some specific peak hours and in some particular routes. Difficulty due to the large diversity of Indian languages in their script and region of existence also creates a communication gap between people standing on bus stops.

And most importantly, the unknown status of bus availability, seat availability in the bus, current location of the bus and the behaviour of the crowd. Along with these, in this project, the aim is to improvise the safety and security features in public and private buses which are used by a large number of people. By installing Internet of Things (IoT) kit on the buses and connecting them to the cloud, above list of issues can be catered. It also

uses the installed sensors in the bus, which can count the number of people on the bus, and also locate the position of the bus using Global positioning system (GPS).

On fetching the current data of position and count of people on the bus, it is uploaded to the cloud. The status is updated very frequently, which can be done easily by the IoT device installed on the bus. These data can be given access to the people via an Android App. Thus, the people standing on the bus stand will come to know that, where their respective bus is right now. How crowded it is. Approximately, how long will it take to reach their bus stop to pick them up.

Depending on the status of the bus, a commuter can plan out his journey that, whether he/she should wait for the bus or take a taxi/auto or take a detour to reach his/her home. Also, it will help the commuters in distributing the crowd if two or more buses are coming back to back. People can access the list and bus numbers from the State road transport corporation website, containing the details of buses which connects different places. Any user can choose his/her respective bus and get the bus numbers.

People can search the status of the buses on their smartphones and be assured with what is coming next. Just by knowing the bus number, a person can get the details of available choices. These data can be shared with traffic and police control room as well, for safety and monitoring purpose.

1.4 Literature survey

Real Time Bus Monitoring System using GPS[1] basically show the current location of buses, with the help of transmitter installing on the buses and receiver installing at bus the bus stops. It helps the user or clients to get bus relevant data like current location, routes, and other information of buses.

Real Time Web Based Bus Tracking System [2] get the data from GPS which is attached with every bus. It uses some external set-up of hardware for implementation, and provide the relevant information like bus routes about the buses to the users.

Dynamic Bus Time-table Using GPS [3] based on GPS and to display timetable and real time location of bused designed manual system which can be useful for public transport system, in this project tools and technology requirement is internet connection and GPS tracker.

Foot movement is used to analyse the gait pattern. Load cells are connected to the bottom of the shoe. Magnitude and direction of the reaction forces are used for analysis of centre of foot pressure. The movement analysis can be used for gait analysis[4].

The reaction of a foot on the rigid surface produces the output proportional to the force and moments in all three directions. However, measurement of only one pattern is sufficient to tell about the load analysis on hip joints and heels[5].

Chapter 2

Tools and technologies used

In this chapter all the required technologies and tools for this project have been discussed in detail.

2.1 Software

2.1.1 Android Studio

We have many choice to choose the platform to build this app but we choose android studio it is very famous and useful now a days. Android Studio 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
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.

2.1.2 Java

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture

2.1.3 XML

In computing, Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data.

2.1.4 GPS

The Global Positioning System (GPS), originally Navstar GPS, is a satellite-based radionavigation system owned by the United States government and operated by the United States Air Force.

It is a global navigation satellite system that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings block the relatively weak GPS signals.

2.1.5 Google Maps

Google Maps is a web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle (in beta), or public transportation.

2.1.6 Google maps API

After the success of reverse-engineered mashups such as chicagocrime.org and housingmaps.com, Google launched the Google Maps API in June 2005 to allow developers to integrate Google Maps into their websites. It is a free service, and currently does not contain ads, but Google states in their terms of use that they reserve the right to display ads in the future.

By using the Google Maps API, it is possible to embed Google Maps into an external website, on to which site-specific data can be overlaid. Although initially only a JavaScript API, the Maps API was expanded to include an API for Adobe Flash applications (but this has been deprecated), a service for retrieving static map images, and web services for performing geocoding, generating driving directions, and obtaining elevation profiles.

2.1.7 Firebase database

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

Firebase is a mobile platform from Google offering a number of different features that you can pick n mix from. Specifically, these features revolve around cloud services, allowing users to save and retrieve data to be accessed from any device or browser. This can be useful for such things as cloud messaging, hosting, crash reporting, notifications, analytics and even earning money through AdMob.

In short, Firebase handles the backend online element for your apps, allowing you to focus on the front-end UI and functionality. All this is done through a single SDK with easy-to-use APIs and excellent integration into Android Studio. This removes the need to create your own server-side script using PHP and MySQL, or a similar set-up. This is Backend as a Service or BaaS, and essentially this means that anyone really can make that ambitious social app. It works with Android apps, iOS apps and web apps and best of all: its free!

2.1.8 Android (Operating System)

Android is an open source and Linux based operating system for smart phones, tablet computers and android wearable devices like watches. It is developed by Google and later the OHA (Open Handset Alliance). The main focus of android project is to create a fruitful real-world product that enhances the mobile experience for users.

