COLLEGE BUS TRACKING HYBRID APP USING ADVANCED GOOGLE CLOUD SERVICES AND IOT

A PROJECT REPORT

Submitted by

NAFEEZ ABBAS J [REGISTER NO: 211417104156]

NARESH D [REGISTER NO: 211417104160]

SALMAN KHAN S [REGISTER NO: 211417104232]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“COLLEGE BUS TRACKING HYBRID APP USIND ADVANCED GOOGLE CLOUD SERVICES AND IOT”** is the bonafide work of **“NAFEEZ ABBAS J (211417104156), NARESH D (211417104160), SALMAN KHAN S (211417104232)”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. S. MURUGAVALLI, M.E., Ph.D.,	**S.A.K. JAINULABUDEEN, M. Tech.,**
PROFESSOR	**SUPERVISOR**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,	NASARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on ....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Our sincere, humble and heartfelt thanks and praises to our honorable Founder and Chairman Late **Dr. JEPPIAAR, M.A., B.L., Ph.D.,** for his sincere Endeavor in educating us in his reputed institution.

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **TMT. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D.,** and **TMT. SARANYA SREE SAKTHI KUMAR, B.E.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the Head of the CSE Department, **Dr. S. MURUGAVALLI, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank our Supervisor **Mr. S.A.K. JAINULABUDEEN, M. Tech.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NAFEEZ ABBAS J
NARESH D
SALMAN KHAN S

ABSTRACT

The current generation requires information on a regular basis. The use of technology has been steadily growing. As a result, we devised an innovative solution to "College Bus Tracking hybrid app using advanced Google cloud services and IOT," combining current technologies with the need for knowledge transmission. The device consists of a Mobile Application for students to use and a Hardware setup that shares the bus's location. The Hardware system uses IOT which includes an Arduino UNO, GSM, GPS modules, etc. The Hardware is setup in the bus such that its location can be accessed. The mobile app uses third party cloud services to access this location information, allowing students to know where they are and make plans accordingly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

GPS	Global Positioning System
GSM	Global System for Mobile Communication
GPRS	General Packet Radio Service
SHA	Secure Hash Algorithm
API	Application Programming Interface
SMS	Short Message Service
IP	Internet Protocol
IDE	Integrated Development Environment
USB	Universal Serial Bus
EEPROM	Electronically Erasable Programmable Read-Only Memory
LCD	Liquid Crystal Display
UI	User Interface
SDK	Software Development Kit
LBS	Location Based Service
HTTP	Hypertext Transfer Protocol

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

Vehicle tracking systems are widely used by fleet operators for fleet management functions such as establishing contact between the college server and the bus system, which is capable of delivering real-time data about the current location of buses, sending group messages, such as warning messages to students waiting at the next stop, changes in the current route, bus number, and so on, saving time of the students. A vehicle tracking system is an electronic device that is mounted in a vehicle which allows the owner or a third party to monitor the location of the vehicle. It works by employing GPS and GSM technologies to continuously track a moving vehicle. A microcontroller is serially connected to a GSM modem and GPS receiver, which is used to transmit the vehicle's position (latitude and longitude) from a remote location. The first fully operational GPS/Loran-based vessel monitoring system keeps track of the workstation, communications, and onboard navigation systems, giving the marine fleet operator a complete image. In applications such as shipping, scheduling, harbour operations, and route verification, the device is a valuable tool for the fleet operator. Furthermore, this concept can be extended to the broader issue of secure hazardous cargo transportation.

This project is implemented using Android Studio with Flutter Framework and the Firebase. The application will automatically view maps and routes to various locations using the GPS system, as well as monitor the bus position using client-server technology and forward it to the client computer. It uses simple distance measurements between two locations and provides necessary route information so that people can easily board buses on the designated route. The user is given specific location information as well as the bus number so that the student can correctly locate the bus. The data is stored on a remote server. As a result, records can be easily manipulated on the computer itself, reducing the server load.

1.2 PROBLEM DEFINITION

Many colleges do use paper and archives to keep track of bus routes and schedules, as well as provide information by notifications, which is ineffective. As a result, a structured method of keeping records and providing information as needed is required. Furthermore, students are unaware of the bus's proper timing. Some students wait for the bus being unaware of that the bus had already been misses and they are late for the class. Therefore, a smart system is required that provides remote users with real-time bus information. As a result, we proposed a new system to address the shortcomings of the college transportation system.

CHAPTER 2
LITERATURE SURVEY

2. LITERATURE SURVEY

A literature Survey is an objective, critical summary of published research literature relevant to a topic under consideration for research. Its purpose is to create familiarity with current thinking and research on a particular topic, and may justify future research into a previously overlooked or understudied area. It is the most important part of the report as it gives a direction in the area of research. It helps to set a goal for the analysis thus giving out problem statement. A literature review in respect of the project, the researches made by various analysts – their methodology (which is basically their abstract) and the conclusions they have arrived at. It also gives an account of how this research has influenced the thesis.

TITLE: Location based alarm system depending on longitude and latitude by G. V. M. Vasuki [1]

DESCRIPTION: Location Based alarm using GPS is an attempt to add an alarm facility for mobiles, based on the location of the device and to find the nearest places from the current location of the mobile device. The location-based alarm will give you alert when you reach your desired destination. Location based alarm is a GPS based alarm, if you set an alarm, it will make a sound and notification once it's detected you are within the user defined range from the destination. The user needs to save the current location using longitude and latitude, the alarm will ring when the user is near to the location. This location- based alarm is useful for the travelling sales persons and persons who are travelling in a train. The travelling sales person needs to do different kind of works in different places. It is difficult to remember all the places for him. So, by using this application he can set an alarm to the places, where he need to go. The GPRS settings must be enabled on a mobile device to use this application. We are using a SHA1 signature to generate a key Google map api key and google play service API for displaying the map in mobile device. The generation of SHA1 signature will be discussed in the methodology.

TITLE: Smart Vehicle Monitoring System using IOT, the paper published on IJDCST at March-April-2017, Issue-V5, I-3, SW-31 its ISSN code is 2320-7884 by N.Upendra Yadav and Professor Kamalakannan [2]

DESCRIPTION: This project uses accelerometer sensor which can detect the unevenness of vehicle and vibrations when an accident is occurred. This sends a signal to microcontroller. Vehicle accident detection system using GSM and GPS modems is done. Message notifications are sent to the mobile number which is prescribed. This monitoring system is composed of a GPS receiver, arduino and a GSM Modem. GPS Receiver gets the geo satellite information satellites in the form of latitude and longitude. The arduino processes this information and this processed information is sent to the user/owner using GSM modem. A GSM modem is interfaced to the MCU. Heat sensor used to detect temperature level and leakage of harmful gases in the vehicle.

TITLE: To determine precise location of object by Abid khan and Ravi Mishra [3]

DESCRIPTION: They have proposed tracking unit which it is attached and using GSM modem this information can be transmit to remote user. This system contains GPS and GSM modems along with ARM processor that is setup in the vehicle. Through SMS the location of vehicle can be reported. GSM and GPS technologies helps to track the vehicles exact information. Real time control is provided by SMS system. You can monitor the location from anywhere using this system.

TITLE: Hybrid Mobile Application Based on Ionic Framework Technologies by Dunka, Bakwa, Emmanuel, Edim, Oyerinde and Yinka [4]

DESCRIPTION: The objectives of this research article were to analyze native, web and hybrid mobile approaches to apps development and also to demonstrate how hybrid mobile apps can be built using Ionic framework. The approach used in this research is the adoption of the Ionic framework as a modern and emerging approach to offer solutions to the cross 7 platform mobile app development problems. In my perspective, the solution offered in this paper is expected to be straightforward yet adaptable. The historical progression in development started with web design approach then native app approach followed by hybrid mobile app approach.

Even if Flutter is a recent framework, it is becoming one of the most popular frameworks for hybrid application development. As Google trends2 show, Flutter is growing in popularity compared to its direct competitor, React Native. This implies that people are becoming more and more curious about Flutter. Even if the developers use React Native more than Flutter, they prefer Flutter, according to a survey performed by Stack Overflow in 2019, one of the most prominent questions and answers websites for programmers.

CHAPTER 3
SYSTEM ARCHITECTURE

3. SYSTEM ARCHITECTURE

System architecture is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems architecture could be seen as the application of systems theory to product development. There is some overlap with the disciplines of system analysis, systems architecture and systems engineering.

3.1 GPS TRACKING UNIT ARCHITECTURE

A GPS tracking unit is a device that calculates and monitors its precise location, and thus that of its carrier, at regular intervals using the Global Positioning System. It is usually carried by a moving vehicle or human. The captured location data can be stored in the tracking device or transmitted via a cellular (GPRS or SMS), radio, or satellite modem embedded in the unit to a central location data base or an Internet-connected computer. This allows the asset's location to be displayed against a map backdrop in real time or later when analysing the track with GPS tracking software. Data tracking software is available for smartphones equipped with GPS.

In order to obtain the GPS signal and measure the coordinates, a GPS tracker must have a GPS module. Data loggers have a large memory to store the coordinates, while data pushers have a GSM/GPRS modem to send the information to a central computer via SMS or GPRS in the form of IP packets.

