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

Team Members

1. Mr. S.A.K. Jainulabudeen, M.Tech., Assistant Professor

2. Nafeez Abbas J

3. Naresh D

4. Salman Khan S

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

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.

I. INTRODUCTION

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.

II. 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.

III. SYSTEM ARCHITECTURE

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.

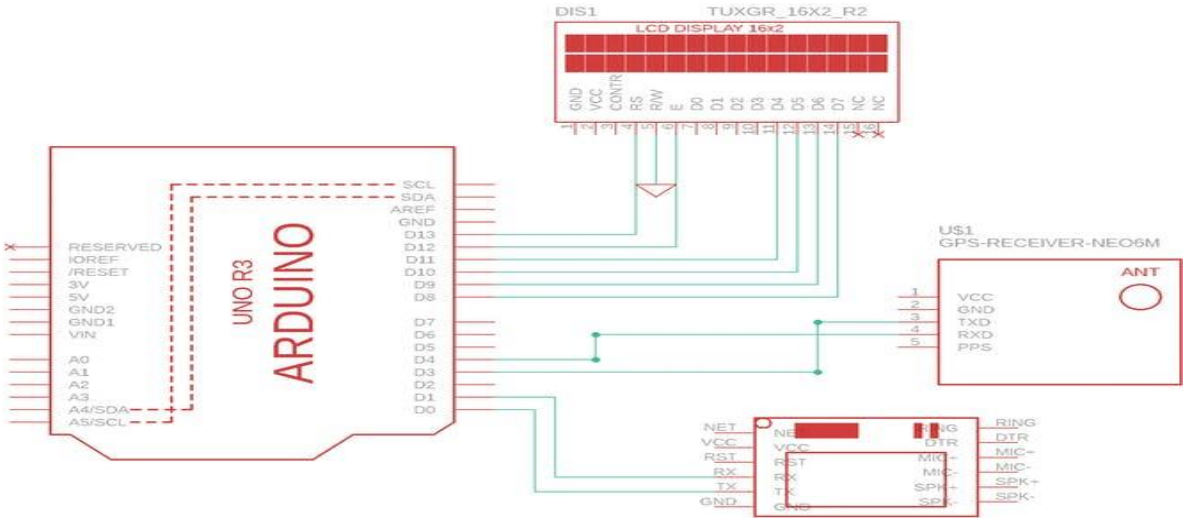


Figure 3.1 GPS Tracking Unit Architecture (Internet Source)

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.