The important features of android are given below:

- Open Source.
- Easy to customize the Android Platform.
- A lot of mobile applications can be chosen by the customer.
- Provides many appealing features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.
- Provides support for messaging services(SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Bluetooth, Wi-Fi etc.), media, handset layout etc.

2.2 Hardware

- Intel Edison Microcontroller
- Kinect sensor
- Arduino UNO

Chapter 3

Proposed System

3.1 Method

In this project I have complete the android application and also used the hardware(IoT) for make them real life implementable. This application contains many modules for make successful, like Authentication for users as well as Driver, Google maps to show location of active buses and its routes,Bus Information and driver details, connection with server to store the bus information and real time updates for the users, and IoT based hardware part to count the people. In this project we have used the android studio to development the desire application, language I have used for that is Java & XML, and used hardware (load cell, and intel kit) for count the people.

This project has two applications, one for user and other for the buses. The user app works as follows:

- First of all, the user has to login for authentication, which causes current location to be detected by the GPS.
- After that, the user will be able to see all the active buses in their locality.
- The user can get the details of each buses by clicking on their icon like current

position, seat availability etc.

- The user can find the list of all the buses. That are to cross the route with their timings. And its details like Bus Number, Drivers Contact Number, Bus Route, Stops, etc.

The app along with the hardware installed in bus works as follows:

- As the bus starts, it gets connected to server and its informations like position, seat availability is being updated with time.

3.2 System Architecture

In the system, a Wi-Fi connected Intel Edison board is used as the IoT client from individual buses. The generated data from individual buses are uploaded continuously with date and time stamp. The receiver on the other side, fetches these data, as per the bus number he/she enters into the app.

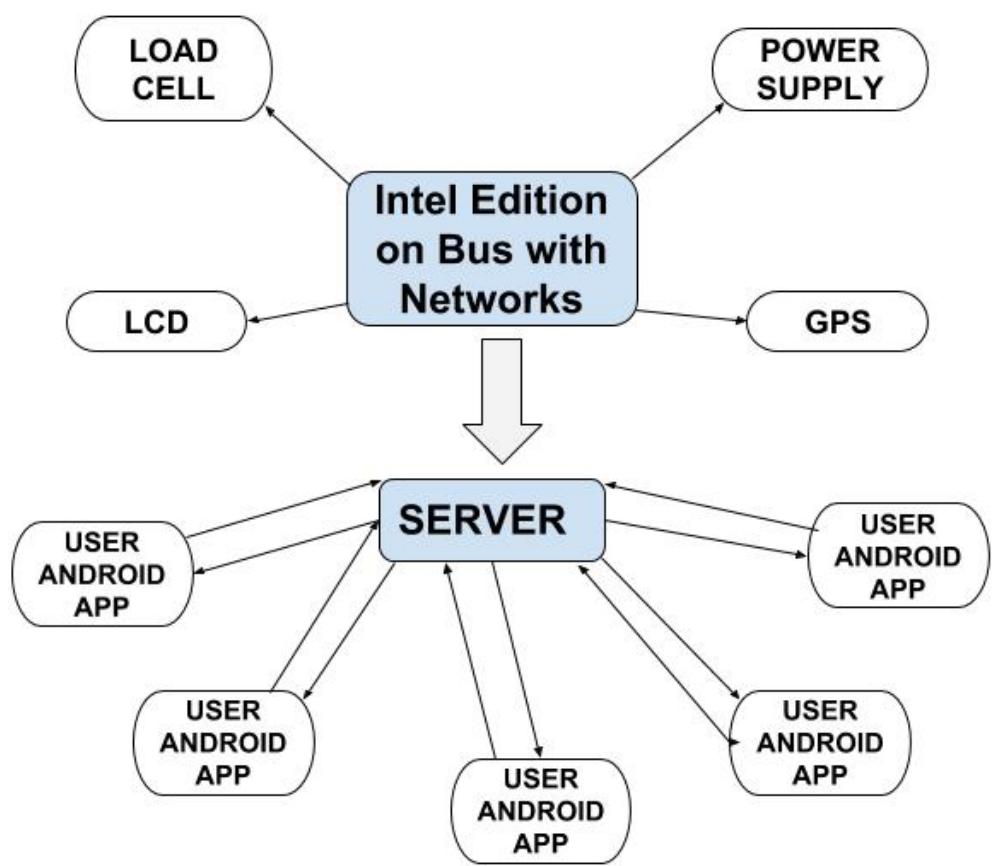


Fig. 3.1: Basic block diagram

3.3 How Our Real Time GPS Tracking System Works?

The proposed system has 3 modules: Vehicle with GPS, the Central control unit (Server), and Client-side application (User). In the Fig.3.2 we can see the flowchart of how the data processing between the three module and how the GPS data has been used/processed with the help of GSM Mobile Tower and GPS Satellite.