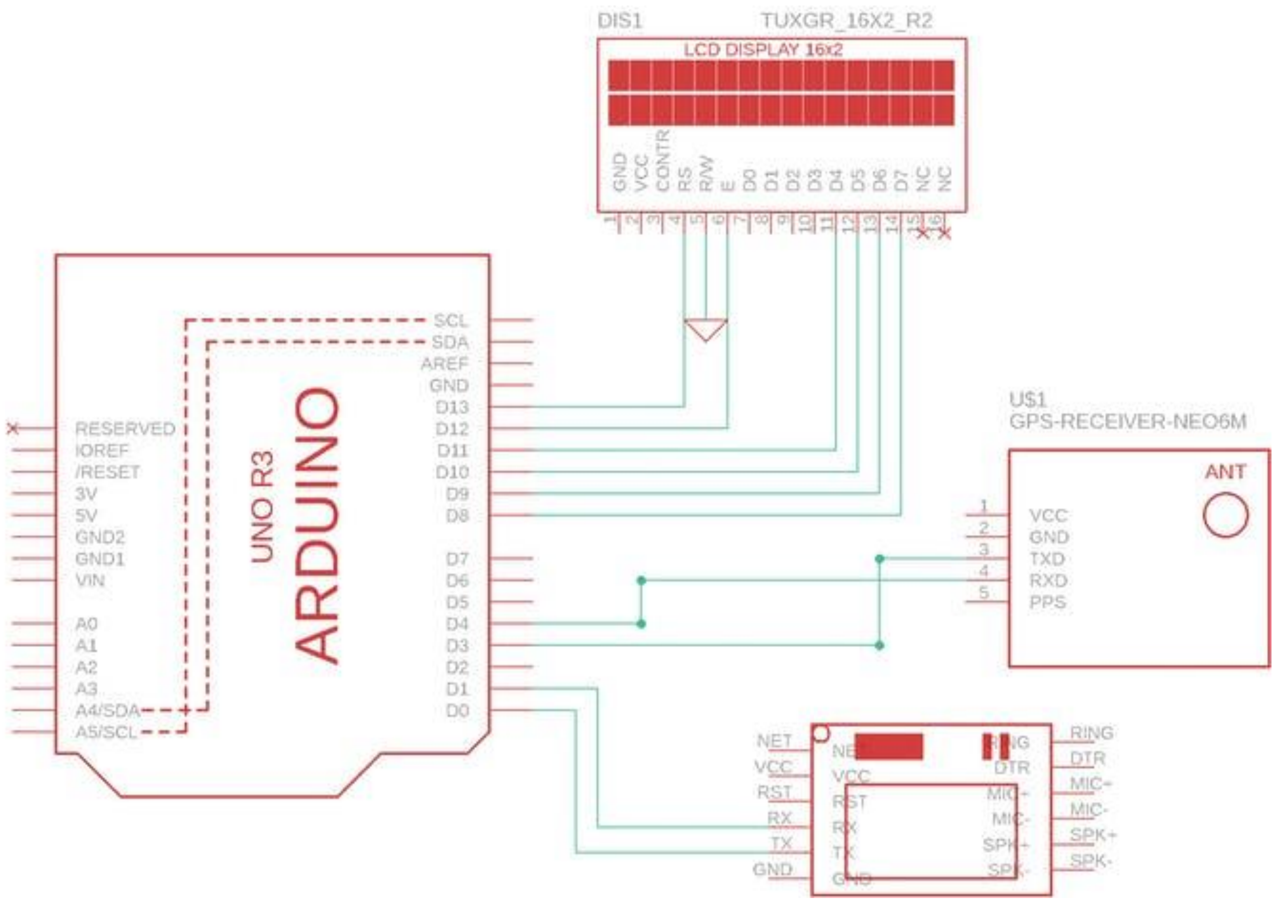
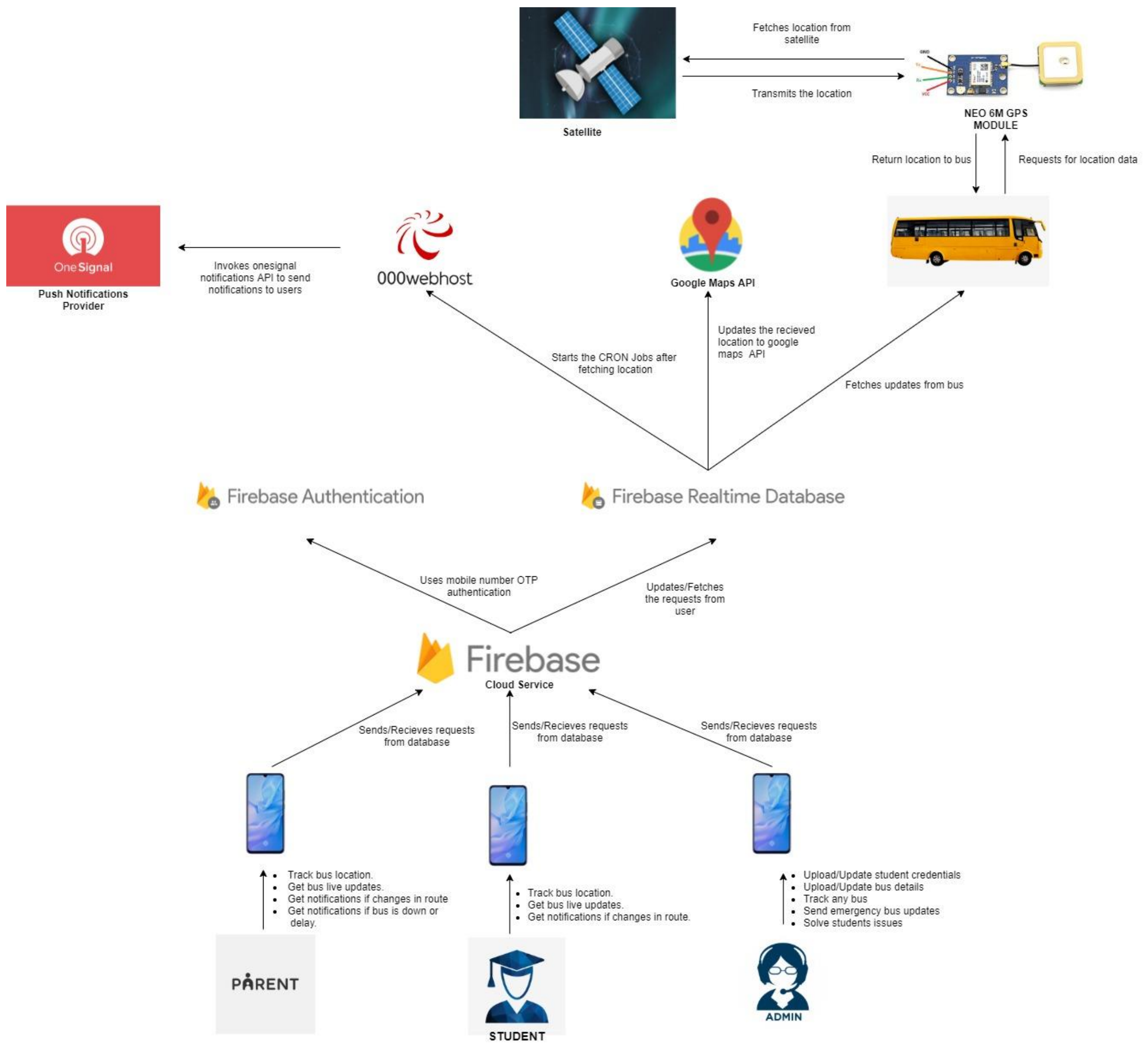


Figure 3.1 GPS Tracking Unit Architecture (Internet Source)

3.2 APPLICATION ARCHITECTURE

The GPS coordinates get stored in the firebase database which are then accessed by the application. The student can login into the app and has the privilege of selecting a route and bus from the list. The app shows the location of the bus on a map upon successful selections. The Admin on the other hand has access to the student details and bus locations. He / She uploads and manages the student data in the database and also can view the location of buses one at a time.



Satellite
Fetches location from satellite
Transmits the location
NEO 6M GPS MODULE
Return location to bus
Requests for location data
One Signal
Push Notifications Provider
Invokes onesignal notifications API to send notifications to users
000webhost
Google Maps API
Starts the CRON Jobs after fetching location
Updates the recieved location to google maps API
Fetches updates from bus
Firebase Authentication
Firebase Realtime Database
Uses mobile number OTP authentication
Updates/Fetches the requests from user
Firebase
Cloud Service
Sends/Recieves requests from database
Sends/Recieves requests from database
Sends/Recieves requests from database
Track bus location.
Get bus live updates.
Get notifications if changes in route
Get notifications if bus is down or delay.
Track bus location.
Get bus live updates.
Get notifications if changes in route.
Upload/Update student credentials
Upload/Update bus details
Track any bus
Send emergency bus updates
Solve students issues
PARENT
STUDENT
ADMIN

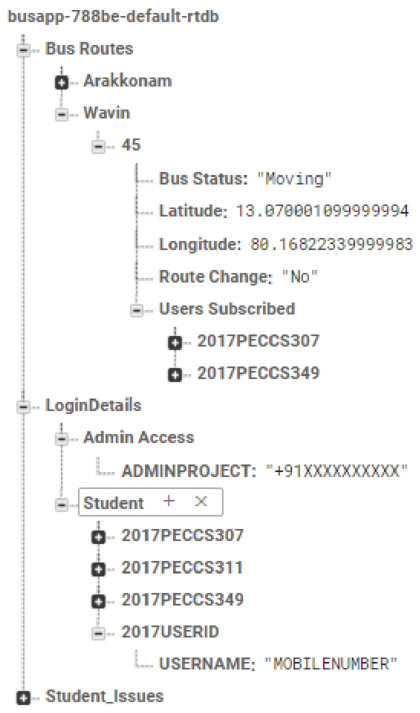
Figure 3.2 Application Architecture

3.3 FIREBASE ARCHITECTURE

Firebase has two major sections

1) Bus Routes
2) Login Details

Bus Routes lists all available routes and the buses under each route. Each bus has its latitude and longitude stored and its status. Login Details contains two sections. Admin Access has the mobile number of the admin. Student section contains details all the students. It includes the students Name, Roll Number and Mobile Number.



```
busapp-788be-default-rtdb
  Bus Routes
    Arakkonam
    Wavin
      45
        Bus Status: "Moving"
        Latitude: 13.070001099999994
        Longitude: 80.16822339999983
        Route Change: "No"
        Users Subscribed
          2017PECCS307
          2017PECCS349
  LoginDetails
    Admin Access
      ADMINPROJECT: "+91XXXXXXXXXX"
    Student + ×
      2017PECCS307
      2017PECCS311
      2017PECCS349
      2017USERID
        USERNAME: "MOBILENUMBER"
  Student_Issues
```

Figure 3.3 Firebase Architecture

CHAPTER 4

SYSTEM DESIGN

4. SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of system analysis, systems architecture and systems engineering.

