REALTIME GPS TRACKER SYSTEM

BY

MOHSHIN AHAMED SHARIF SHAIBAL
ID: 161-15-7348
AND

MIM SAMI AL SHAHARIAR
ID: 161-15-7067

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Fahad Faisal
Assistant Professor
Department of CSE
Daffodil International University

Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2019

APPROVAL

This Project titled **"Realtime GPS Tracker"**, submitted by Mohshin Ahamed Sharif Shaibal, ID No: 161-15-7348 and Mim Sami Al Shahariar, ID No: 161-15-7067 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 05-12-2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain — **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Nazmun Nessa Moon — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Gazi Zahirul Islam — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Mohammad Shorif Uddin — **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Fahad Faisal, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Fahad Faisal
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Mohshin Ahamed Sharif Shaibal
ID: 161-15-7348
Department of CSE
DaffodilInternationalUniversity

Mim Sami Al Shahariar
ID: 161-15-7067
Department of CSE
DaffodilInternationalUniversity

©Daffodil International University ii

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Fahad Faisal**, **Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Embedded System"* to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain, professor & Head of Department of Computer Science and Engineering**, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

This project proposes and executes a cost and user friendly location based tracking system utilizing GPS and GPRS. The project system enables a client to watch over current and previous locations of the target object or device on Google Map through android application. The real time location of the device is pursued using Global Positioning System, the recorded information about the location is sent to a remote server using GPRS technology through POST method of HTTP protocol. That location information is saved in database for real time tracking and to preview past location records. Utilizing Java, xml, PHP, MySQL with implanted Google Map API an android app is created to set an user friendly environment for location viewing parsed from mentioned database. This project is exceptionally valuable for vehicle tracking, vehicle robbery situations, watching over young drivers by guardians (speeding and leaving particular region) and other scenarios human tracking.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
CHAPTER	
CHAPTER 1: Introduction	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	3
CHAPTER 2: Background	4-5
2.1 Introduction	4
2.2 Related Works and Technologies	4-4
2.2.1 GPS	4
2.2.2 GSM	4
2.3 Comparative Studies	5
2.4 Scope of Problems	5

2.5 Challenges	5
CHAPTER 3: Requirement Specification	6-18
3.1 Business Process Modeling	6
3.2 Requirement Collection and Analysis	7-15
3.2.1 Hardware Requirements	7-13
3.2.1.1 Arduino Uno R3	8
3.2.1.2 Ublox Neo 6m GPS Module	9
3.2.1.3 SimCom SIM 900A GSM Module	10
3.2.1.4 TP4056 Micro USB Charging Module	11
3.2.1.5 Li-ion Battery	12
3.2.1.6 Mini Solar Cell	13
3.2.2 Software Requirements	14-15
3.2.2.1 Arduino IDE	14
3.2.2.2 Android Studio IDE	15
3.2.2.3 Google Maps API	15
3.3 Use Case Modelling	16
3.4 Logical Data Model	17
3.5 Design Requirements	18
CHAPTER 4: Design Specification	19-26
4.1 Front–end Design	19
4.2 Back–end Design	20-22
4.2.1 GPS Data Processing	20

4.2.2 Working Structure of Arduino Code 21

4.2.3Working Structure of Java code 22

4.3 Interaction Design and UX 23-25

4.3.1 UX of Start Screen 23

4.2.2 UX of Live Tracker Screen 24

4.2.3 UX of Location History Screen 25

4.4 Implementation Requirements 26

CHAPTER 5: Implementation and Testing 27-

5.1 Implementation of Database 27-28

5.1.1 Database Table of "locData" 27

5.1.2 Database Table of "authentication" 28

5.2 Implementation of Front-end Design 29

5.3 Testing Implementation 30-34

5.3.1 Implementation of Live Tracking Screen 30-31

5.3.1 Implementation of Location History Screen 32-34

5.4 Test Results and Reports 35-36

5.4.1 Test Case Generation of Start Screen 35

5.4.2 Test Case Generation of Live Tracking Screen 35

5.4.3 Test Case Generation of Location History Screen 36

5.4.4 Test Case Generation of GPS Device 36

CHAPTER 6: Conclusion and Future Scope 37-37

6.1 Discussion and Conclusion 37

6.2 Scope of Future Development 37

APPENDIX 38-38

Appendix A 38

Appendix B 38

REFERENCES 39

List of Figures

FIGURES	PAGE NO
Figure 3.1: Business Process Modelling of Project	6
Figure 3.2: Arduino Uno R3	8
Figure 3.3: Ublox neo 6m GPS module	9
Figure 3.4: SIM 900A module	10
Figure 3.5: TP4506 micro USB charging module	11
Figure 3.6: 2500 mAh Li-ion battery	12
Figure 3.7: Mini solar cell	13
Figure 3.8: Arduino IDE interface	14
Figure 3.9: Android studio working IDE	15
Figure 3.10: Use case model for the system	16
Figure 3.11: Logical data model of the system	17
Figure 4.1: Device connection front-end setup.	19
Figure 4.2: Receiving GPS data and processing	20
Figure 4.3: Arduino code structured flow diagram	21
Figure 4.4: Flow diagram of android intents in app	22
Figure 4.5: Start screen UX	23
Figure 4.6: Live tracker screen UX	24
Figure 4.7: Location History Screen UX	25
Figure 4.8: Implantation of specified requirements	26
Figure 5.1: Database view of “locData” table	27
Figure 5.2: Database view of “authentication” table	28
Figure 5.3: Implantation of front-end of device	29
Figure 5.4: Implementation of tracking screen 1	30
Figure 5.5: Implementation of tracking screen 2	31
Figure 5.6: Location history screen 1	32

Figure 5.7: Location history screen 2 33
Figure 5.8: Location history screen 3 34

LIST OF TABLES

TABLES	s
Table 3.1: Hardware component list required for project	7
Table 5.1: Start screen test case	35
Table 5.2: Live tracking screen test case	35
Table 5.3: Location history screen test case	36
Table 5.4: GPS device test case	36

CHAPTER 1

Introduction

1.1 Introduction

A GPS positioning or tracking unit is such a mechanized gadget or device that used Global Positioning System to decide the exact area of a vehicle, individual or other resources with what it is planted to store the location data of that resource at a controlled time delay. This may be recorded within the gadget or transferred to a database using internet or cellular network or radio frequencies. [1] This permits any back-end tool to access that stored or received data from gadget and show visualization over map.

1.2 Motivation

Tracking devices are vastly used now-a-days for locating purposes. Rapid advancements in technologies has allowed regular normal persons to use this system. Knowing the whereabouts of anyone or thing has been a keen information in human life in lots of scenarios. As there are easy scopes of using this technology this system is very useful. Suppose, you own a car. You'd obviously try to cover its safety from theft or other hazardous occurrences. You can place the device under the dash or embed inside the car. So, if the car is ever stolen, you will be able track its position on the go, find its speed, travelling direction. This will be enough for police to catch the culprit. This applies for other moving vehicles also. Let's consider another scenario. If you have a small baby, and you are always busy for which you never have enough time to always be around him/her. In case that baby ever gets lost or abducted and in its bag the device was hidden, you can watch over where is the baby. Remember is this situation every single second counts. Else, let's think you have a teenage son or daughter. He/she is learning or have just started self drive. You would want to see if they are over speeding as it can turn to fatal accidents, or if they are wandering off from a certain area etc. If you have a transport company, with this system you can keep track of every assets or a delivery service, you can track down delivery progress and updates.

1.3 Objectives

This concept of a tracking gadget system is not new and there are loads of web and mobile software or apps which implements GPS usage. About all high-end cars have GPS system on board, all smartphones have installed GPS based map services. But this project is different from those. Because here the project is not focused on "what's the position of us?" but actually on "what is the position of him/her/it?" . Here the goal is locate a remote portable device attached to the asset. First objective is to collect coordinate data of location and check its validity at a regular interval. This is done with the help of microcontroller and GPS receiver. Second is to transmit that data to a secured remote server to store in a database. This goal is achieved by embedding a GPRS supported peripheral. And the final objective is to parse that continuously updatingdatasetand visualize over map on receiver of client. This is done on an android app.

1.4 Expected Outcome

As per objectives stated on previous sub chapter, the outcomes are also expected to be in stepwise. Actually three phases of outcome is expected here. The first one is the transmitter, the second one is a middleware which in this case is a data storage of a remote server and the third one is the client receiver. The transmitter will be of Arduino, GPS and GPRS related.[2] The working function will be to sense and send data. The middleware is a PHP and MySQL based database which receives formatted data, stores it according to the table presets and then whenever requested from authenticated user it will send data according to request. The client receiver android app will request data from database, visualize response.

Combining these three factors, the whole system will be functional.

1.5 Report Layout

Chapter 1: Introduction

Inside this chapter the system is introduced briefly. Questions regarding explanation of this system, reasons of implementing the system, what are the goals to be achieved and with which outcome goals are met are answered here.

Chapter 2: Background

We discussed background studies and related works needed here. Related technological introductions are also given. We showed the challenges and the problems related to this project.

Chapter 3: Requirement Specification

This chapter includes all specification of required hardware and software tools. Brief description about those components are also given.

Chapter 4: Design Specification

The aspects about design related works are given here. This includes the planning phase of design of the project. Back-end and front-end designs are specified here.

Chapter 5: Implementation and Testing

In this chapter we discussed about the interaction design and UX, implementation specified requirements and test analysis of the project.

Chapter 6: Conclusion and Future Scope

We summarized the project, discussed about scopes of further development and upgradable options.

CHAPTER 2

Background

2.1 Introduction

Global Positioning System, vastly known as GPS is the one complete Global Navigation System. This system utilizes 24 – 32 numbers of satellites which orbits around the globe 24/7. They send precise microwave signal. From at least three locked satellites, receiver can get latitude, longitude and kinetic measurements. And if there is another one, meaning four locked satellite then the value can be three dimensional. So, along with latitude and longitude, altitude can also be measured. This was actually developed by US Department of Defense and primarily was used for military use only. But later it became available for free in civilian usage.

2.2 Related Works and Technologies

2.2.1 GPS

GPS is a positioning system depending on the constellation of satellites launched in space developed by US. With this latitude, longitude, altitude and speed of receiver object can be detected.[3][4]

2.2.2 GSM

Global Systems for Mobile, this is a service for mobile phones. With a sim card using this technology a device is registered to certain operator network. Our phones work in this method. There are specialized modems which accepts sim cards to register into GSM network for embedded devices just like phones. With registration into this network GPRS is also accessible through which low bandwidth data connection can be created.