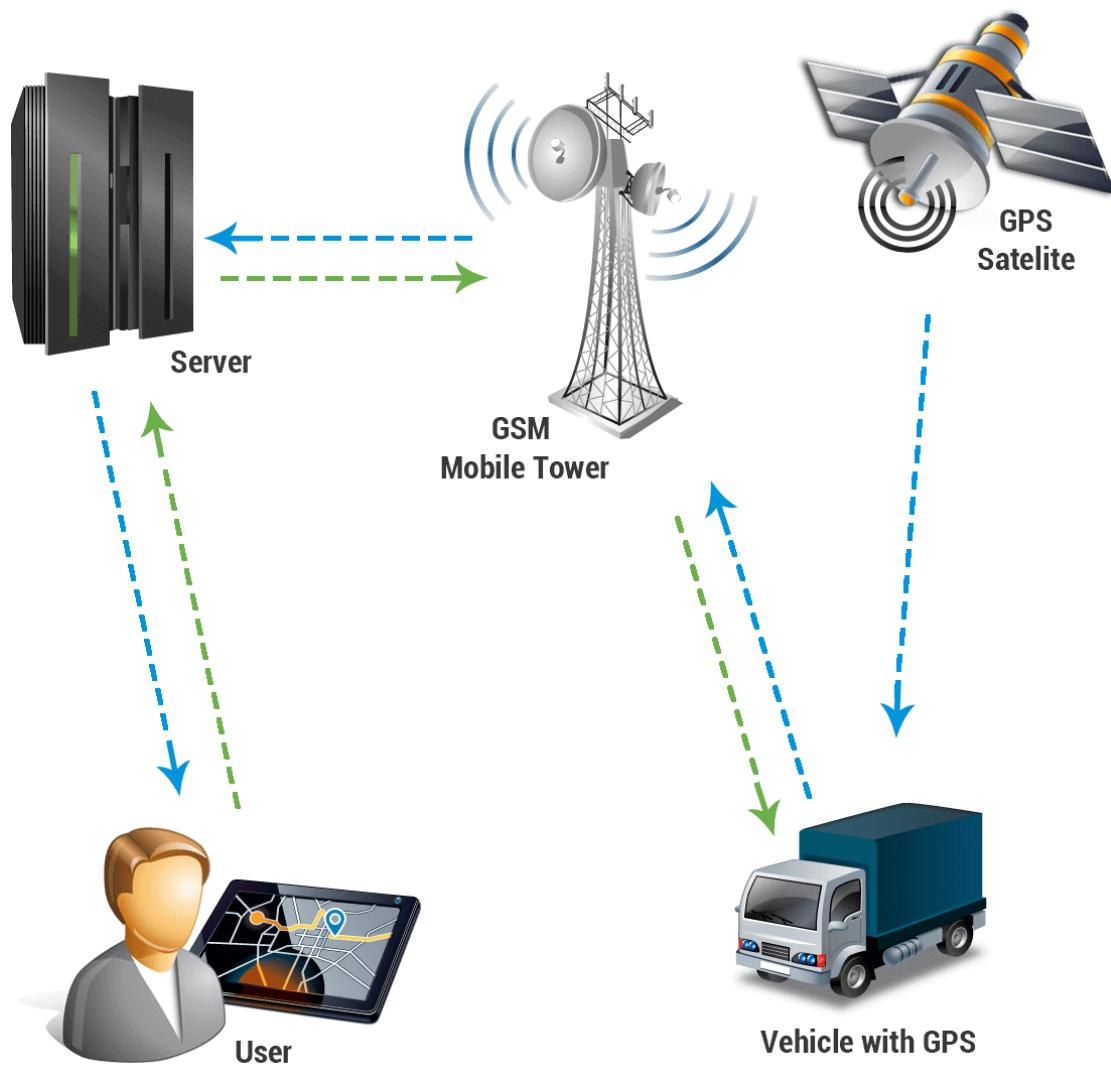


Fig. 3.2: Application Working diagram

Chapter 4

Software design

Before going to start the project we have to set environment in your system for that we need to install some software for windows users, for more information you can read the documentation of android studio[6], list of softwares given below:

- Java JDK5 or later version
- Android Studio

After the successful installation of all the software create the new project in android studio. And create different activities according to requirement. Now we are explaining all the activities of our application step by step:

4.1 Login credential and Authentication for user

4.1.1 Why authentication process is required?

Every system administration have requirement of some data about the system like how many people access the application, how many are active users, when the maximum user access. For managing all these thing system required track of all user through the login

credential & authentication, Authentication is a process in which the credentials provided are compared to those on file in a database of authorized users information on a local operating system or within an authentication server. If the credentials match, the process is completed and the user is granted authorization for access. The permissions and folders returned define both the environment the user sees and the way he can interact with it, including hours of access and other rights such as the amount of allocated storage space.