The importance of this phase may be understood by reason of the fact that it involves identifying data sources, the nature and type of data that is available. This facilitates understanding what kind of data is available and by whom it is supplied to the system so that the system may be designed considering all the relevant factors.

4.1 PROPOSED SYSTEM

The App will be designed in a such a way that the student can select the route at the time of sign up. This enables the student to view of list of buses available in that particular route and can also access the locations of them.

In Addition to it, Students will have an option to set a reminder received when the bus is nearing the student's location. The student can also receive updates on bus route changes and the buses that are available on a particular day.

The Admin can have access to all the bus routes that are provided and can view their locations. Also, the Admin has access to the driver details. He / She is responsible for updating the bus route changes such that the information is made available to the students.

4.2 SYSTEM REQUIREMENT SPECIFICATION

A System Requirements Specification (abbreviated SRS when need to be distinct from a Software Requirements Specification SRS) is a structured collection of information that embodies the requirements of a system.

A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

4.3 HARDWARE ENVIRONMENT

ARDUINO UNO

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 (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE, via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word “uno” means “one” in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The Atmega328 on the board comes pre-programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

NEO-6M GPS MODULE

The NEO-6M GPS module is a well-performing complete GPS receiver with a built-in 25 x 25 x 4mm ceramic antenna, which provides a strong satellite search capability. With the power and signal indicators, we can monitor the status of the module. This module has the data backup battery which can save the data when the main power is shut down accidentally.

Features of NEO-6M GPS Module are:

- A complete GPS module with an active antenna integrated, and a built-in EEPROM to save configuration parameter data.
- Built-in 25 x 25 x 4mm ceramic active antenna provides strong satellite search capability.
- Equipped with power and signal indicator lights and data backup battery.
- Power supply: 3-5V; Default baud rate: 9600bps.
- Interface: RS232 TTL.

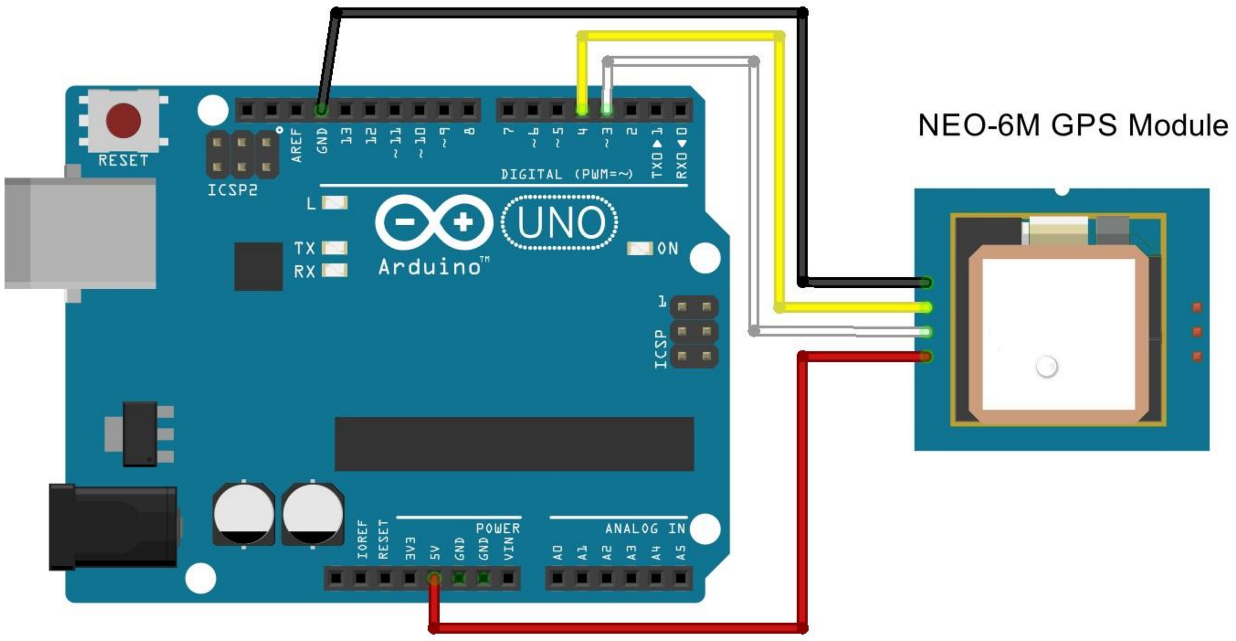


Figure 4.1 NEO-6M GPS Module (Internet Source)

SIM900A GSM / GPRS MINIMUM SYSTEM MODULE

GPRS module is a breakout board and minimum system of SIM900 Quad-band/SIM900A Dual-band GSM/GPRS module. It can communicate with controllers via AT commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands). This module supports software power on and reset.

Features of SIM900A GSM are:

- Quad-Band 850/ 900/ 1800/ 1900 MHz
- Dual-Band 900/ 1900 MHz
- GPRS multi-slot class 10/8GPRS mobile station class B
- Compliant to GSM phase 2/2+Class 4 (2 W @850/ 900 MHz)
- Class 1 (1 W @ 1800/1900MHz)
- Control via AT commands
- Low power consumption: 1.5 mA (sleep mode)
- Operation temperature: -40°C to +85 °C

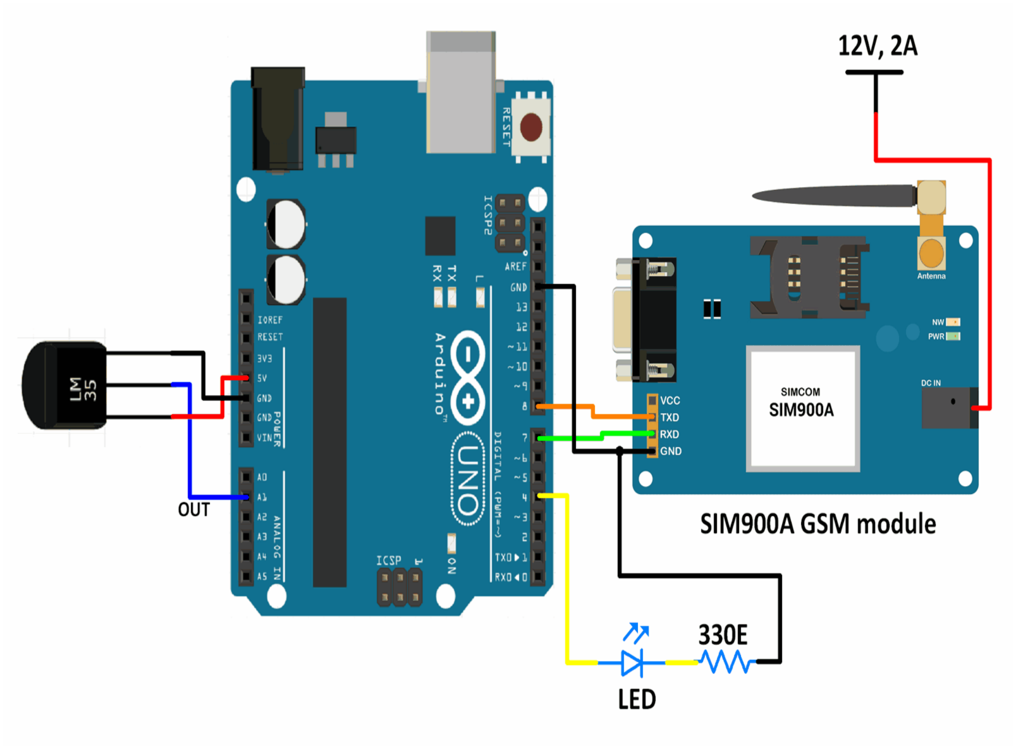


Figure 4.2 SIM 900A GSM Module (Internet Source)

LIQUID CRYSTAL DISPLAY (LCD)

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and seven-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made from a matrix of small pixels, while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the color of the backlight, and a character negative LCD will have a black background with the letters being of the same color as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance.

KEYPAD

A keypad is a set of buttons arranged in a block or pad which bear digits, symbols or alphabetical letters. Pads mostly containing numbers and used with computers are numeric keypads. Keypads are found on devices which require mainly numeric input such as calculators, television remotes, push-button telephones, vending machines, ATMs, Point of Sale devices, combination locks, and digital door locks. Many devices follow the E.161 standard for their arrangement.

4.4 SOFTWARE ENVIRONMENT

ANDROID STUDIO WITH FLUTTER FRAMEWORK

Android 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:

- 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[18]
- 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[19]
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase.

The first version of Flutter was known as codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit, with the stated intent of being able to render consistently at 120 frames per second. During the keynote of Google Developer Days in Shanghai, Google announced Flutter Release Preview 2, which is the last big release before Flutter 1.0. On December 4, 2018, Flutter 1.0 was released at the Flutter Live event, denoting the first “stable” version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.

On May 6, 2020, the Dart SDK in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the Metal API, improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications as well as early-access desktop application support for Windows, MacOS, and Linux.[9]

The major components of Flutter include:

- Dart platform
- Flutter engine
- Foundation library
- Design-specific widgets
- Flutter DevTools

Dart platform

Flutter apps are written in the Dart language and make use of many of the language’s more advanced features.