2.3 Comparative Studies

As mentioned before GPS technology is neither a new nor a non-frequently used tool. GSM is also used in the sector of sending sensor or reaction data/info over GSM or GPRS technology. To combine this we had to undergo basic working principle, functional detailed data about power consumption, delay analysis, connectivity status, overall signal strength quality of operators. Analyzing these a frame of plan was set and scoping of problems is set.

2.4 Scope of Problems

Embedding various components and setting up a system is never an easy task. There are some problems to be risen. First of all with the GPS module. To catch and get a lock on at least three satellite is the main key for getting a position update. The delay and accuracy depend on this process. Then comes GSM module. The device is for portable moving purpose. On movement the signal strength can dramatically get low. Also with low power, signal strength and finding operator will be hard.

2.5 Challenges

Implementing the system includes challenges regarding critical situations, new issues with algorithm implementation and how they are solved. The modules using GSM and GPS are continuously searching for their designated type of signal. With this power consumption can be high. As it is portable so, portable battery power won't be that much of effect unless backed up. If the power consumption is set to low then signal strength can be lower causing failure connecting to network. Like mentioned before, the updated location will be sent to server at a regular interval. When the device is located one place still, if that data is sent to base like normal time, this will drain battery and waste storage just for nothing.

CHAPTER 3

Requirement Specification

3.1 Business Process Modeling

Managing the project is not an easy task, so we divided the whole work process in various sub processes. Connections, analyzation of these processes are easily manageable.

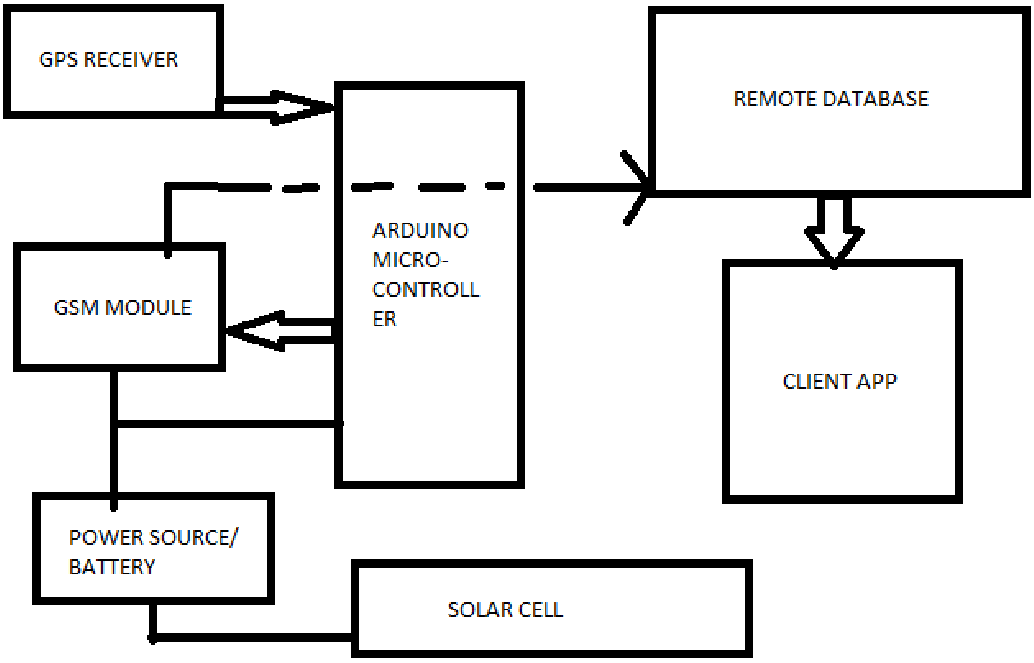


Figure 3.1: Business Process Modelling of Project

Building up this model did ease up the whole process and helped getting a clearer idea of processes combined.

3.2 Requirement Collection and Analysis

3.2.1 Hardware Requirements

Here the list of all required hardware components is given below.

Table 3.1: Hardware component list required for project

NO.	Component name	Dimension
01.	Arduino Uno R3	68.6*53.4mm
02.	Ublox Neo 6m GPS Module	23*30mm
03.	SimCom SIM900A GSM Module	24*24*3 mm
04.	TP4056 Micro USB 5v 1A Charging Module	27*17mm
05.	Li-ion Battery 2500 mAh	2.25*1.5 inches
06.	6v 1w Mini Solar Cell	110*60*2.5mm

Apart from these we will also need wires, containing box, welder etc. This list of components will serve the sole purpose of building up the whole project hardware system. Setting up this, then we can build and upload the software system that we need for the project. Details of each is given on next sections.

3.2.1.1 Arduino Uno R3

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits .A simple easy to use microcontroller. This is the most used board among all the Arduino boards. The ATmega328p is used here as the processor and its operating voltage is 5v. The recommended input voltage used here between 7-12V. The input voltage limit is between 6-20V.

Figure 3.2: Arduino Uno R3[5]

14 digital I/O pins are used here of which there are 6 provide PWM output. There are 6 analog input pins and DC current per I/O pins are recommended 20 mA. For the DC current of 3.3V pins are 50mA. There are Flash memory, SRAM and EEPROM inside it. The Flash memory has a storage of 32 KB, SRAM has a storage of 2 KB and the EEPROM has a storage of 1 KB. It has length of 68.6 mm and a width of 53.4 mm and the weight is 25 g.

3.2.1.2 Ublox Neo 6m GPS Module

The NEO-6M GPS module is used to get GPS data with the help of Arduino UNO. The global positioning system is used to determine the position, time and speed. We get raw data from the module. The NMEA encoded data is collected from the module and it shows the speed, latitude, longitude of the object that we are considering.

Figure 3.3: Ublox neo 6m GPS module[6]

We use four pins here. First of all the VCC that can be operated by 3.3V but the recommendation is to use 5V.Then the RX is connected to pin no 3 and then TX is connected to pin no 4. The GND pin is connected with the GND. There are an internal battery in the module and also sometimes a backup battery is used to operate the module more efficiently. When we get raw data from the module we use Software Serial to establish the connection. The Neo-6M module has a baud rate of 9600 bps. The NMEA encoded data always starts with the $ sign and we separate it using the comma sign. $GPGGA is a basic message from the encoded data that provides 3D position and the data accuracy.

3.2.1.3 SimCom SIM900A GSM Module

The SIM900A GSM module is very effective for wireless communications and it use a dual band GSM/GPRS where it performs 900/1800 MHz and it can transmit SMS or voice data. It’s efficient because it can operate in a low power mode which saves energy.

Figure 3.4: SIM 900A module[7]

When it runs on class 4 then it use 2W @900 MHz and when class 1 is used then it use 1W @1800MHz .The voltage range vary from 3.1-4.8V. When it is in sleep mode it uses 1.5mA. It can operate in -40°C and the highest is +85 °C. Also an analog audio interface is used here. One antenna is used here for the connectivity.

3.2.1.4 TP4056 Micro USB 5v 1A Charging Module

The TP4056 micro USB has a female micro USB port that can be connected with a device that has a male micro USB port. The input voltage used here is 5V.The type of the battery is Lithium. It uses a Micro USB for connecting the breakout board to any computer or USB wall adapter. It works with linear charging method.The default resistor soldered in on the board is 1.2K Ohm.

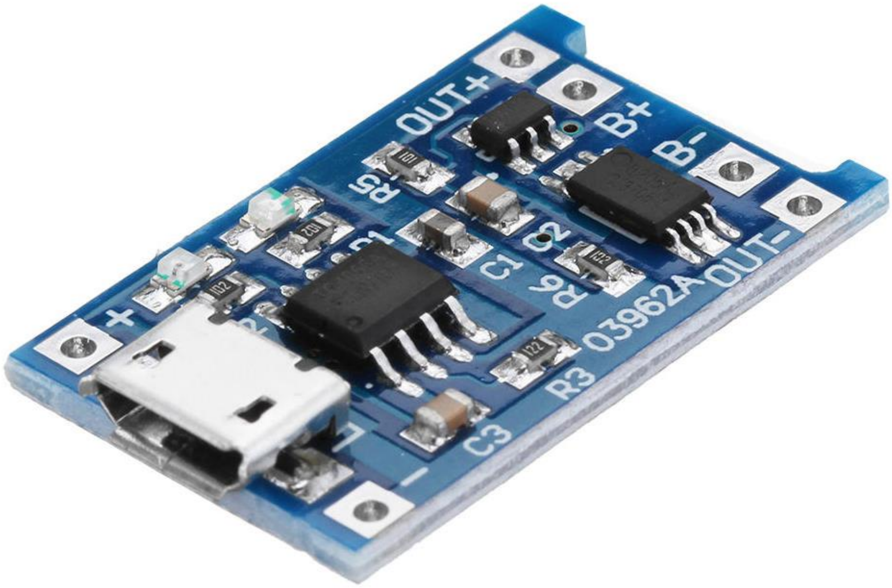


Figure 3.5: TP4506 micro USB charging module[8]

It has maximum charging current capacity if 1000 mA and its charge cut-off voltage is 4.2V +/-1. Here the battery over-current as protection current is 3A. All the input interface used here is micro USB. And its size is 27*17mm. This module will be used to power and simultaneously charge the battery or alone.

3.2.1.5 Li-ion Battery 3500 mAh

We used Walton smartphone battery here because it’s smaller in size and also very cheap. It’s a lithium-ion battery that’s why it has a longer battery life. For charging it uses 5V charger and its normal operating voltage is 3.7V. The charger here also can be used 2A for a fast charging experience.

Figure 3.6: 2500 mAh Li-ion battery

The dimension of the battery is 2.25*1.5 inches.Its capacity is 3500 mAh. The battery is also very suitable for the smartphones because it has a long charging and discharging limits. The battery on the other hand is efficient because it automatically stops charging when it is fully charged.

3.2.1.6 6v 1w Mini Solar Cell

We used 6V 1W mini solar cell here to power up the battery. The solar panel can charge up the battery in a lower sunlight. This has a higher conversion rate and also high efficiency output. The working voltage is 6sV here and the output is 1W. The working current range is 0-200 mA.

Figure 3.7: Mini solar cell

The dimension of the product is 110*60*2.5mm. It's also used in home lighting and for DIY solar power toys. In our system when the battery is low the solar can charge up the battery in daylight.

3.2.2 Software Requirements

3.2.2.1 Arduino IDE

It is an open source tool to compile codes into Arduino boards and debug if needed. It is very easy to learn and easy to use software. It supports numerous types of board including Uno R3. This also can be installed in all versions of Windows, Mac and Linux based computers able to run on java based platforms. The environment supports both C and C++ codes.