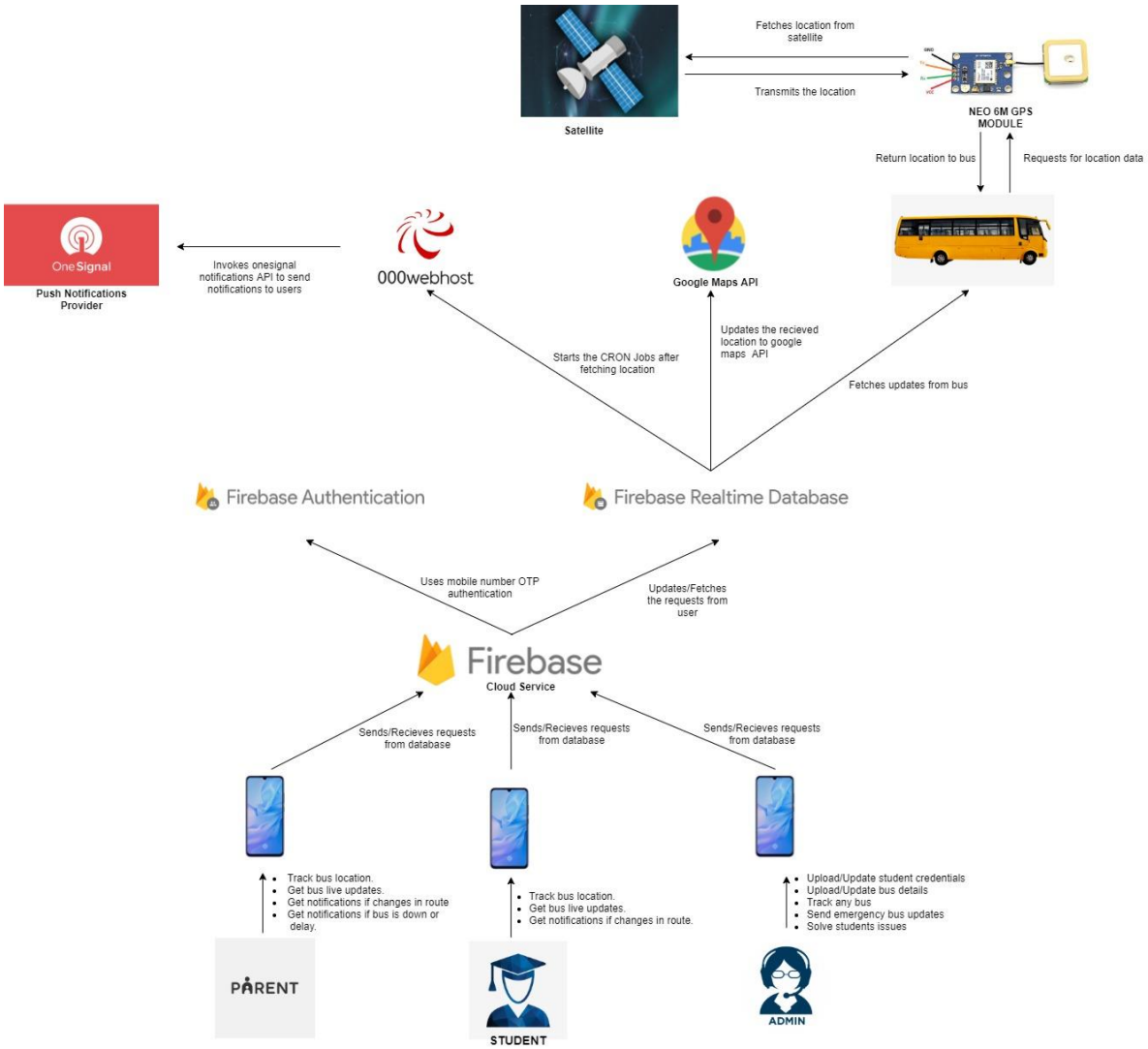


Figure 3.2 System Architecture

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.

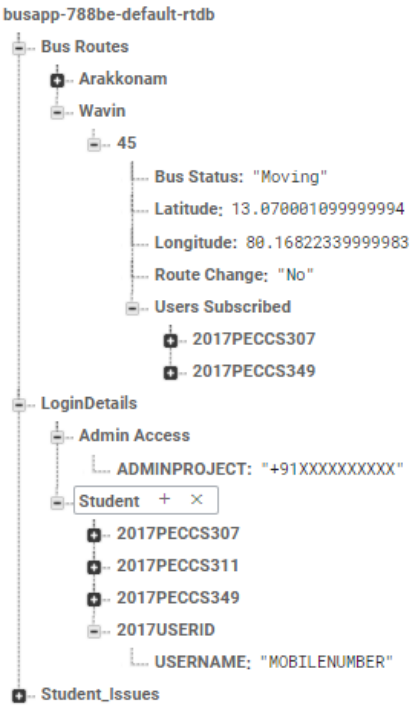


Figure 3.3 Firebase Architecture

IV. WORKING PROCEDURE

4.1 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.