On Windows, macOS, and Linux Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for “hot reload”, with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

Release versions of Flutter apps are compiled with ahead-of-time (AOT) compilation on both Android and iOS, making Flutter's high performance on mobile devices possible.

Flutter engine

Flutter's engine, written primarily in C++, provides low level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS. The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

Foundation library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

Design-specific widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and Cupertino widgets implement Apple's iOS Human interface guidelines.

Widgets

Flutter uses a variety of widgets to deliver a fully functioning application. These widgets are Flutter's framework architecture. Flutter's Widget Catalogue provides a full explanation and API on the framework.

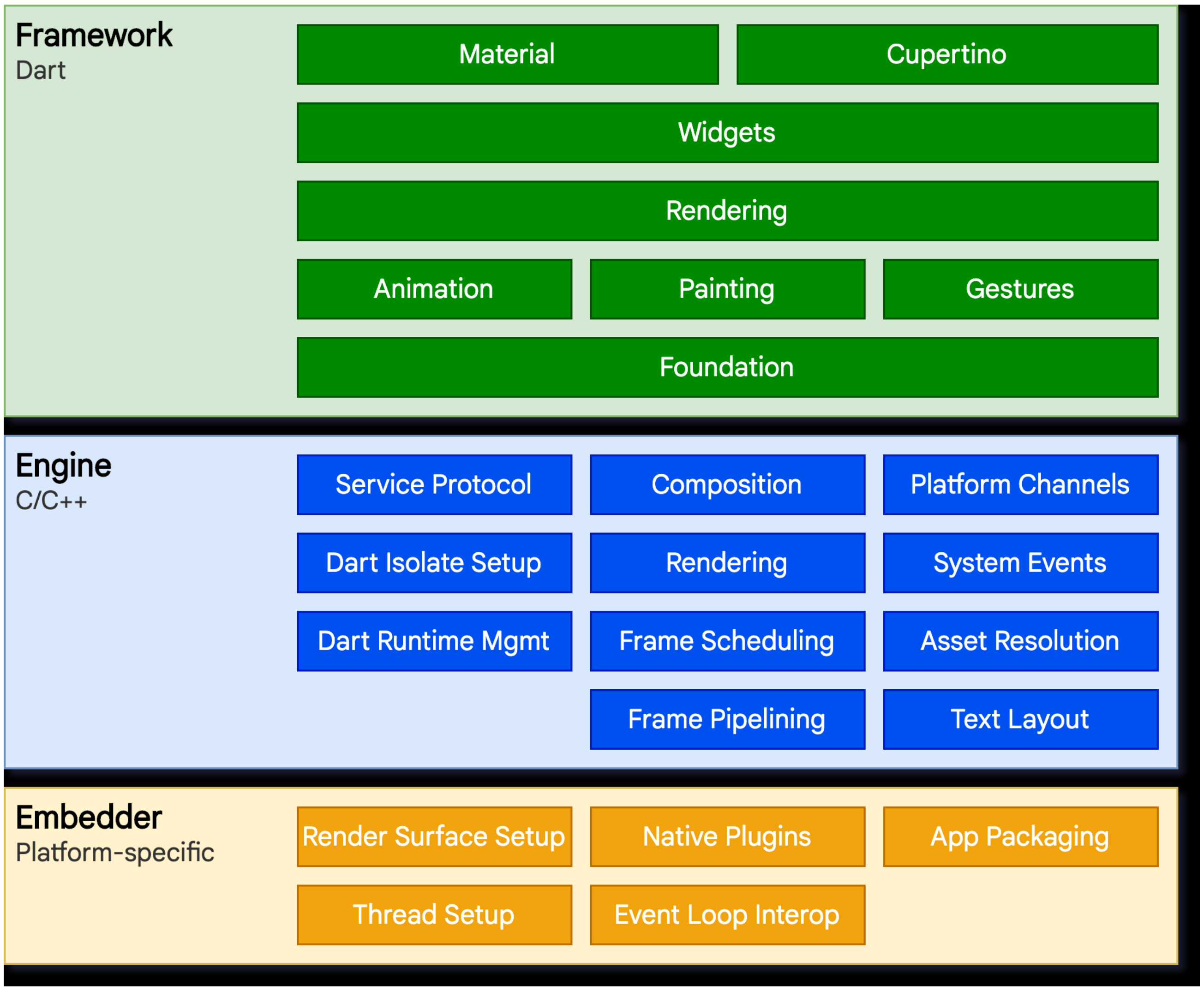


Figure 4.3 Flutter Architecture (Internet Source)

FIREBASE

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development.

Firebase evolved from Envolve, a prior start-up founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in September 2011 and it launched to the public in April 2012.

Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

In May 2012, a month after the beta launch, Firebase raised $1.1 million in seed funding from venture capitalist Flybridge Capital Partners, Greylock Partners, Founder Collective, and New Enterprise Associates. In June 2013, the company further raised $5.6 million in Series A funding from Union Square Ventures and Flybridge Capital Partners.

In 2014, Firebase launched two products. Firebase Hosting and Firebase Authentication. This positioned the company as a mobile backend as a service.

In October 2014, Firebase was acquired by Google. A year later, in October 2015, Google acquired Divshot, an HTML5 web-hosting platform, to merge it with the Firebase team.

In May 2016, at Google I/O, the company’s annual developer conference, Firebase introduced Firebase Analytics and announced that it was expanding its services to become a unified backend-as-a-service (BaaS) platform for mobile developers. Firebase now integrates with various other Google services, including Google Cloud Platform, AdMob, and Google Ads to offer broader products and scale for developers. Google Cloud Messaging, the Google service to send push notifications to Android devices, was superseded by a Firebase product, Firebase Cloud Messaging, which added the functionality to deliver push notifications to both iOS and web devices. In January 2017, Google acquired Fabric and Crashlytics from Twitter to add those services to Firebase.

In October 2017, Firebase has launched Cloud Firestore, a real-time document database as the successor product to the original Firebase Realtime Database.

The Firebase platform has 18 products split into three groups: Develop, Quality, and Grow.

Figure 4.4 Firebase Logo

ARDUINO IDE

The Arduino IDE is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. 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 are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board’s firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

Arduino IDE is a derivative of the Processing IDE, however as of version 2.0, the Processing IDE will be replaced with the Visual Studio Code-based Eclipse Theia IDE framework.

With the rising popularity of Arduino as a software platform, other vendors started to implement custom open-source compilers and tools (cores) that can build and upload sketches to other microcontrollers that are not supported by Arduino’s official line of microcontrollers.

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging and other advanced features.

CHAPTER 5
SYSTEM IMPLEMENTATION

5. SYSTEM IMPLEMENTATION

A process of putting a planned system into action. The stage of systems development in which hardware and software are acquired, developed and installed, the system is tested and documented, people are trained to operate and use the system, and an organization converts to the use of a newly developed system.

5.1 OVERVIEW

On installation or boot up, the location-based bus tracking device server should automatically start. The server database will then compare the current latitude and longitude value to the previous latitude and longitude value. The database manager manages the application's database, after which the user searches for the bus and retrieves information from the database for that specific location. The application then acts in accordance with the user interface, allowing the user to add, delete, and update the database.

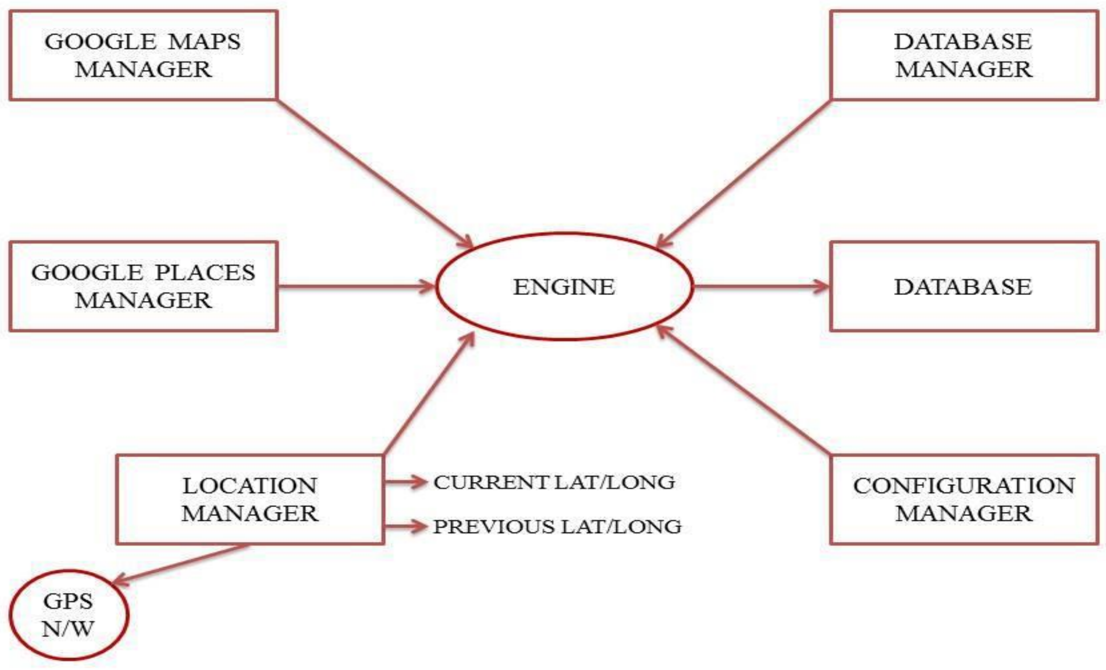


Figure 5.1 Working of a Location Based Service (LBS) Component (Internet Source)

5.2 IMPLEMENTION

The app is developed using Flutter SDK. Flutter is a new open-source technology from Google that allows you to build native Android and iOS apps with a single codebase. The reactive development architecture is followed by Flutter, but with a twist. The most important thing to understand about reactive programming is that it automatically updates UI content when we change variables in the code. This idea is followed by React Native, but it makes use of the JavaScript bridge to access OEM widgets. However, since the app must cross this bridge each time it accesses a widget, it creates performance issues. Flutter, on the other hand, skips the bridge entirely and interacts with the native platform through Dart.