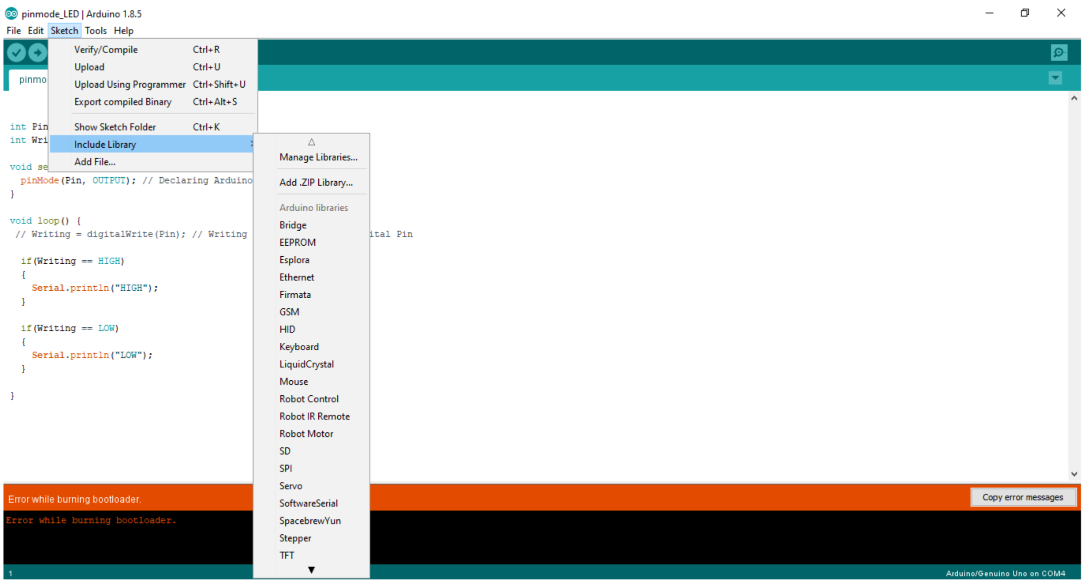


Figure 3.8: Arduino IDE interface

The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop. That code is uploaded via various means to the Arduino board’s memory. Uploading the code, the serial monitor of the ide can also be used to transmit and receive data to and from the target board through selected port.

3.2.2.2 Android Studio IDE

Android studio is the official IDE of Google Android Inc. This is the tool for creating the android app. It is very useful in building, look for errors, debugging android apps. The UX setups and back-end supports can be done in this environment. Kotlin and Java can be used as source language. We can export or import the project files here. We can also build apks for supported android version with debug or release credentials.

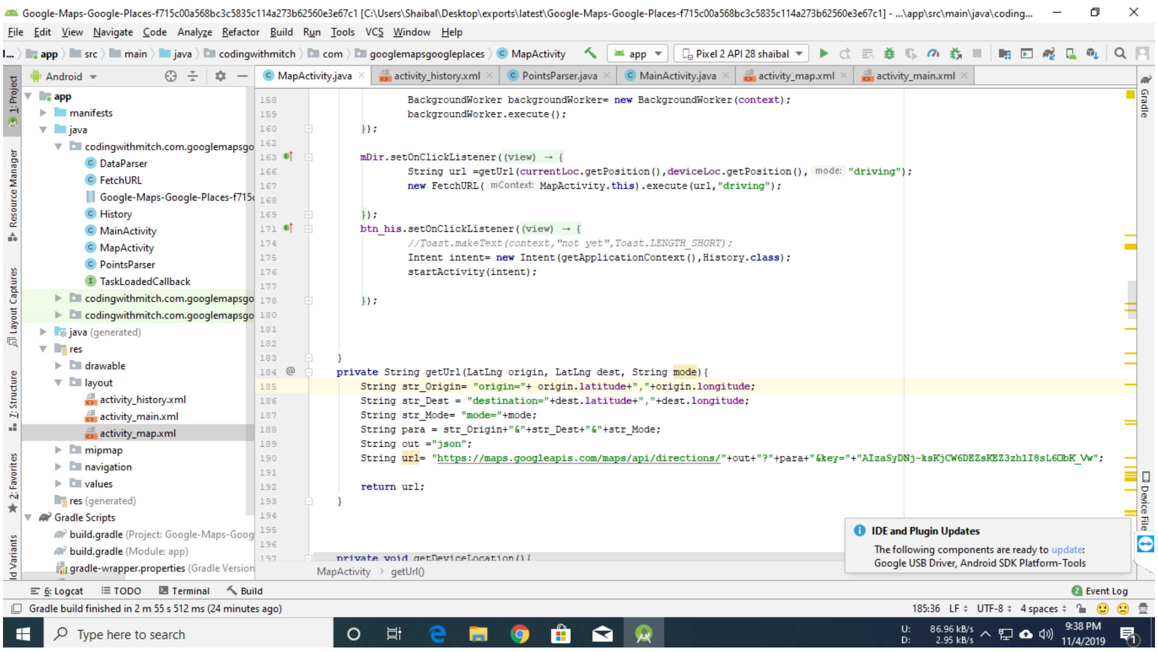


Figure 3.9: Android studio working IDE

3.2.2.3 Google Maps API

To use map view in our android app, we need to have some kind of framework to define the map, markers, routes etc. For that we used Google Maps API provided free of cost for limited requests to developers. We generated a unique key for API and added to the manifest file of the project. Linking with that, the map fragment works fine showing the details in map and markers set.

3.3 Use Case Modeling

A use case model generally describes how different users interact with the system for different purposes. It shows the goals of particular users, mutual interactions between system and clients, and needed response of the process to achieve those specified goals. The use case model of our system is given below.

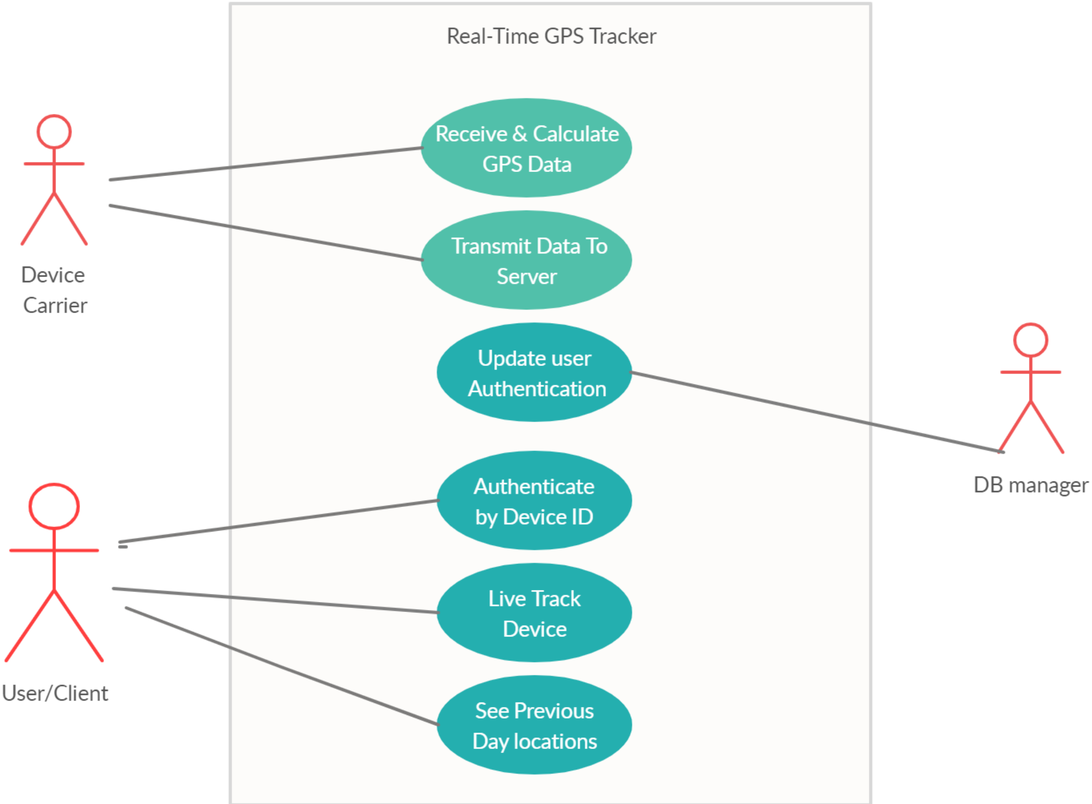


Figure 3.10: Use case model for the system

3.4 Logical Data Model

This model shows in which manner the data is structured in the system. This includes all the data entities in the set and their relationships with user and each other. Abstraction and inheritances are also shown. The data needed for our project system is given in a model below.

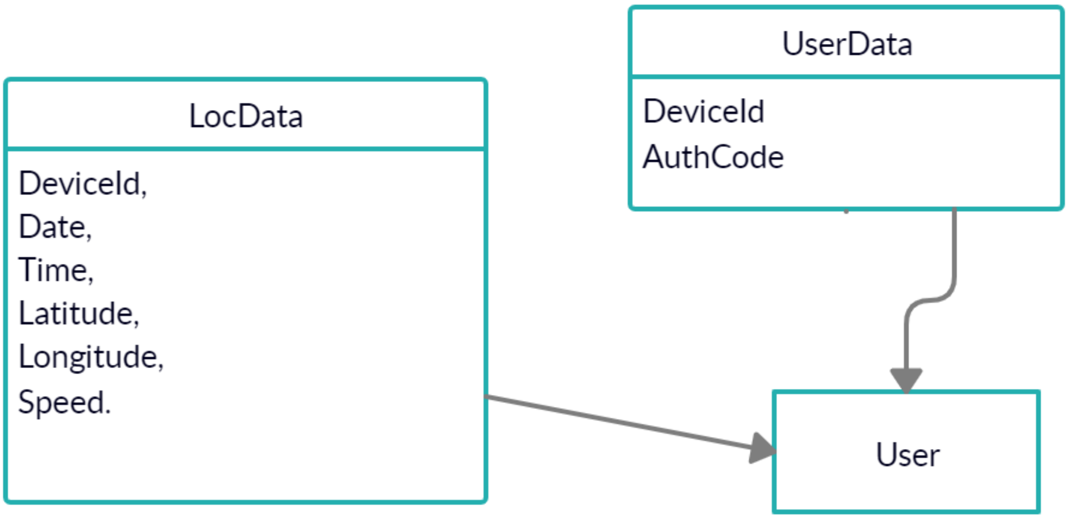


Figure 3.11: Logical data model of the system.

Location related attributes consist LocData entity. All these are private to unique field deviceId. This is accessed by User. DeviceId and authCode includes unique combination of verification of user.

3.5 Design Requirements

Though target user is just client type but actually there will be two types of users of the system. Except client Database Manager will be active in the system. This manager will overview the device id with unique authentication code assignments. The overall security lies in these two number combination.