The process of an administrator granting rights and the process of checking user account permissions for access to resources are both referred to as authorization. The privileges and preferences granted for the authorized account depend on the users permissions, which are either stored locally or on the authentication server. The settings defined for all these environment variables are set by an administrator.

4.1.2 How to implement it in Android application

For implementing this we first create new activity and then write some basic XML code of layout and Java code for activities. Now for authentication we are using below given code and many supportive code is there:

```
firebaseAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener( activity: this, (task) -> {
        if (task.isSuccessful()) {
            //Successful registrated
            //profile activity
            Toast.makeText(getApplicationContext(), text: "Registration Sucessful", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(getApplicationContext(),ProfileActivity.class) );
            progressDialog.dismiss();
            finish();
        }
        else{
            Toast.makeText(getApplicationContext(), text: "Registration Unsuccesful", Toast.LENGTH_SHORT).show();
            progressDialog.dismiss();
        }
    });
}
```

This above code are used for create new user and send the login credential to the server and save the details in the database. If everything happened successfully then generate toast message for user Registration Successfull Here we are using real time online firebase

database[7].

```
firebaseAuth.signInWithEmailAndPassword(email,password)
    .addOnCompleteListener( activity: this, (task) -> {
        if(task.isSuccessful()){
            //Successful login
            progressDialog.dismiss();
            finish();
            Toast.makeText(getApplicationContext(), text: "Login Successful",Toast.LENGTH_SHORT).show();
            startActivity(new Intent(getApplicationContext(),HomeActivity.class));
        }
        else{
            Toast.makeText( context: LoginActivity.this, text: "Login is Unsuccessful ",Toast.LENGTH_SHORT).show();
            progressDialog.dismiss();
        }
    });
}
```

This above code are used for signin user and send the login credential to the server and if login credential is correct then signin is successful then generate toast message Login Successful other give message Login Unsuccessful. Here also we are using real time online firebase database[7].

4.1.3 Simulation results

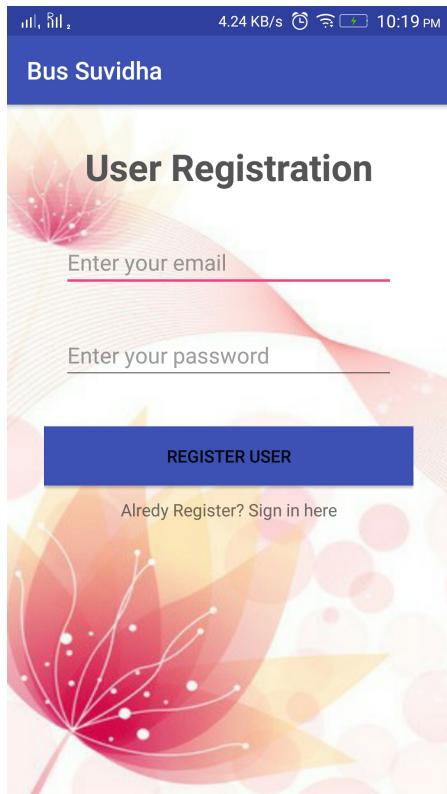


Fig. 4.1: Registration page

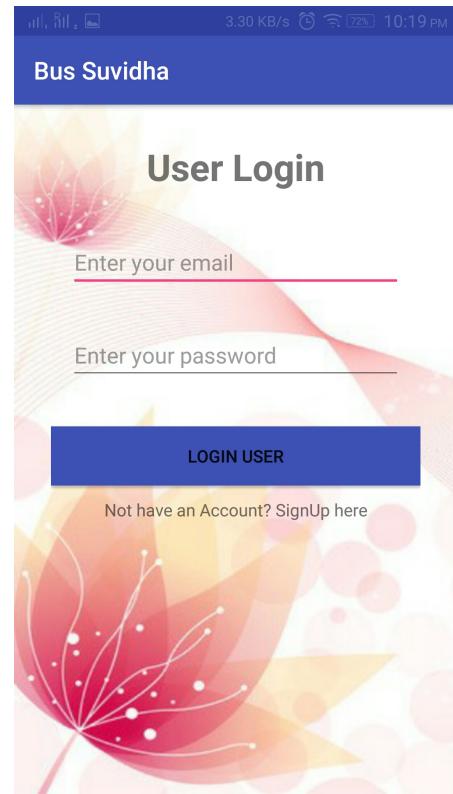


Fig. 4.2: Login page

4.2 Accessing GPS information with android

4.2.1 Why accessing of GPS information is required?

In our project we were trying to build an application which can track the real time and location of bus (gps devices), for that we need to send the driver location to the server with the help of the android application of driver so we have to fetch GPS information of driver android phones. Hence came the necessity to use Google maps API for this. First of all we connect google map with our android application and next step is sending the real time location and time of google map data to server which is helps us to track the other devices which are already registered with our server.