Project is designed in four modules where each module is responsible for different aspects.

Module 1 – Admin

The admin logs into the system with user name and password, the admin manages routes, students and drivers. Admin can add and update the students details and the driver details as well. Also, the Admin manages the list of routes and the buses available in each route. All the information is made persistent in the database. Admin manages the routes and makes changes in it if necessary. The Admin also has the access to the location of the buses where he / she can choose the Bus Number and view its current location using the Google Maps API.

Module 2 – Student / Parent

Student can login to the system using the Unique Register Number provided to him / her by the college and the mobile number submitted. Upon successful login, the student will prompted to select the route and bus number to track it. Students can view the location of bus in real-time on the map. Students can also set an alarm which starts when the bus is nearing him. Student receives updates on route and bus changes as well.

Module 3 - Firebase

Firebase provides the NOSQL database for storing the student and bus record. Lot of API’s and function calls have been used in order to create, read, update, delete (CRUD) operations for user and admin data. Other than that firebase also provides push notification facility to our app. Adding to this, another third-party provider called as 000webhost.com used to update the bus co-ordinates from the Arduino UNO to database.

Module 4 - Hardware

This module tracks and updates the bus coordinates to the database through API calls. It needs a proper working sim card with data transfer facility (GPRS). The GPS coordinates are made persistent in a third-party cloud using the GSM. The GSM makes an HTTP request to the Cloud and hence stores the coordinates which are in turn fetched at the Mobile App. Google Map API uses these coordinated to show the location of the bus.

5.3 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

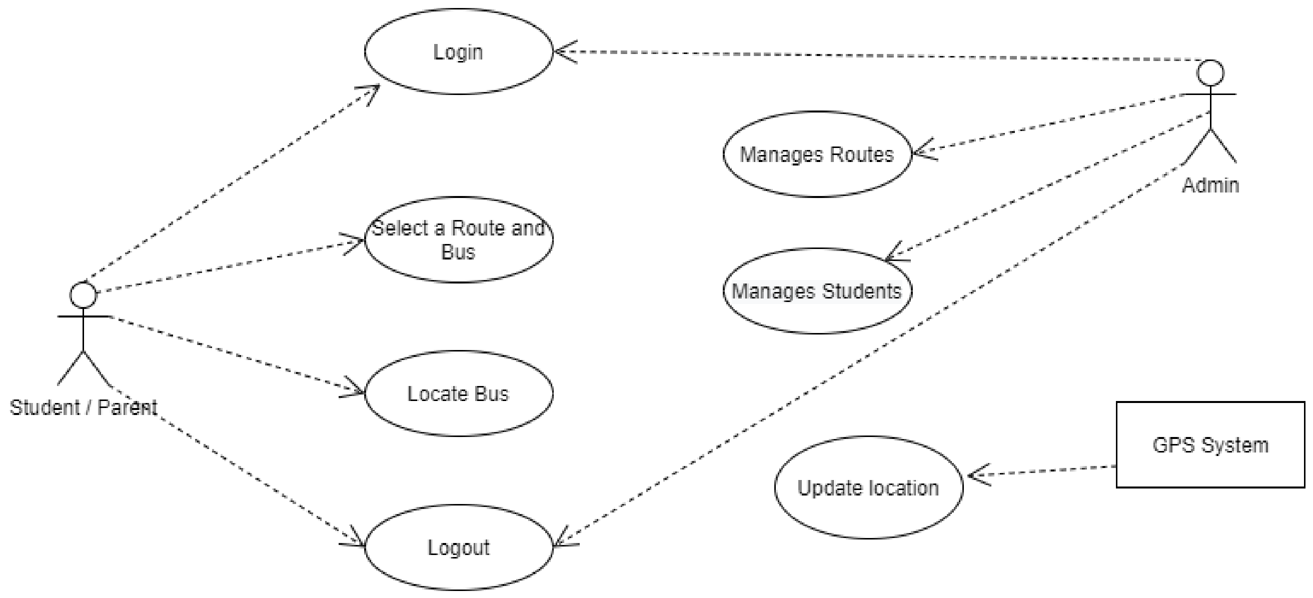


Figure 5.2 Use Case Diagram

5.4 ACTIVITY DIAGRAM

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.

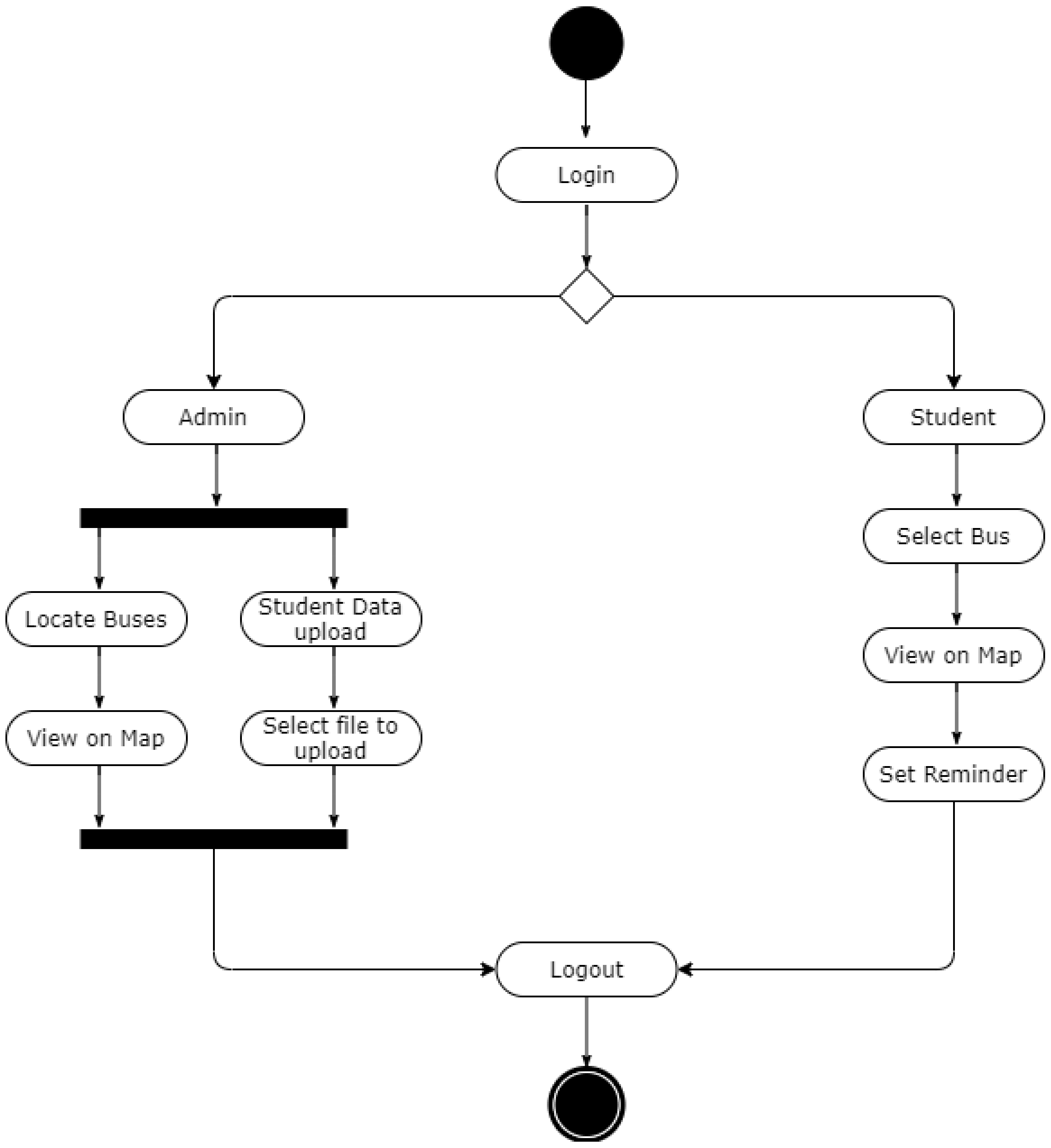


Figure 5.3 Activity Diagram

CHAPTER 6

SYSTEM TESTING

6. SYSTEM TESTING

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

Test Case No.	Action	Expected Output	Actual Output	Result
1	Number of characters of Roll No. is equal to 12 characters	Login button is enabled	Login button is enabled	Pass
2	Number of characters of Roll No. is less than 12 characters	Login button is disabled	Login button is disabled	Pass
3	Number of characters of Roll No. is greater than 12 characters	Subsequent characters are not accepted	Subsequent characters are not accepted	Pass
4	Click Help	Displays Help screen	Displays Help screen	Pass
5	Click Login button	Enters into OTP page, OTP is verified automatically and home page is displayed	Enters into OTP page, OTP is verified automatically and home page is displayed	Pass
6	Click Select Route	Displays the list of routes and buses available	Displays the list of routes and buses available	Pass
7	Click OK button	Current location of the bus is displayed in the map	Current location of the bus is displayed in the map	Pass

Test Case No.	Action	Expected Output	Actual Output	Result
8	Click Reminder	Displays the popup to enable or disable the reminder	Displays the popup to enable or disable the reminder	Pass
9	Click Updates	Displays the updates of the bus	Displays the updates of the bus	Pass
10	Click Log out button	Logout from the app	Logout from the app	Pass

6.1 TESTING STRATEGIES

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

6.1.1 Unit Testing:

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

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. This testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

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. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false Sides.
- Execute all loops at their boundaries and within their operational bounds
- Execute internal data structures to ensure their validity

6.1.2 Integration Testing:

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.