- Client user can log in through device id.
- Client user can track live location of device if online.
- Client user can see past data selecting date, if available.
- Manager can edit, update, delete user data according to request,
- Manager can't see Location Data protecting privacy of user.

CHAPTER 4

Design Specification

4.1 Front-end Design

Setting up the collected required hardware will complete in front-end setup of device. Here, through breadboard we have connected needed connections. The Arduino Uno is connected to battery with Vin and GND. The charging cables of battery is connected to charging module and in points of module with solar panel. Rx and Tx pins of GPS module and SIM module is connected to Arduino digital pins. The power pins of those are connected to appropriate source.

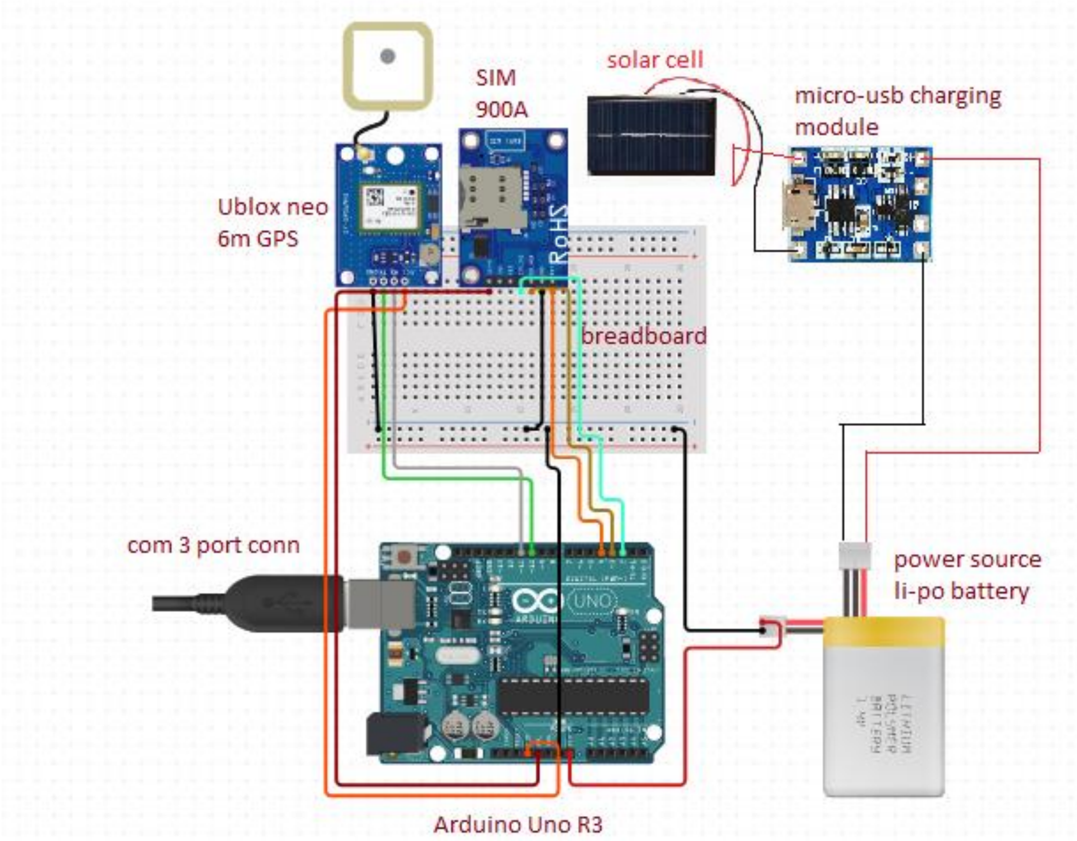


Figure 4.1: Device connection front-end setup.

4.2 Back-end Design

4.2.1 GPS data processing

The GPS module gives new positions in certain time delay. First the system listens for new positions. Then checks validity. As the data is in NMEA format, it coverts that to decimal Latitude and Longitude values.[9] Then stores the data and forwards to necessary process.

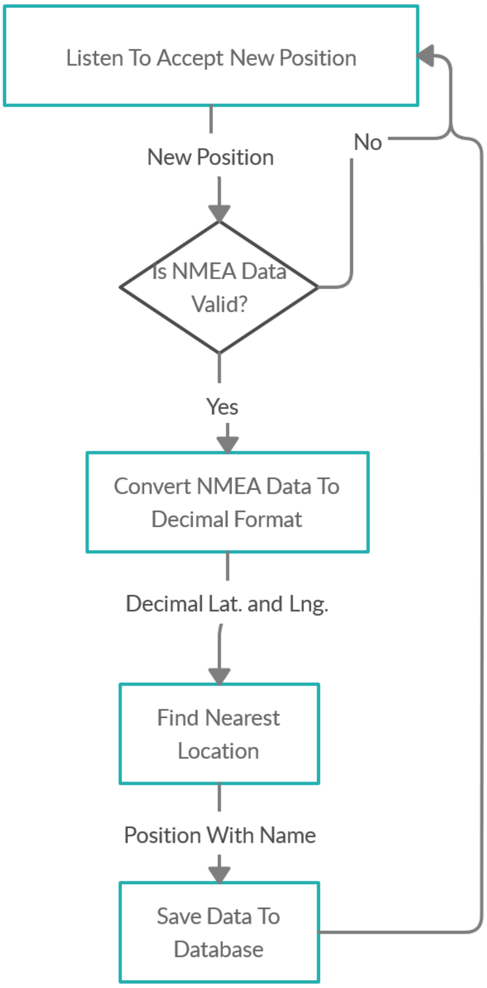


Figure 4.2: Receiving GPS data and processing.

4.2.2 Working structure of Arduino code used in project

Here is a brief graphical representation of how the system in device works in organized flow one after one process. The working combination of Arduino Uno, GPS module and SIM module is given in a diagram below.

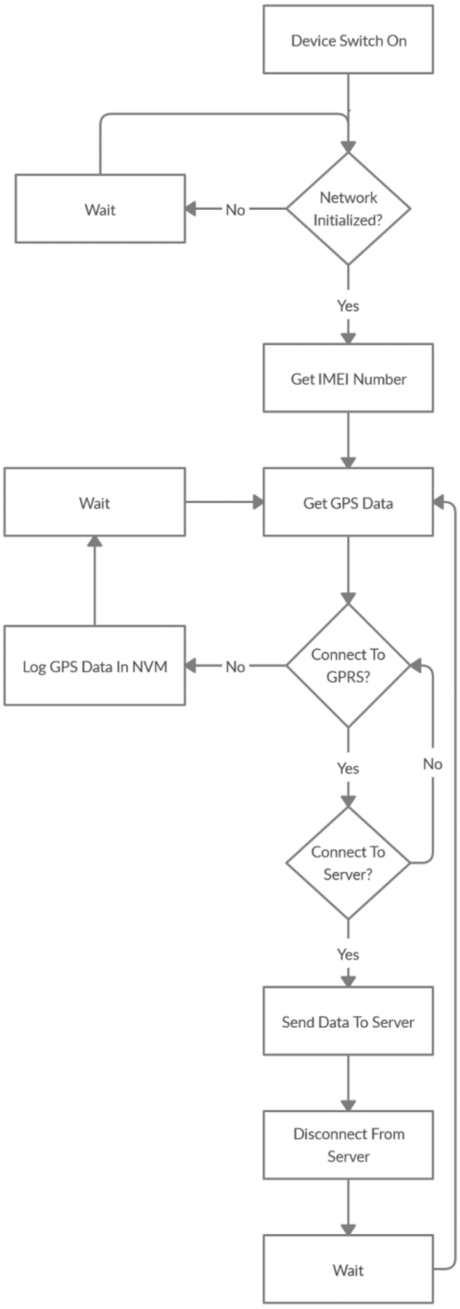


Figure 4.3: Arduino code structured flow diagram

4.2.3 Working structure of Java code used in project app

For the app we used Java language to code. There are in total six Java classes for three intents we used. In those classes we used different methods to do some certain works. These works are stated in a diagram below through flow connections.

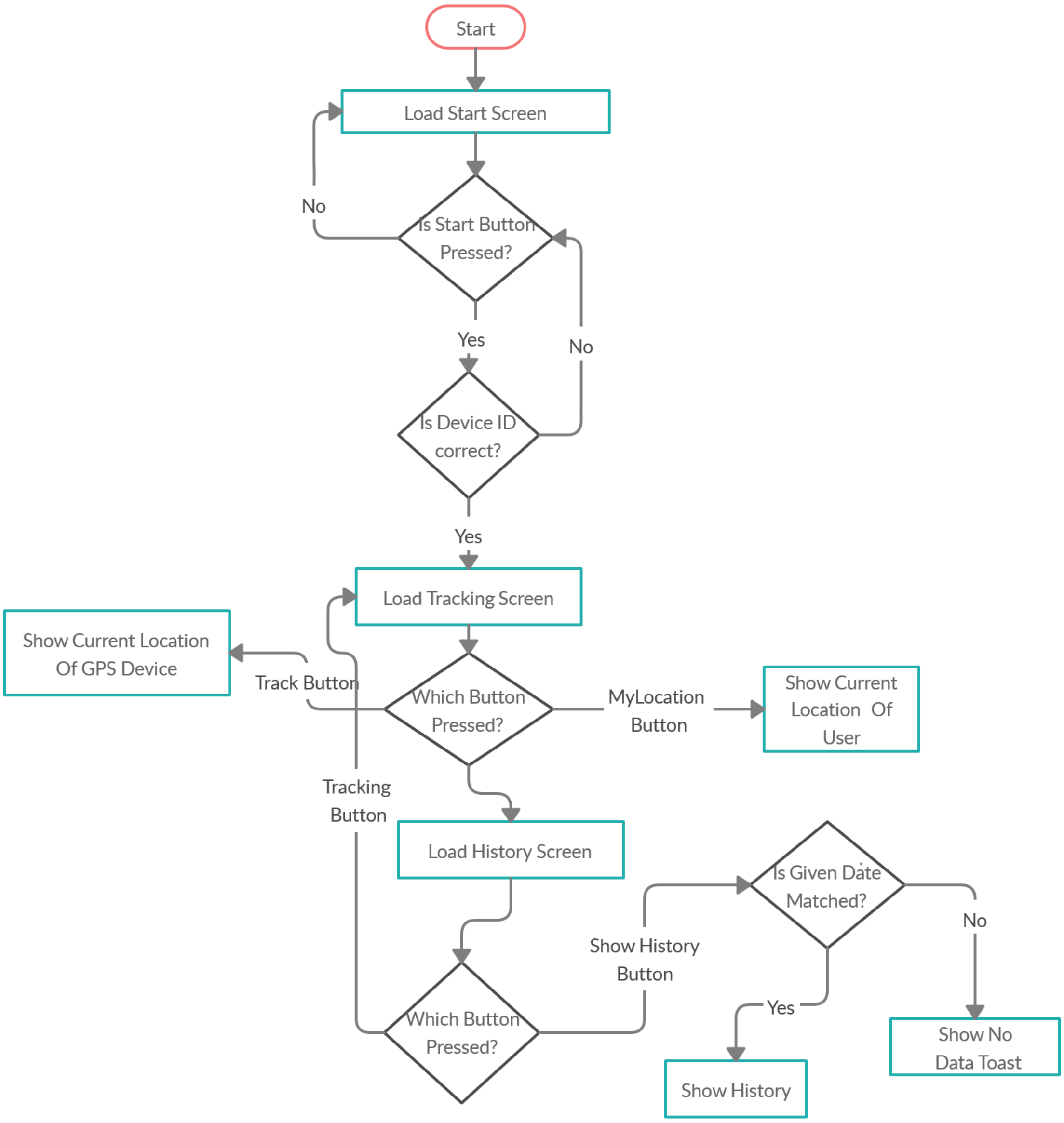


Figure 4.4: Flow diagram of android intents in app

4.3 Interaction Design and UX

4.3.1 UX of Start Screen

This consists of three main sections. The logo of the app is on the top. Below that there is a textbox to take input from the user for Device ID. There is also a button to submit the input to the system.