4.2.2 How to access GPS information?

For implementing this we first create new Google map activity in your project and then write XML code for its layout and Java code for its activities. Before proceeding forward first give the user permission of ACCESS_FINE_LOCATION or ACCESS_COARSE_LOCATION to your application in the manifest file. here we are try to explain what is main part of code for that and how it works, main function implementation code is given below:

```
if(ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
    ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED )
{
    return;
}
mLastLocation= LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
```

In the if part give the permission to the application for accessing the device gps and Location information. Line below the if condition fetch the location and store it into mLastLocation with the help of LocationService and FusedLocationApi.

4.2.3 Simulation results

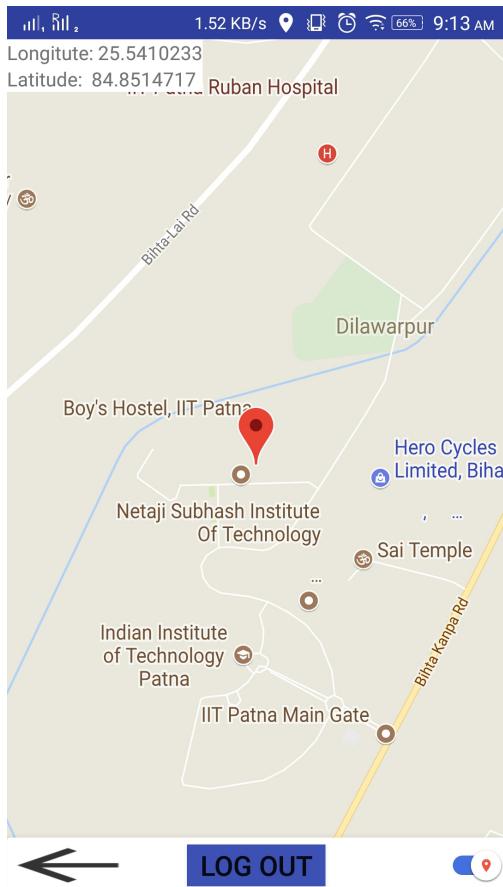


Fig. 4.3: bus location in driver app

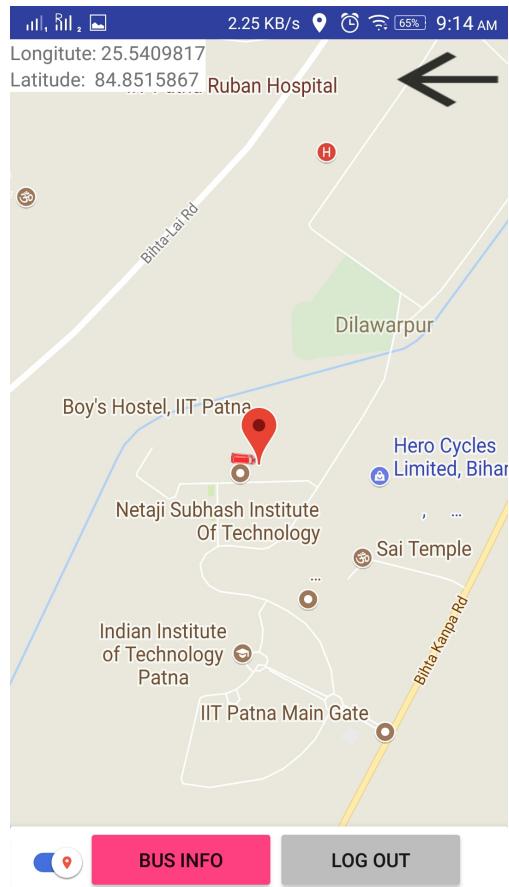


Fig. 4.4: bus location in user app

4.3 Current Location & Routes

When buses are running then our app provide the movement of buses along the way in which bus are going. And also provide your current current location. So you can easily find the way to reach the particular bus. In the given figure blue marker show the current location of user and red marker show the way on which bus are traveling. One red marker is static that means rest one is not running but it is active.