6.1.3 System 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.

6.1.4 Acceptance Testing:

It is a pre-delivery testing in which entire system is tested at client’s site on real world data to find errors.

6.2 Testing Approaches

Testing can be done in two ways:

6.2.1 Bottom-up Approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower-level ones they are tested individually and then linked with the previously examined lower-level modules.

6.2.2 Top-down approach:

This type of testing starts from upper-level modules. Since the detailed activities usually performed in the lower-level routines are not provided stubs are written. A stub is a module shell called by upper-level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower-level module.

6.3 Validation and Verification:

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfil its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle.

Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review.

Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

Verification: Are we building the product right?

Validation: Are we building the right product?

According to the Capability Maturity Model (CMMI-SW v1.1),

Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase [IEEE-STD-610].

Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements [IEEE-STD-610].

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product actually meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right" Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfil its intended use.

From testing perspective:

Fault – wrong or missing function in the code.

Failure – the manifestation of a fault during execution.

Malfunction – according to its specification the system does not meet its specified functionality.

CHAPTER 7

CONCLUSION

7. CONCLUSION

The findings of this study show that having a good understanding of a particular domain will help you get better results. This Project has been implemented on Hybrid platform. In addition, various attributes have been applied to the project that will benefit the system. The specifications and criteria are described above. This project is implemented using Android Studio with Flutter Framework and the Firebase. The application will automatically view maps and routes to various locations using the GPS system, as well as monitor the bus position using client-server technology and forward it to the client computer. It uses simple distance measurements between two locations and provides necessary route information so that people can easily board buses on the designated route. The user is given specific location information as well as the bus number so that the student can correctly locate the bus. It uses remote server as its database. As a result, records can be easily manipulated on the computer itself, reducing the server load.

APPENDICES

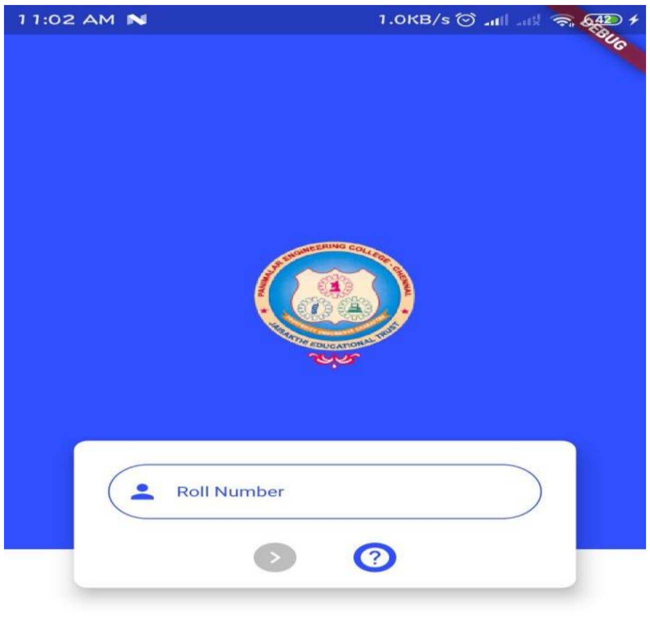
A.1 CODING

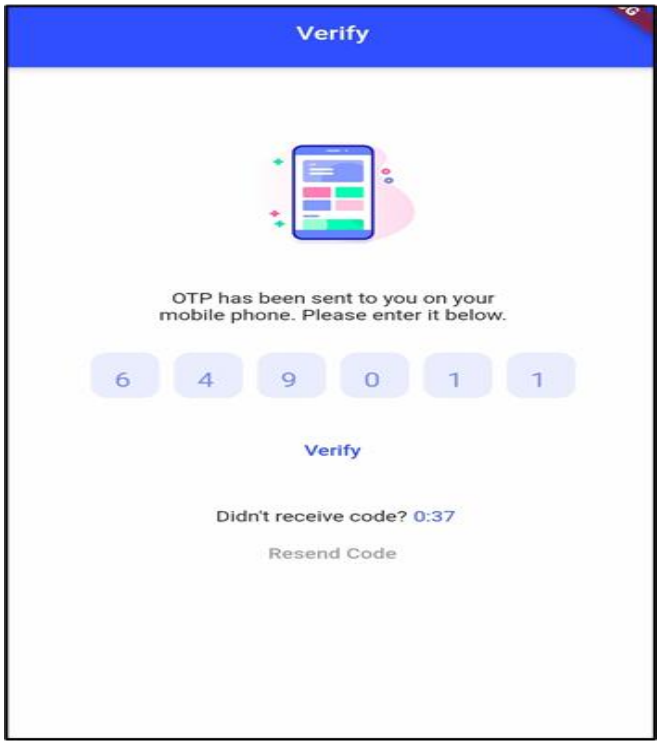
```
import 'dart:ffi';
import 'dart:typed_data';
import 'dart:async';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:shared_preferences/shared_preferences.dart';

class MapScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
    title: 'Flutter Google Maps Demo',
    home: MapSample(),
    );
  }
}


class MapSample extends StatefulWidget {
  @override
  State<MapSample> createState() => MapSampleState();
}


class MapSampleState extends State<MapSample> {

    Completer<GoogleMapController> _controller = Completer();

    /* The Lat Lng mentioned here is default Tamil Nadu location co-ordinates */

    static double Lat = 11.127;
    static double Lng = 78.6569;
    static LatLng lastPosition = LatLng(Lat, Lng);
    static double zoomLevel=14;
    String route, busNumber;

    /* This function checks if the route is already selected by user or not */
    void getValues() async {

       SharedPreferences prefs = await SharedPreferences.getInstance();
       route = await prefs.getString("route");
       busNumber = await prefs.getString("bus");

    }

    Map<dynamic, dynamic> values;
    var db = FirebaseDatabase.instance.reference();
    Marker marker;
    Circle circle;

    /* Initial camera position in google maps */
    CameraPosition _busPosition = CameraPosition(

       bearing: 192.8334901395799,
       target: LatLng(Lat ,Lng),
       zoom: zoomLevel,

    );
    Timer timer;

    @override
    void initState() {
       super.initState();
      /* For every two seconds we will get the values from database for updated
GPS Position */
                timer = Timer.periodic(Duration(seconds: 2), (timer)
                 =>_findmyBus());
    }


    /* A marker icon to show the bus position */
    Future<Uint8List> getMarker() async {

       ByteData byteData = await DefaultAssetBundle.of(context).load(
       "assets/busMarker.png");

       return byteData.buffer.asUint8List();
```

```
}


@override
Widget build(BuildContext context) {


return new Scaffold(
        /* The screen where we use google maps api to display the location of      th
   e bus */

   body: GoogleMap(

      markers: Set.of((marker != null) ? [marker] : []),
      circles: Set.of((circle != null) ? [circle] : []),
      initialCameraPosition: _busPosition,
      onMapCreated: (GoogleMapController controller) {
         _controller.complete(controller);
      },
   ),
 );
}

/* As bus moves the marker update call happens here */
void updateMarker (LatLng location, Uint8List imageData) {

   setState(() {
      marker = Marker(
         markerId: MarkerId("Bus"),
         position: location,
         draggable: false,
         zIndex: 2,
         flat: true,
         anchor: Offset(0.5, 0.5),
         icon: BitmapDescriptor.fromBytes(imageData)
      );


      circle = Circle(
         circleId: CircleId("car"),
         radius: 200,
            zIndex: 1,
            strokeColor: Colors.blue,
            strokeWidth: 2,
            center: location,
            fillColor: Colors.blue.withAlpha(50)
        );
    });
}

/* Function used to maintain new position, zoom angle of the bus */
Future<void> _findmyBus() async {
    await getValues();
    final GoogleMapController controller = await _controller.future;
    await db.once().then((DataSnapshot snap) {
        values = snap.value;
        values = values['Bus Routes'];
        values = values[route];
        values = values[busNumber];
        double aPos = values['Latitude'];
        double bPos = values['Longitude'];
        setState(() {
            Lat = aPos;
            Lng = bPos;
        });
    });

    await controller.getZoomLevel().then((value) {
        zoomLevel=value;
    });

    CameraPosition _busPosition1 = CameraPosition(
        target: LatLng(Lat ,Lng),
        zoom: zoomLevel,
    );

    Uint8List imageData = await getMarker();
    updateMarker(LatLng(Lat, Lng), imageData);
    LatLng latLng=LatLng(Lat, Lng);
    controller.animateCamera(CameraUpdate.newCameraPosition(_busPositi
    on1));
}}
```