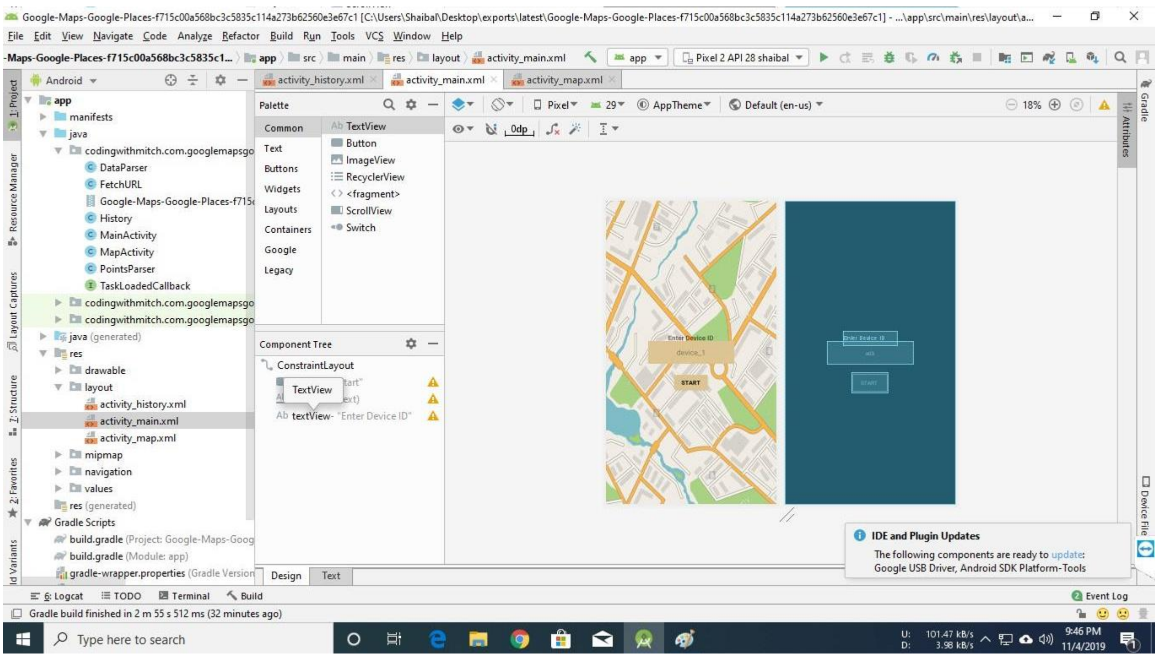


Figure 4.5: Start screen UX

This was created using android studio in layout xml. All the components in the layout screen is on a constrain layout and the components are constrained to the main layout's boundaries. So on screen size difference meaning of use in different device will make no change in placements.[10]

4.3.2 UX of Live Tracker Screen

This screen has all total three buttons, and one text view, not to mention the map fragment. The buttons are “Show Device Location” button, “Show User Location” button and “Go To Location History” button. The text box shows battery status value.

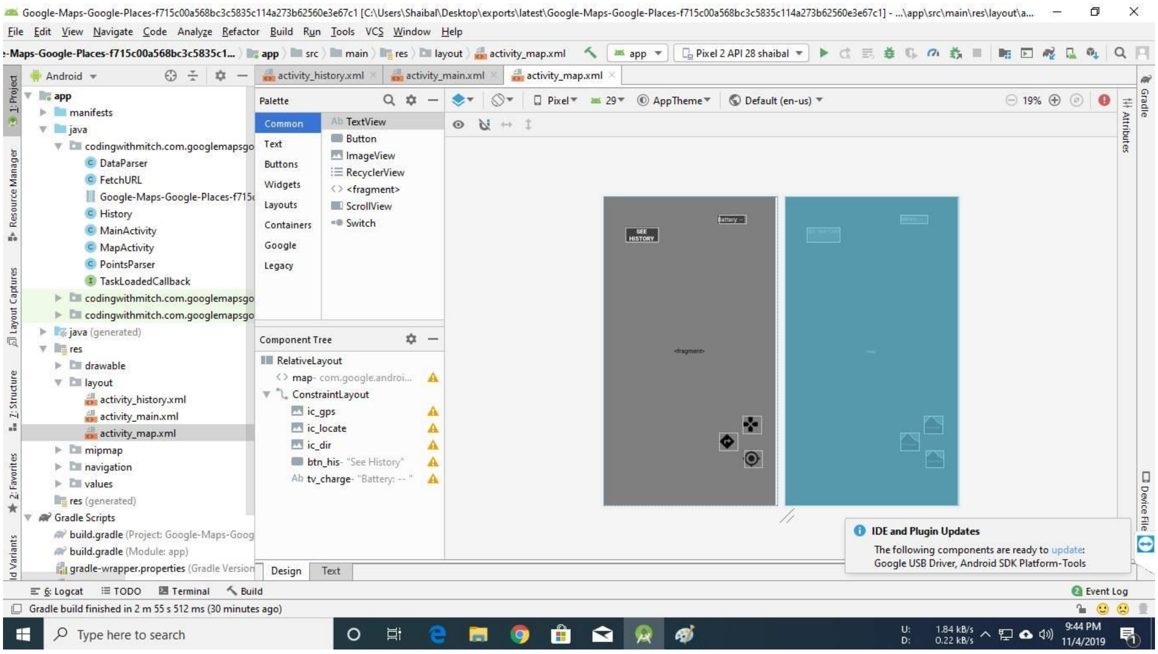


Figure 4.6: Live tracker screen UX

The components are constrained to the border of a constraint layout. This makes the screen to adjust with variations in different screen resolution and pixel count. The button views and text vies are set according to user friendliness and required task of the UI. The image properties are compressed to save memory space.

4.3.3 UX of Location History Screen

This has a map fragment in total screen layout. Over that there are two buttons and one text field. One button is for going to back screen and the other is for submitting the date extracted from text field.

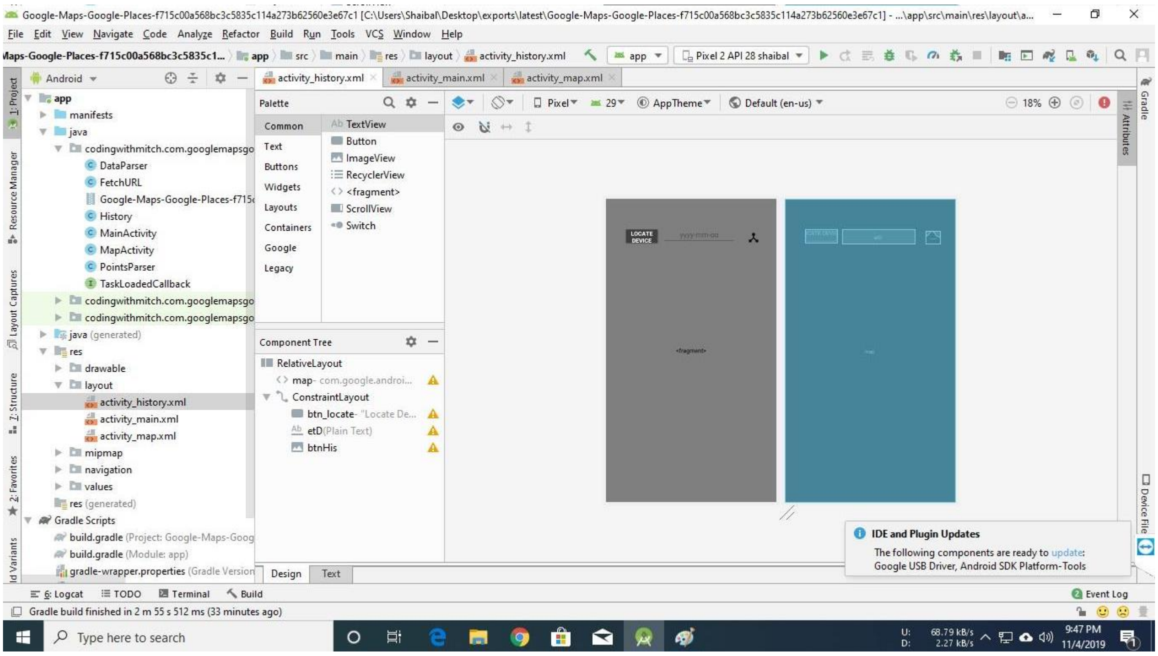


Figure 4.7: Location History Screen UX

The required operation of the UI is going to take large space on the screen of the app. For that purpose, required space is left below the linearly set text views and button. The constraint setup of the components used makes the app to set views indifferent to devices with various resolutions.