4.3.1 Simulation results

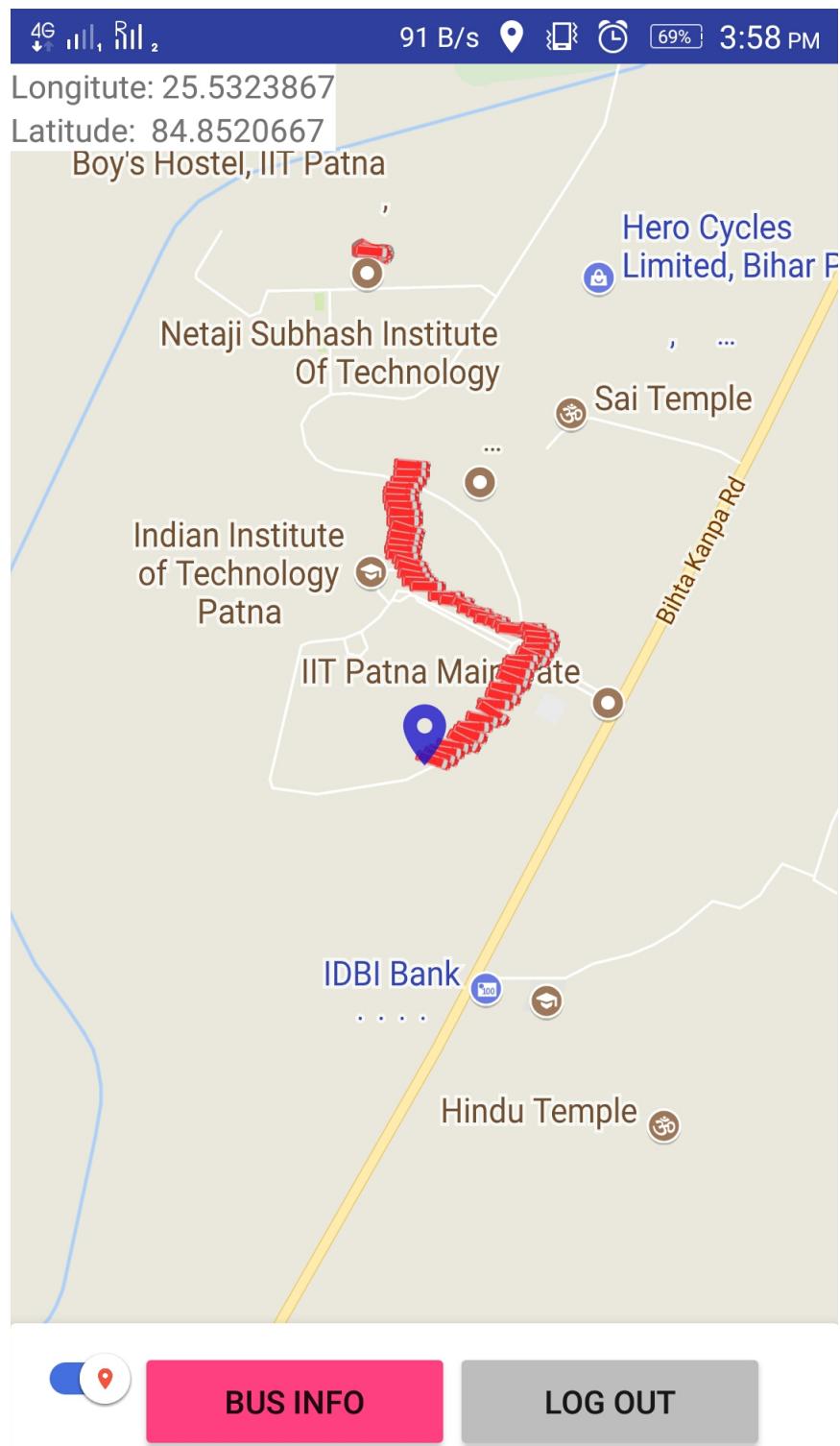


Fig. 4.5: Route with current location

4.4 Bus Information

In this application we are trying to provide all the information about the buses like bus current location(Latitude, Longitude), bus name, bus contact details, bus routes, and expected arrival time to each and every users. Because of user make a plane according to prior information of buses, and save and use the time. For implementing them we are using two types of methodology, first one is when you tap on the bus marker then you can find the details of the bus in Bus Information windows, for closing them you can tap anywhere on the screen other than Information windows. Second method is you can press Bus Info button and get the list of all registered buses details. For implementing the Bus Info button we are using list view methodology to show all the data getting from the server.

4.4.1 Simulation results

In the below given picture when you press the BUS INFO button then Bus Available window will open, it contains Bus driver name, city name, contacts for a time being. But we can add many more entry here. And right now BUS DETAILS and BUS LOCATION button are not working.