A.2 SAMPLE SCREENS

i. Student Login

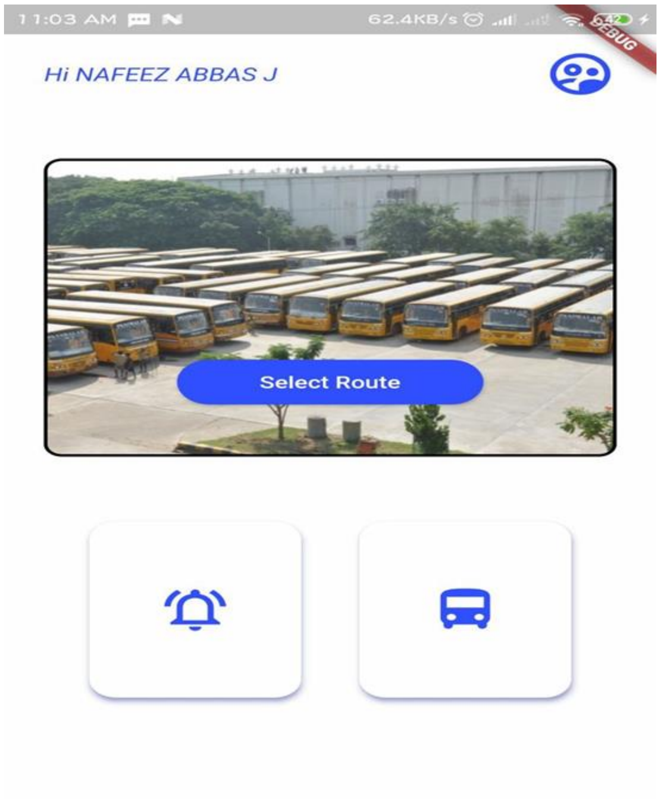


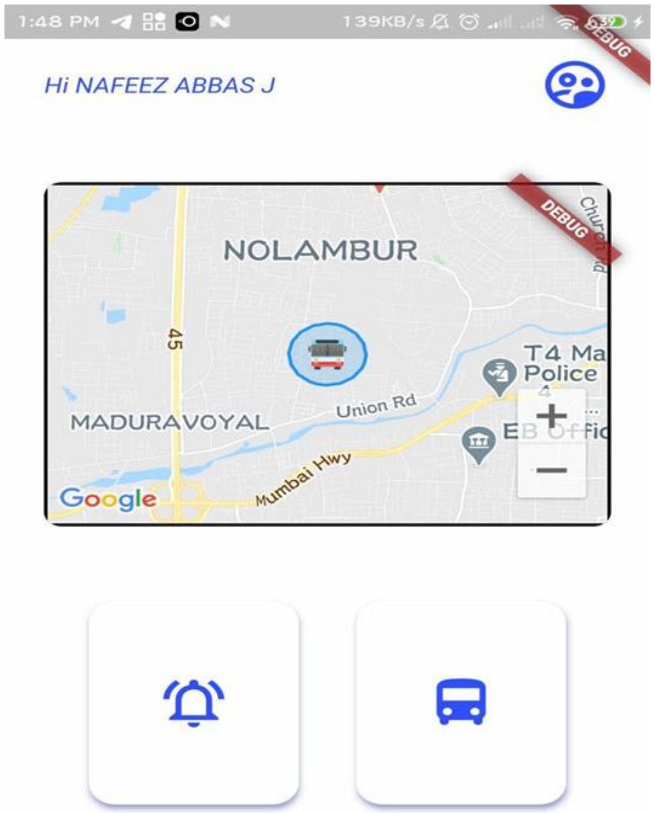


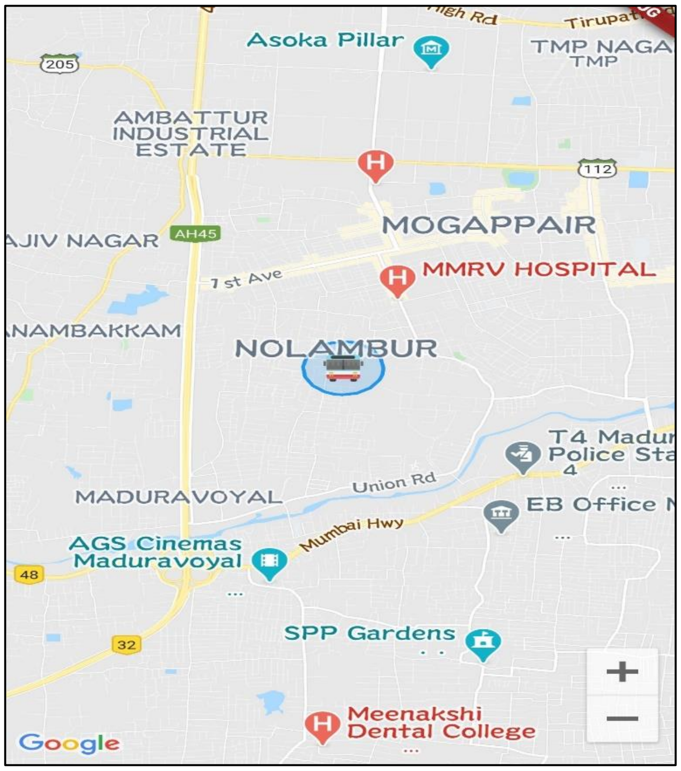
OTP has been sent to you on your mobile phone. Please enter it below.

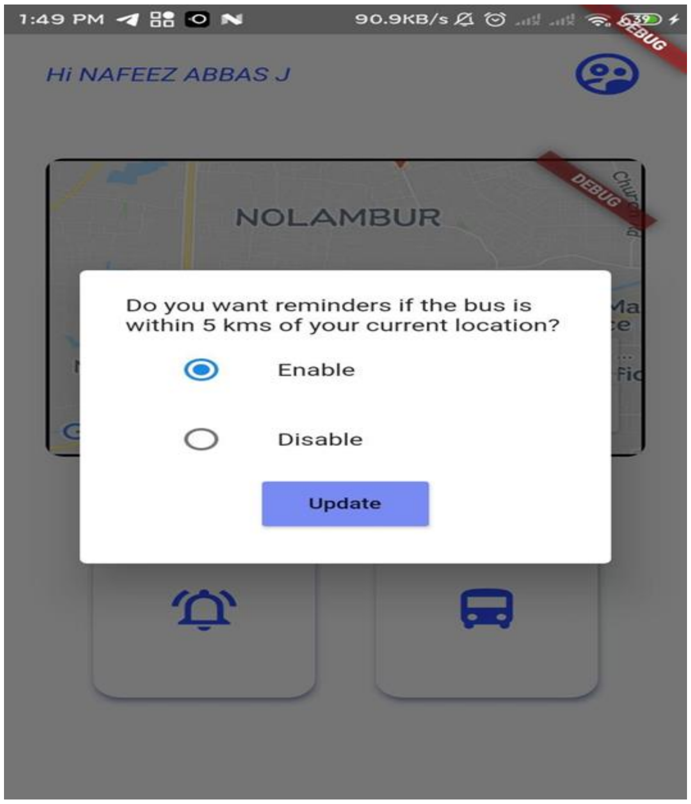
6 4 9 0 1 1

Didn't receive code? 0:37


11:03 AM
62.4KB/s
DEBUG
Hi NAFEEZ ABBAS J
Select Route


1:48 PM
139KB/s
DEBUG
Hi NAFEEZ ABBAS J
NOLAMBUR
45
T4 Ma
Police
MADURAVOYAL
Union Rd
EB Offic
Mumbai Hwy
Google

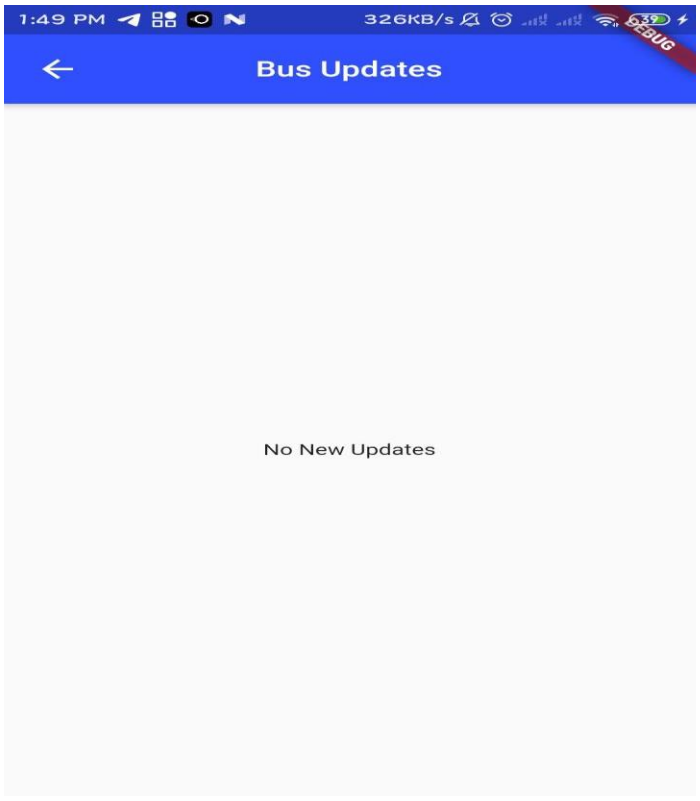




Do you want reminders if the bus is within 5 kms of your current location?

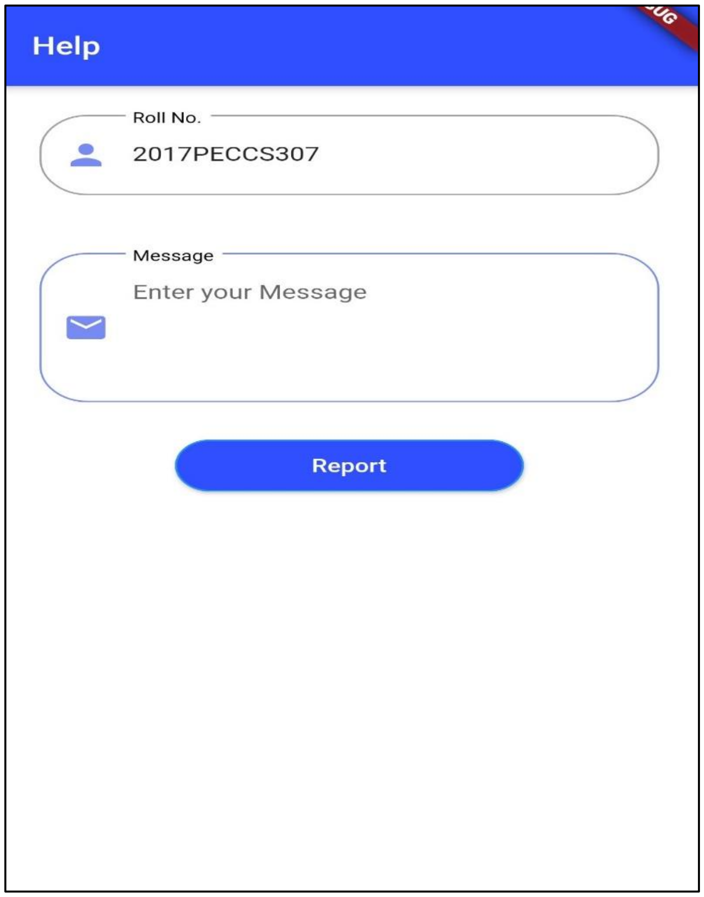
[✓ Enable]

Disable


1:49 PM 326KB/s

Bus Updates

No New Updates


Help

Roll No.