4.4 Implementation Requirements

The implementation includes the flow connections of all required components in a structured manner. Microwave signal transmitted from satellites are received and calculated for processing in GPS module and saved in memory. [11]

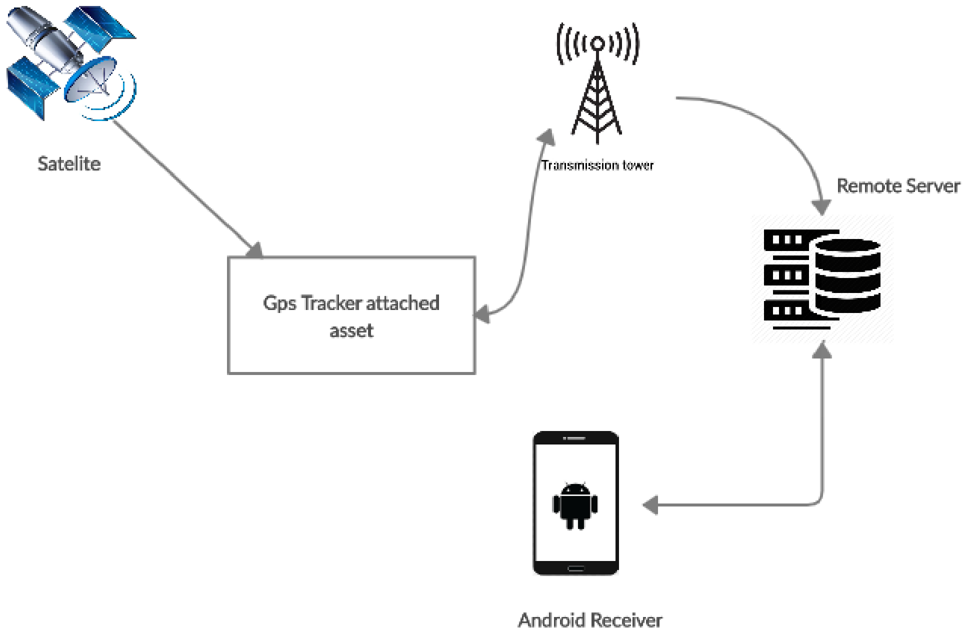


Figure 4.8: Implantation of specified requirements.

Connecting networks with SIM module and transmitting GSM tower will establish network to transmit data. Transmitted data is stored in remote database after successful receiving. From that on request the data is loaded to android app screen. The app will have location data and device power stats in background when requested. After that on request and preset of different views and operations these data will be used in different ways.

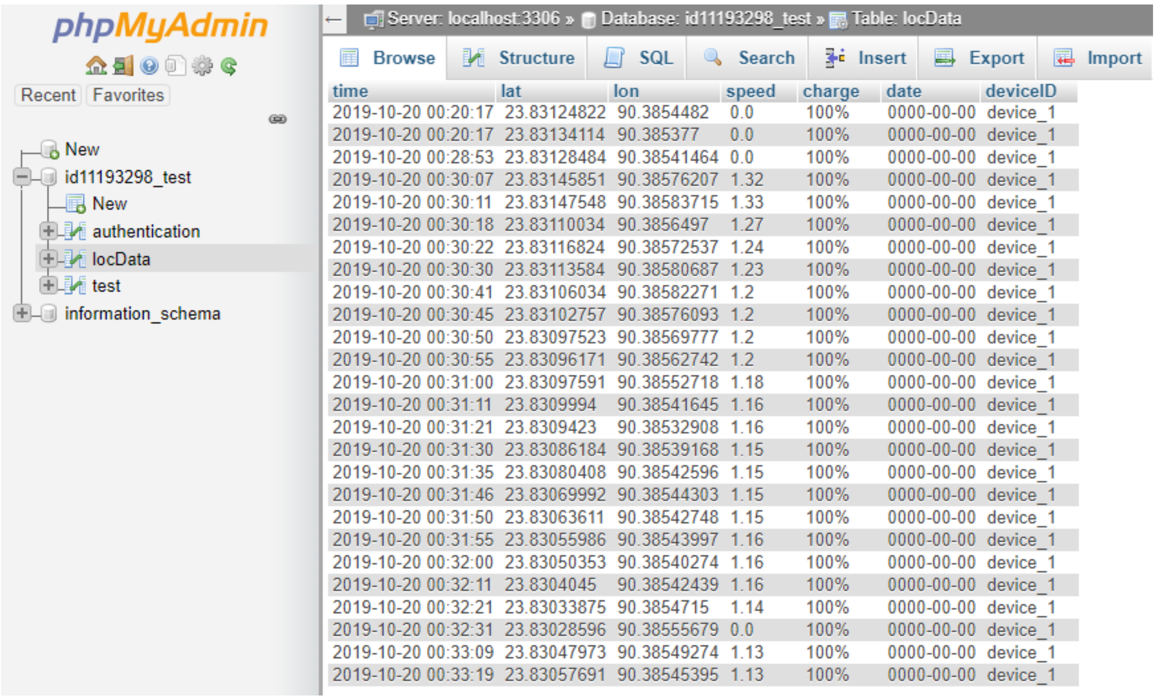
CHAPTER 5

Implementation and Testing

5.1 Implementation of Database

5.1.1 Database Table of "locData"

We used MySQL database for the data management of this project. First table is "locData". This holds data sent from device in data fields. The data fields are timestamp, latitude, longitude, speed, charge, date and device id.



time	lat	lon	speed	charge	date	deviceID
2019-10-20 00:20:17	23.83124822	90.3854482	0.0	100%	0000-00-00	device_1
2019-10-20 00:20:17	23.83134114	90.385377	0.0	100%	0000-00-00	device_1
2019-10-20 00:28:53	23.83128484	90.38541464	0.0	100%	0000-00-00	device_1
2019-10-20 00:30:07	23.83145851	90.38576207	1.32	100%	0000-00-00	device_1
2019-10-20 00:30:11	23.83147548	90.38583715	1.33	100%	0000-00-00	device_1
2019-10-20 00:30:18	23.83110034	90.3856497	1.27	100%	0000-00-00	device_1
2019-10-20 00:30:22	23.83116824	90.38572537	1.24	100%	0000-00-00	device_1
2019-10-20 00:30:30	23.83113584	90.38580687	1.23	100%	0000-00-00	device_1
2019-10-20 00:30:41	23.83106034	90.38582271	1.2	100%	0000-00-00	device_1
2019-10-20 00:30:45	23.83102757	90.38576093	1.2	100%	0000-00-00	device_1
2019-10-20 00:30:50	23.83097523	90.38569777	1.2	100%	0000-00-00	device_1
2019-10-20 00:30:55	23.83096171	90.38562742	1.2	100%	0000-00-00	device_1
2019-10-20 00:31:00	23.83097591	90.38552718	1.18	100%	0000-00-00	device_1
2019-10-20 00:31:11	23.8309994	90.38541645	1.16	100%	0000-00-00	device_1
2019-10-20 00:31:21	23.8309423	90.38532908	1.16	100%	0000-00-00	device_1
2019-10-20 00:31:30	23.83086184	90.38539168	1.15	100%	0000-00-00	device_1
2019-10-20 00:31:35	23.83080408	90.38542596	1.15	100%	0000-00-00	device_1
2019-10-20 00:31:46	23.83069992	90.38544303	1.15	100%	0000-00-00	device_1
2019-10-20 00:31:50	23.83063611	90.38542748	1.15	100%	0000-00-00	device_1
2019-10-20 00:31:55	23.83055986	90.38543997	1.16	100%	0000-00-00	device_1
2019-10-20 00:32:00	23.83050353	90.38540274	1.16	100%	0000-00-00	device_1
2019-10-20 00:32:11	23.8304045	90.38542439	1.16	100%	0000-00-00	device_1
2019-10-20 00:32:21	23.83033875	90.3854715	1.14	100%	0000-00-00	device_1
2019-10-20 00:32:31	23.83028596	90.38555679	0.0	100%	0000-00-00	device_1
2019-10-20 00:33:09	23.83047973	90.38549274	1.13	100%	0000-00-00	device_1
2019-10-20 00:33:19	23.83057691	90.38545395	1.13	100%	0000-00-00	device_1

Figure 5.1: Database view of "locData" table.

The table has seven columns of required different attributes and rows of insertion according to column. If somehow in a dataset one or multiple data is missing null value will be inserted and handled for data extraction.

5.1.2 Database Table of “authentication”

This table named “authentication” stores user data for admin view. User login credentials are here. The combination of “deviceId” and “authCode” value is unique.

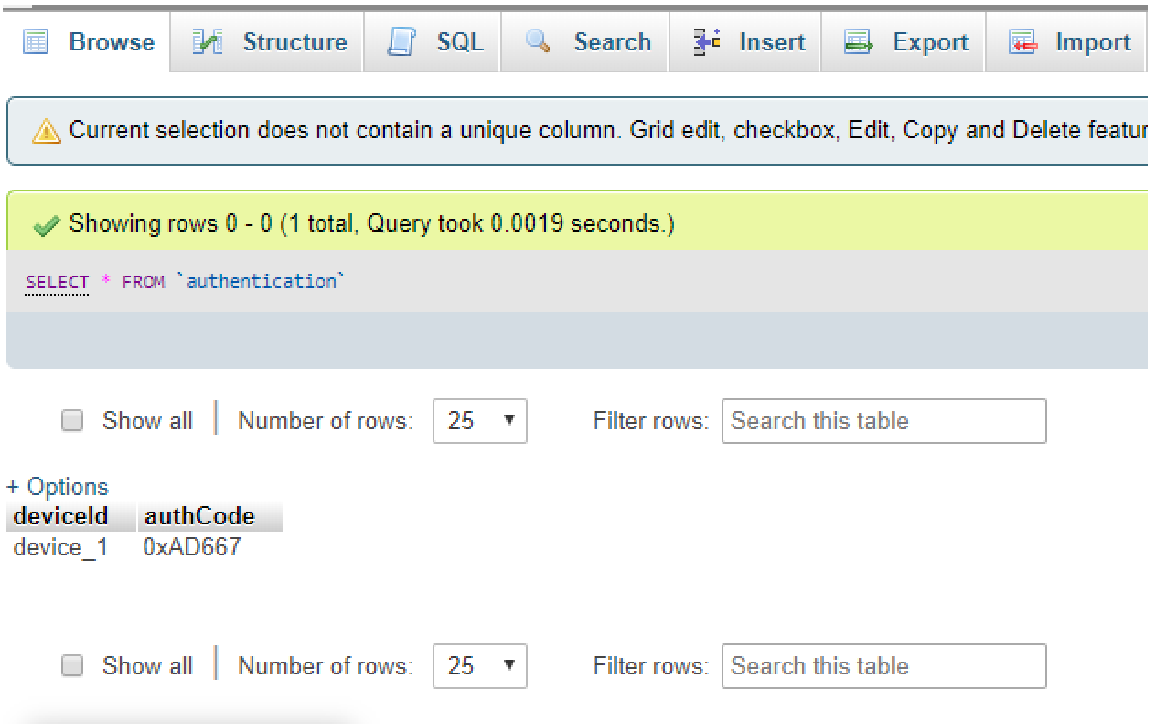


Figure 5.2: Database view of “authentication” table.

This table has two columns only. The authCode field will hold data which might be sensitive to user. For that encryption method is set up for this field only.[12] This will make the admin having access to this table impossible to misuse that authority.