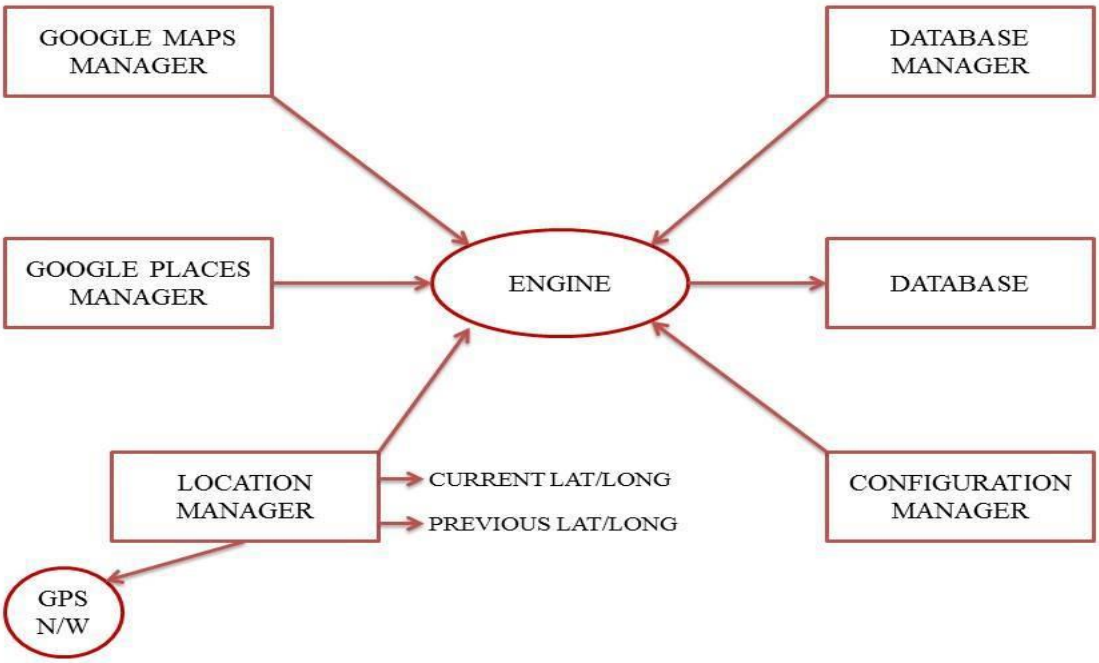


Figure 4.1 Working of a Location Based Service (LBS) Component (Internet Source)

4.2 IMPLEMENTATION

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 be 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.

V. 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.

CODING

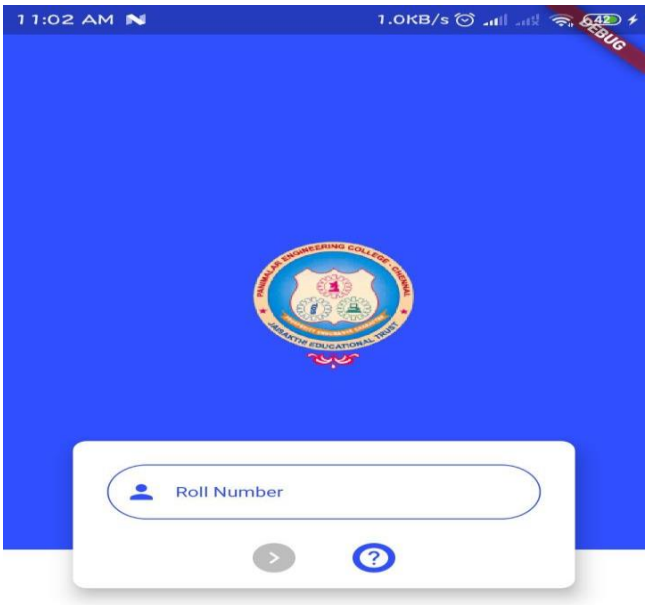
```
import 'dart:ffi';
import 'dart:typed_data';
import 'dart:async';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:shared_preferences/shared_preferences.dart';

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 */
```

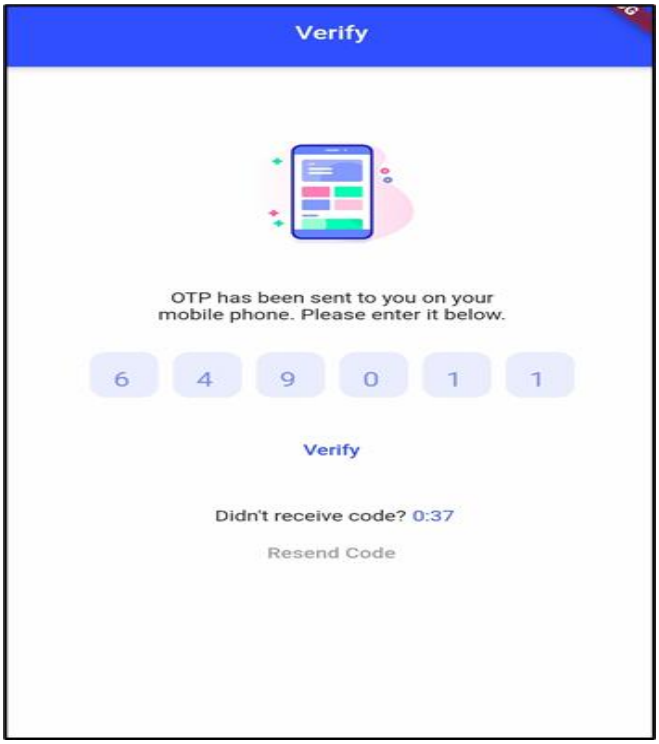
```
void getValues() async {

   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();
}
```