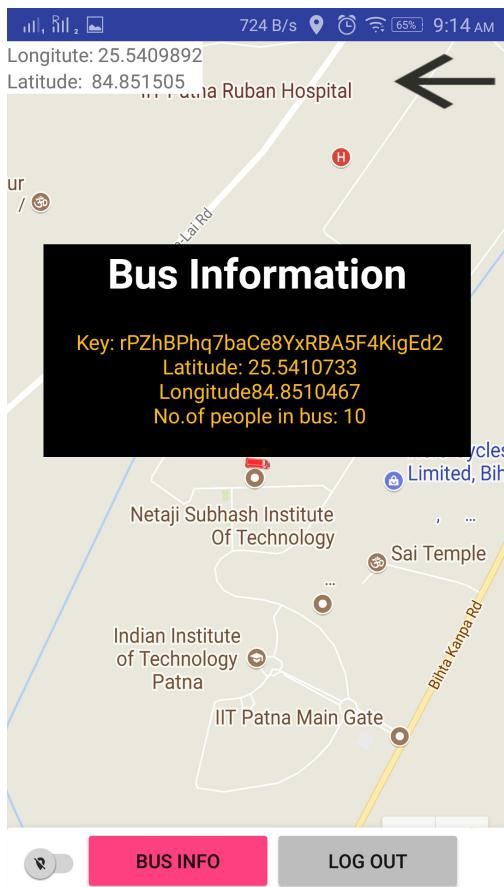


Fig. 4.6: Marker info window

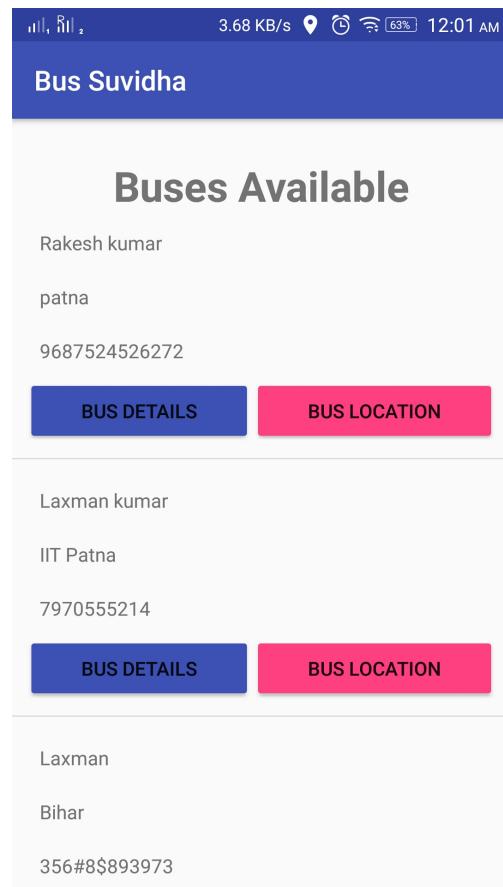


Fig. 4.7: Bus info of all buses

4.5 Server side implementation

Server side is important because it receives all the data which is coming from client (front-end) side. This data is stored on the server database for later use and data analysis. But main problem with implementation is how to connect device with server and fetch

the data from server. For implementation of this we need database management system (In this project I am using Real time online firebase database management system). I created and arranged all the tables according to our requirements. All the information in Firebase database save in the json format. Implementation of this part is easy but logic and understanding is important because it depends on all the fields of front-end side like - what type of input is present, how many fields of inputs are required for specific work like registration, login, real time, location, and data sending that's why this part is important.

4.6 DataBase management

Any project in which we work depends on data manipulation and storing data. Hence we need some storage methods and techniques for manipulation, for this we need to have a database. In this project we use Firebase database which are real time online database. This help us to manage all the data and tables which is used to develop a real time tracking device. For completing our purpose we need at initial phase two tables which are integrated with each other first is a Register table (which will store all information about registered gps device like username, password, bus number, route, bus stops in the route, timing etc) and second one is a location table (which stores latitude, longitude and saving time of current position of device). In current implementation, all registered gps device position records are stored in a single table, but for better performance and analysis of data we need one table for each registered device. This will be taken care of, when we scale the application.

4.7 Principle of Operation of counting

The basic principle of operation is based on simple analog beam balance approach to count the number of people getting into the bus or getting out of the bus. If a person is standing still, the force exerted on the ground by his foot will be closely equal. However, when

a person starts walking or climbing stairs, he lifts his one feet forward and lands it on the ground. As he lands his feet on the ground, the load on the front feet start increasing and the load on the rear feet start decreasing. A point comes when the load on both the feet are equal. At that point, we can count the number of people as incremented by one.

The sole purpose is to segregate human motion from luggage getting into the bus and give a relative correct count of the number of people in the bus.