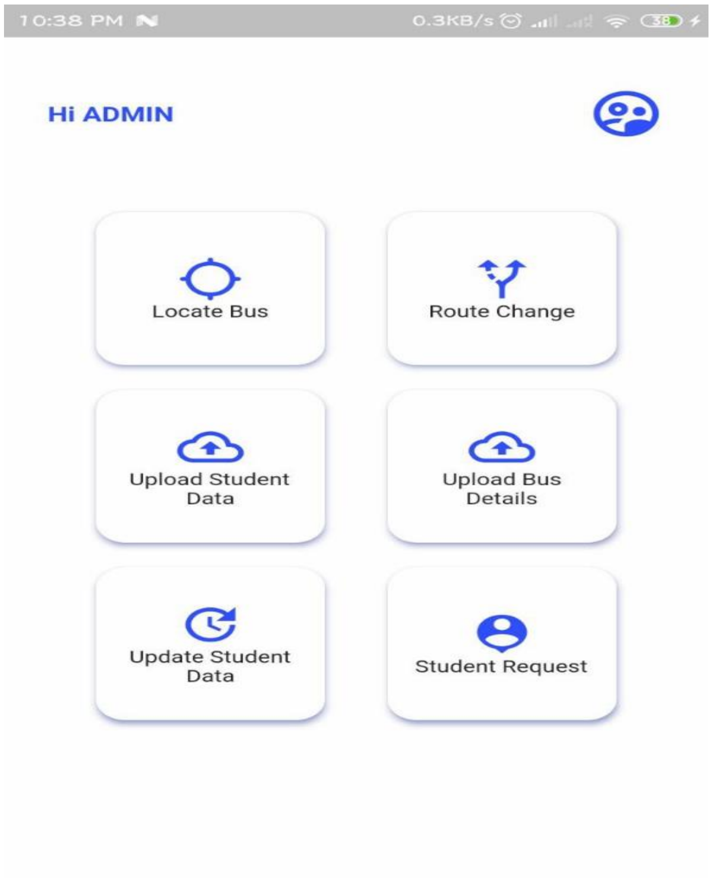
2017PECCS307

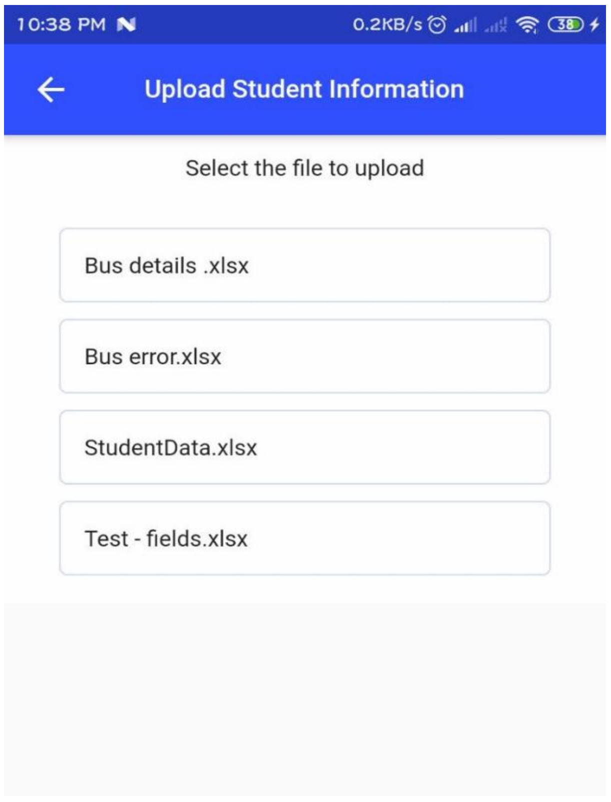
Message

Enter your Message

Report

ii. Admin Login





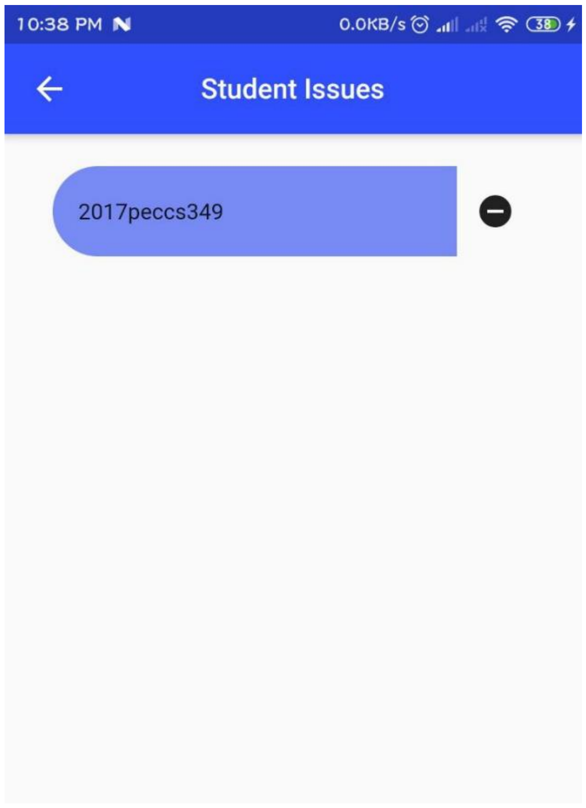
Select the file to upload

Bus details .xlsx

Bus error.xlsx

StudentData.xlsx

Test - fields.xlsx



2017peccs349

FUTURE WORKS

In future there will be an implementation for tracking a student’s location automatically and provide updates if student transfers from one bus to another. It will be very helpful to parents for tracking the student’s exact location. There will be also Estimated Time of Arrival (ETA) of bus which makes the tracking app more precise to get bus location. There will be an individual login for teachers/staffs which will be implemented in future.

REFERENCES

[1] A. Goel noV. Gruhn," Fleet Monitoring System for Advanced Tracking of Economic Vehicles ", Procedures for 2006 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2006), pages 2517-2522, Taipei, Taiwan, 08.10.2006-11.10.2006 .

[2] Chia-Hung Lien, Chi-Hsiung Lin, Ying-Wen Bai, Ming-Fong Liu and Ming-Bo Lin, "Home Remote Control System for Home Energy Management," Continued on 2006 IEEE Tenth International Symposium on Consumer Electronics ISCE 2006), eSt. Petersburg, Russia, pages 7-12, June 28 - July 1, 2006.

[3] E. D. Kalpan, GPS Understanding: Principles and Applications, Artech house Publishers, ISBN 0890067937, February 1996.

[4] Junaid Ali, Shaib Nasim, Taha Ali, Naveed Ahmed and syed Riaz un Nabi, "GSM implementation based on Commercial Automobile Tracker Using PIC 18F452 and Google Earth Embedded Development" Proceedings 2009 at the 2009 student conference IEEE on Research and development (SCOReD 2009), 16-18 Nov, 2009, UPM Serdang, Malaysia.

[5] M. McDonald, H. Keller, J. Klijnhout and V. Mauro, “Intelligent Transport Systems in Europe: Opportunity for future Research” International Science publisher , ISBN 981270082X, 2006.

[6] Muhammad Ali Mazidi, Janice Gillspie, McKinlay, Rolin D., "The Microcontroller in Embedded System: using Assembly and C," is the second edition published by Pearson Education. Panwar, Meenakshi and Mehra, Pawan. (2011),

[7]. B. Ferris, K. Watkins, and A. Borning, “Local tools to improve public transportation,” IEEE Pervasive Comput., Vol. 9, no. 1, pages. 13–19, January - March 2010.

[8]. B.aulfield and M.O’Mahony, “Assessment of users’ public transport information needs, ”IEEE Trans. Intell. Pass. Syst., Vol. 8, no. 1, pages 21-30, March 2007

[9] .M.Arikaa, S.Konomi, and K.Ohnishi, “Navitime: Supporting pedestrian walk in the real world,” IEEE Pervasive Comput., Vol. 6, no. 3, pages 21–29, Jul. - Sep. 2007.

[10]. K.Rehrl, S. Bruntsch, and H-J.Mentz, “Assisting Travelers: Development and Modeling of Private Travel,” IEEE Trans.. in Intelligent Transportation Systems, vol. 8, no. 1, pages 31-42, Mar 2007.

[11]. A. Repenning and A. Ioannidou, "Agents Agents: Guiding and Tracking Public Transportation Users," Proc. Working Conf. Advanced Visual Interfaces, ACM Press, 2006, pages 127-134.

[12] Development of Mobile Applications for Posyandu Administration Services Using Google Maps API Geolocation Tagging published in 2018 International Conference on Sustainable Information Engineering and Technology (SIET).

[13] Arduino Based Smart Energy Meter using GSM published in 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU).

[14] Dart and Flutter Packages, https://pub.dev/