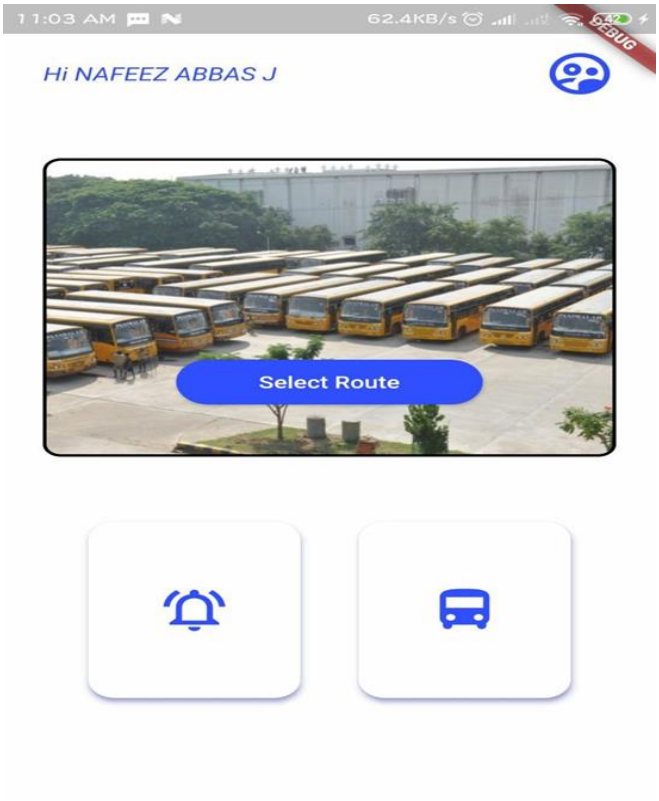
```
@override
Widget build(BuildContext context) {


return new Scaffold(
        /* The screen where we use google maps api to display the location of
        the 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,
```

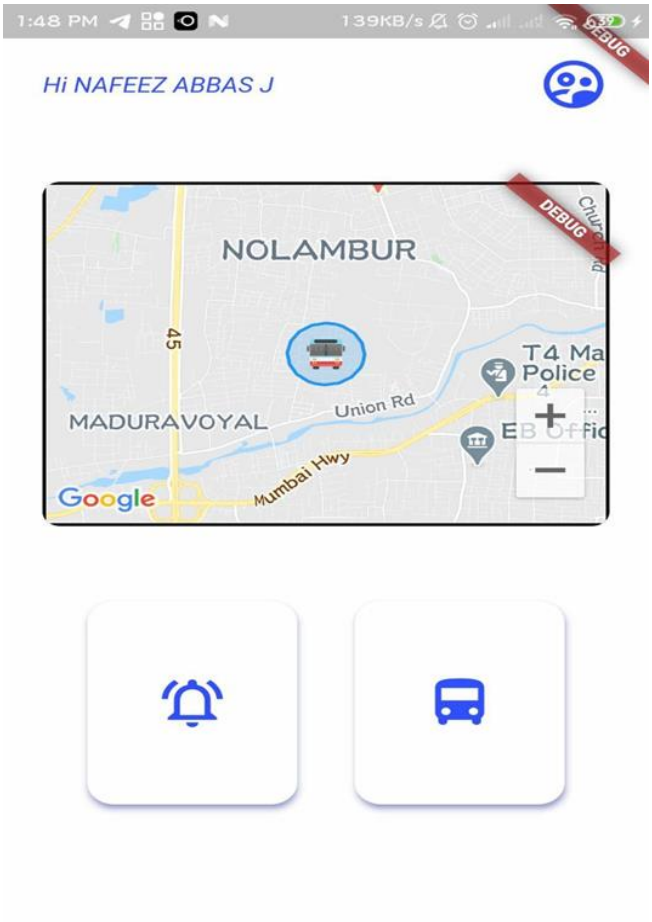
```
            center: location,
            fillColor: Colors.blue.withAlpha(50)
          );
        });
    }

    /* 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));
    }
}
```