5.2 Implementation of Front-end Design

As stated in the design section the hardware components are connected with one another. The battery is charged before using. Each component was tested individually before connecting to look for any kind of error which would be hard to find if not done before. After implementation we check for data insertion in database. If there is data in expected time delay, in our case 12 seconds a data, the system is working.

Figure 5.3: Implantation of front-end of device

5.3 Testing Implementations

5.3.1 Implementation of Live Tracking Screen

This is our second screen in app. Here we show the current location of user and current location of the attached GPS device through two buttons. If you notice on the top right corner you can see current battery status of the device also.

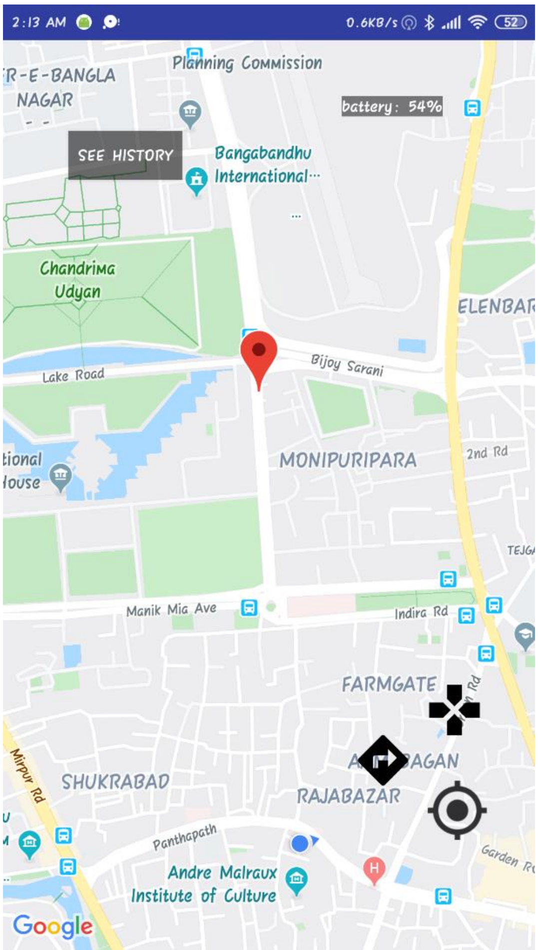


Figure 5.4: Implementation of tracking screen 1

As per stated in the design section the buttons are not obstructing the views for the preset opacity value and size.

In the second figure we can see updated location. The previous screenshot is at 2.13 am. Whereas this is at 2.14 am. In this time the marker has moved to updated location in real time. On time change as battery state changed, that is also on the changed screen.

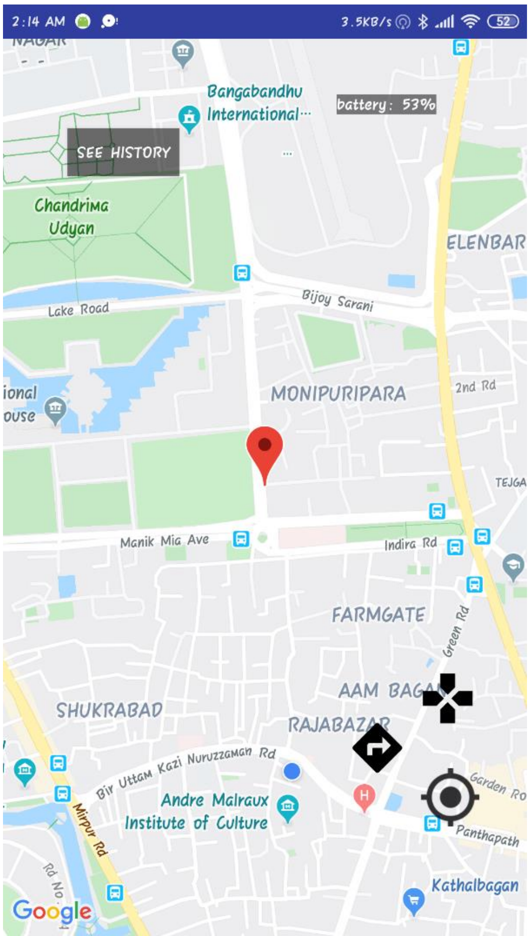


Figure 5.5: Implementation of tracking screen 2

5.3.2 Implementation of Location History Screen

On this screen, we show data of previously travelled location points if available. The data is parsed from database table according to the date given in the textbox. Here is a screenshot of history on 2019-11-02.

Figure 5.6: Location history screen 1

The operation of this screen needed space. So we can see that enough space for the marker is left to set under the text and button section.

Here is a screenshot of location history of 2019-11-01. This screen is exactly like previous with a change of input data and marker setups of the locations.

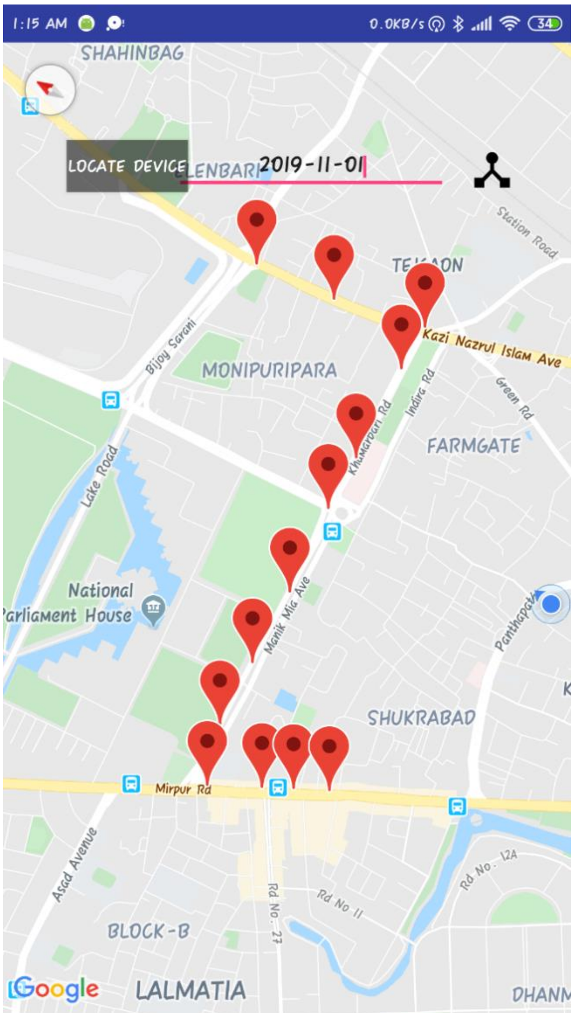


Figure 5.7: Location history screen 2

As the date given in the text box is of year 2017, there is no data for that year in database. So there is no marker in the screen instead a toast message is shown with “No Data” text.

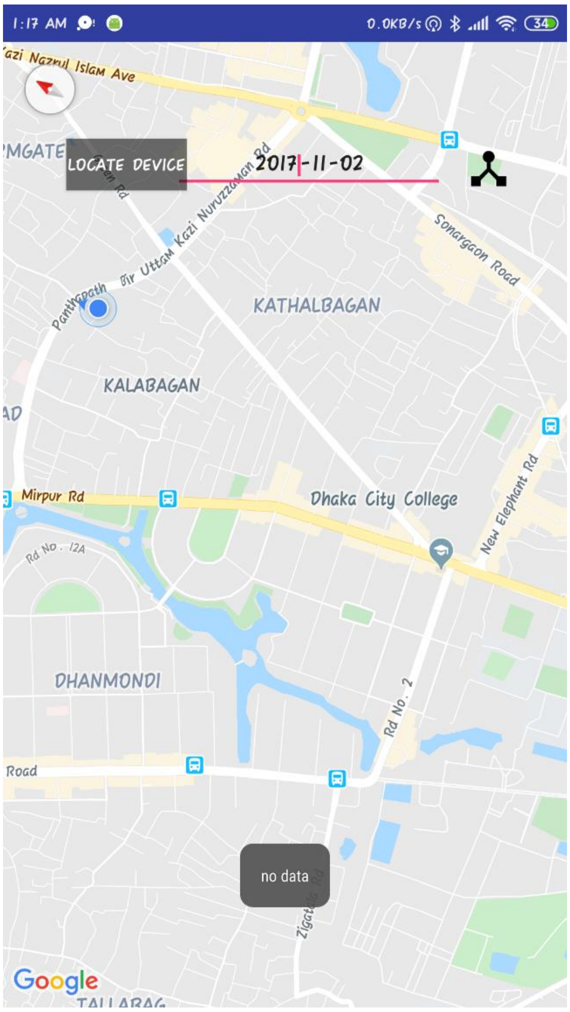


Figure 5.8: Location history screen 3

The toast message can be seen for a time not to bore the user also for not missing the output message.

5.4 Test Results and Report

5.4.1 Test Case Generation of Start Screen

On “Start Screen”, there are a text field, and a button. Test cases from those are given here.

Table 5.1: Start screen test case

Test Case No.	Action	Inputs	Expected Outcome	Actual Outcome	Test Result
#1	Login	No Input	Message: Field Empty	Message: Field Empty	Pass
#2	Login	Wrong Input	Message: Invalid Device Id	Message: Invalid Device Id	Pass
#3	Login	Right Input	Start Map Intent	Start Map Intent	Pass

5.4.2 Test Case Generation of Live Tracking Screen

On “Live Tracking Screen” there are three buttons. Test result of those are given here.

Table 5.2: Live tracking screen test case

Test Case No.	Action	Inputs	Expected Outcome	Actual Outcome	Test Result
#1	Button Press	My location button pressed	Show My Location	Show My Location	Pass
#2	Button Press	Device location button pressed	Show Device Location	Show Device Location	Pass
#3	Button Press	History button pressed	Go to History Intent	Go to History Intent	Pass

5.4.3 Test Case Generation of Location History Screen

"History Intent" has a button and an input field. Test case and result generated from scenario is given below.

Table 5.3: Location history screen test case

Test Case No.	Action	Inputs	Expected Outcome	Actual Outcome	Test Result
#1	Button press	Locate button pressed	Go to home screen	Go to home screen	Pass
#2	See History	Wrong Input	Message: No Data	Message: No Data	Pass
#3	See History	Right Input	Show History	Show History	Pass
#4	See History	No Input	Message: No Input	Message: No Input	Pass

5.4.4 Test Case Generation of GPS Device

Test cases from device operation are like stated here in below table.