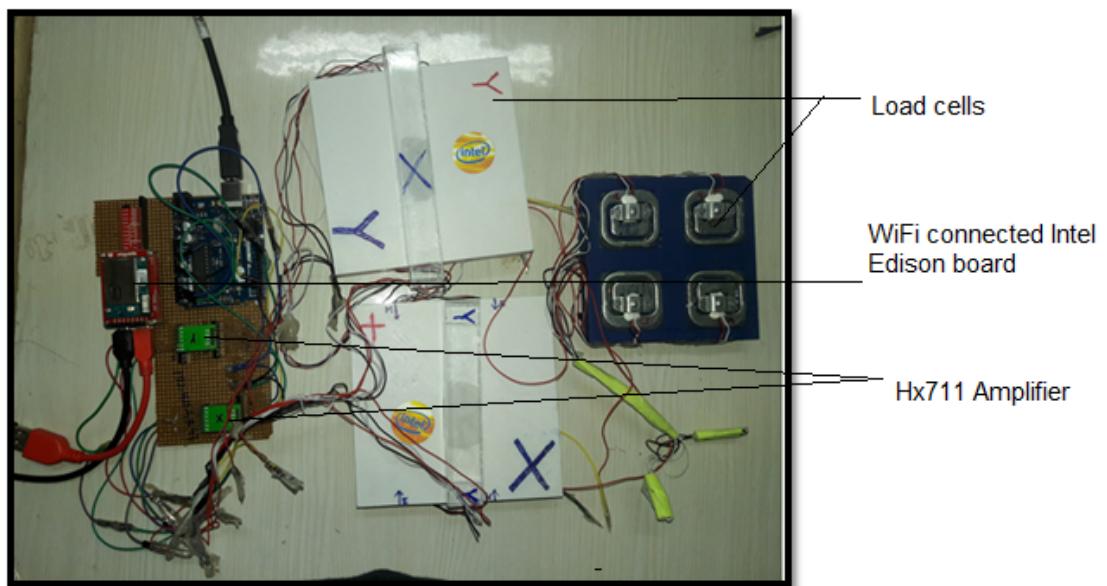


Fig. 4.8: Hardware setup

Chapter 5

Conclusion and Future Work

5.1 Future Work

To make the application more user friendly following, are the developments needed to be made in the application:

- Providing the application with facility to send emergency signals of accidents.
- Should show the hospitals, toilets, police stations in the route.
- Improved Android app will have higher utility in suburban towns and cities as well.
- Improve counting method in case of crowd in bus.
- Show a cluster map of waiting passengers to the bus driver, and transport officials.
- Analyse traffic information and route times from collected data for the transport route.
- Improve the features and UI of the mobile application.

5.2 Conclusion

Finally I was able to complete the mobile application which keeps a track of Registered buses and makes the data available to all the mobile users who have the app. Presently the application is workable & able to do the following stuffs:

- Application is able to show all the multiple buses in an area.
- Application is also capable of real time tracking of the active buses.
- User can get details about seat availability and route of active buses.
- Application is ready to be used and will be deployed.
- Hardware implementation is capable of counting passengers using load cells and validate using Microsoft Kinect.

Overall the project was a great learning experience for me. I became aware of several technologies which are currently used in this industry. I learnt how to work in deadlines, how to deliver what is asked of me, how to engage in meaningful discussions to gather knowledge, how to read documentation, how to learn a new technology quickly. I have now become more confident of taking up more projects and dont fear them at all.

References

- [1] Dr.(Mrs.) Saylee Gcharge, Manal Chhaya, Gaurav Chheda, Jitesh Deshpande, Niket Gajra, *Real Time Bus Monitoring System Using GPS*. Engineering Science and Technology: An International Journal (ESTIJ), ISSN: 2250-3498, Volume 2, Number 3, June 2012.
- [2] Manini Kumbhar, Meghana Survase, Pratibha Mastud, Avdhut Salunke, Shrinivas Sirdeshpande, *Real Time Web Based Bus Tracking System*. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 03, February 2016.
- [3] Gunjal Sunil N. , Joshi Ajinkya V. , Gosavi Swapnil C. , Kshirsagar Vyanktesh B, *Dynamic Bus Timetable Using GPS* International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) , ISSN:2278-1323, Volume 3, March 2014.
- [4] Harcharan Singh Ranu *Miniature load cells for the measurement of foot-ground reaction forces and centre of foot pressure during gait*. November 1985
- [5] Michael Grundy, Blackburn, P.A.Tosh, R.D.McLeish and L. Smidt, Manchester, England *An investigation of centre of pressure under the foot while walking*. November 1985.
- [6] Android studio documentation:
<https://developer.android.com/docs/>

[7] Real time online database:

<https://firebase.google.com/docs/>