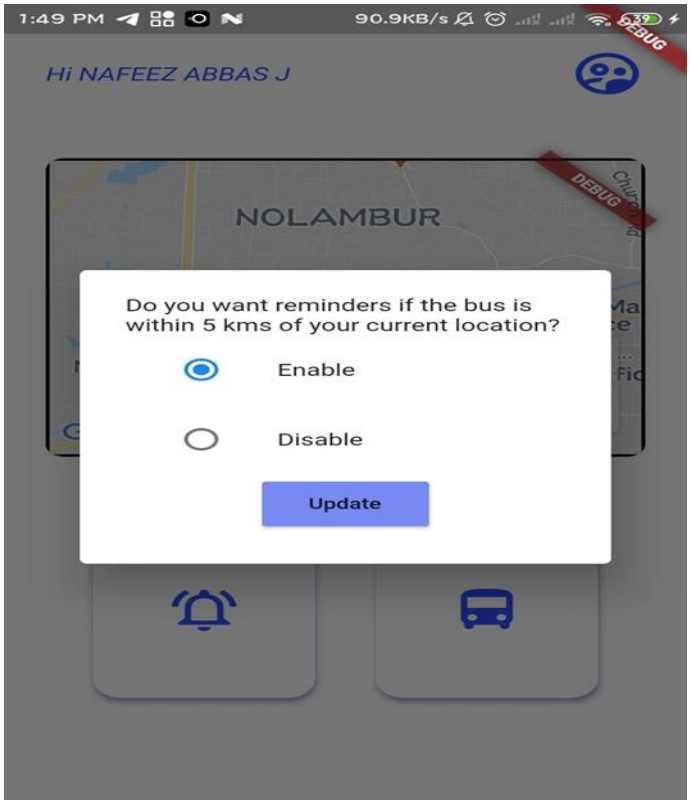
SCREENSHOTS

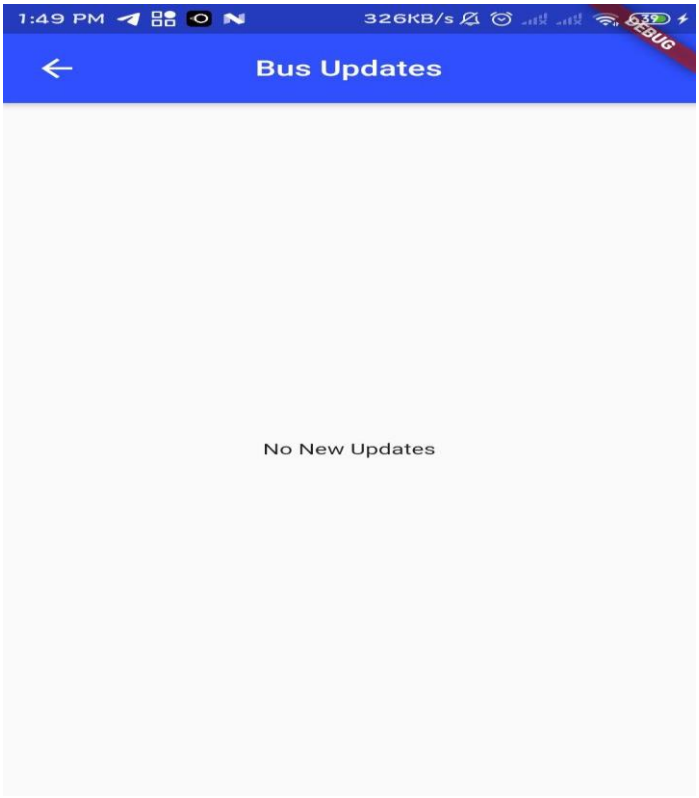





11:03 AM
62.4KB/s
DEBUG
Hi NAFEEZ ABBAS J
Select Route


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


1:49 PM
90.9KB/s
DEBUG
Hi NAFEEZ ABBAS J
NOLAMBUR
DEBUG
Do you want reminders if the bus is within 5 kms of your current location?
Enable
Disable
Update


1:49 PM
326KB/s
DEBUG
Bus Updates
No New Updates

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.