Table 5.4: GPS device test case

Test Case No.	Status	Expected Outcome	Actual Outcome	Test Result
#1	Device Switched Off	No Update	No Update	Pass
#2	Device On, Network Connected	New Location Update	New Location Update	Pass
#3	Device On, No Network Connected	No Update	No Update	Pass

CHAPTER 6

Conclusion and Future Scope

6.1 Discussion and Conclusion

In this project we developed a GPS tracking system and which can take additional power from a solar panel. GPS stands for Global Positioning System which is invented for the US military operations. But for its so many advantages it's spreading worldwide. Now a days GPS tracker is used everywhere. We can use it in our vehicles to track the position of the vehicle. We don't have to worry about the vehicle because we can find the location of the vehicle sitting anywhere. We can also get back our stolen vehicle by using our real time positioning system. Our developed app is easy to use and can access from any ISP data. It can also be used for wildlife tracking, asset tracking or many more. It's cheap and easy to use that's why many people rely on the tracking system.

6.2 Scope of Future Development

In future we can setup some sensors to track the speed of the object in precise value connected with the device. As the tracker works by directly taking data from satellite we can get more convenient data set by using sensors.[13]

We can use integrated microphones for getting voice data.[14] As we used a sim module inside the tracker then it can be possible to transmit voice data from the device and we can also store data inside a built in memory chip.

By using sensor we can stop the engine of the vehicle. If we think the vehicle is going to an unauthorized route or the driver is doing something suspicious we can stop the engine of the vehicle from anywhere by the help of sensors and the sim module.

References

[1] Integration of GPS and GSM for Determination of cellular coverage area by A. D. Sarma, P. S. Ravikanth and D. Krishna Reddy.

[2] Multi-Tracking System using GPS and GSM” by Sameer Darekar, Atul Chikane, Prof. Archana Shinde from Proceedings of International Conference on ACCT-2012.

[3]For working of GPS Devices and Network, available at <<www.casde.iitb.ac.in/IMSL/motion/gps/Seminar%20on%20GPS.ppt>>, last accessed on 2019-11-02 at 3.00 pm.

[4] How GPS receiver works, available <<http://electronics.howstuffworks.com/gadgets/travel/gps.html>>, last accessed on 2019-11-02 at 3.00 pm.

[5] Arduino Uno rev3, available at <<https://store.arduino.cc/usa/arduino-uno-rev3>>, last accessed on 2019-11-02 at 3.00 pm.

[6] Neo 6m GPS, available at <<https://potentiallabs.com/cart/buy-ublox-neo-6m-gps-module-online-india>>, last accessed on 2019-11-02 at 3.00 pm.

[7] SimCom GSM SIM 900A module <<https://www.w11stop.com/simcom-gsm-sim-900a-module>>, last accessed on 2019-11-02 at 3.00 pm.

[8]Public release Version, “NAVSTAR GPS USER EQUIPMENT INTRODUC-TION”, September 1996,

<<https://www.banggood.com/TP4056-Micro-USB-5V-1A-Lithium-Battery-Charging-Module-Board-TE585-Lipo-Charger-p-1225993.html?cur_warehouse=CN>>, last accessed on 2019-11-02 at 3.00 pm.

[9]How to use GPS with Arduino, <<https://core-electronics.com.au/tutorials/how-to-use-gps-with-arduino.html>>, last accessed on 2019-11-02 at 3.00 pm.

[l2] Getting your constraints right-Constrain layout, <<https://proandroiddev.com/getting-your-constraints-right-constraintlayout-786254ed1166>>, last accessed on 2019-11-028 at 3.00 am.

[l1] How GPS works, <<https://www.maptoaster.com/maptoaster-topo-nz/articles/how-gps-works/how-gps-works.html>>, last accessed on 2019-11-028 at 3.00 am.

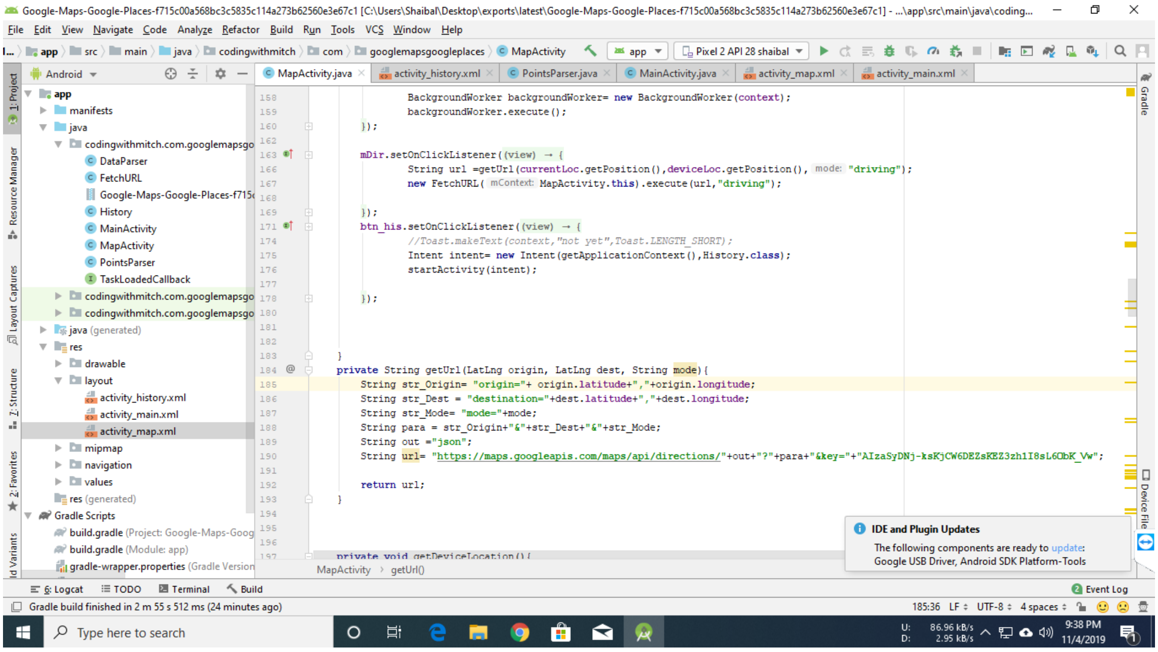
[l5] Encrypt a column of data in SQL, <<https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/encrypt-a-column-of-data?view=sql-server-ver15>>, last accessed on 2019-11-028 at 3.00 am.

[l3] Speed sensor, <<https://www.brainy-bits.com/speed-sensor-with-arduino/>>, last accessed on 2019-11-028 at 3.00 am.

[l4] Microphone Sound sensor, <<https://randomnerdtutorials.com/guide-for-microphone-sound-sensor-with-arduino/>>, last accessed on 2019-11-028 at 3.00 am.

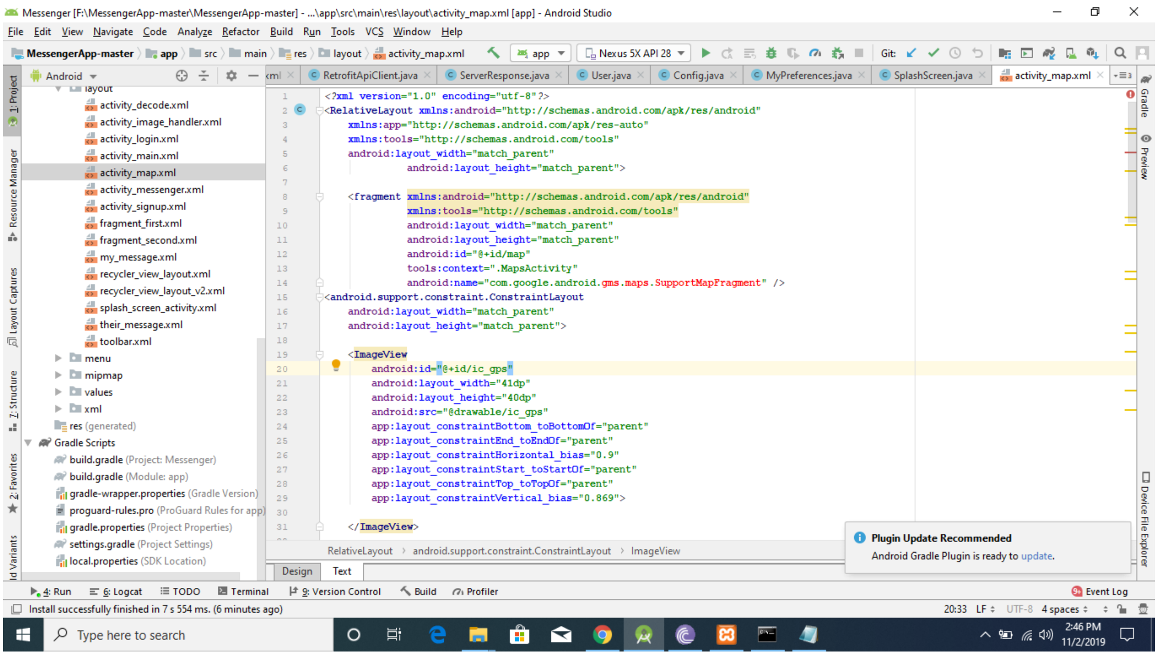
Appendix A

Java Code



Appendix B

XML Code



Document Viewer

Turnitin Originality Report

Processed on: 02-Nov-2019 17:47 +06

ID: 1205483081

Word Count: 4383

Submitted: 1

REALTIME GPS TRACKER SYSTEM By Mohshin Ahamed Sharif Shaibal

Similarity Index	Similarity by Source	
10%	Internet Sources:	4%
	Publications:	1%
	Student Papers:	10%

include quoted exclude bibliography exclude small matches mode:

quickview (classic) report ▼ Change mode print refresh download

1% match (Internet from 13-Sep-2013)
http://www.ijser.org

1% match (student papers from 09-Apr-2018)
Submitted to Daffodil International University on 2018-04-09

1% match (student papers from 03-Apr-2019)
Submitted to Ganpat University on 2019-04-03

<1% match (student papers from 11-Jan-2018)
Submitted to Staffordshire University on 2018-01-11

<1% match (student papers from 30-Oct-2019)
Submitted to Daffodil International University on 2019-10-30

<1% match (Internet from 18-Dec-2018)
https://www.innovativeeideas.com/2017/03/

<1% match (student papers from 07-Apr-2018)
Submitted to Daffodil International University on 2018-04-07

<1% match (student papers from 07-Apr-2018)
Submitted to Daffodil International University on 2018-04-07

<1% match (Internet from 28-Oct-2019)
http://single9.net

<1% match (student papers from 28-Apr-2019)
Submitted to University of Liverpool on 2019-04-28

<1% match (student papers from 28-May-2019)
Submitted to Edith Cowan University on 2019-05-28

<1% match (student papers from 30-Nov-2018)