International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Smart Bus Tracking Application using IOT

Badghainya Anchal[1]*, Maripally Lokesh*[1]*, Mohd Saleem*[1]*, Umama Mahreen*[1]*, Ramesh Alladi*[2]

[1]Undergraduate students, Ace Engineering College,Ghatkesar,Hyderabad,Telangana,501301,India.

[2]Associate Professor, Ace Engineering College,Ghatkesar,Hyderabad,Telangana,501301,India.

ABSTRACT:

The objective of the paper is to build a mobile application for real-time bus tracking. The real-time tracking of the bus can be done and this information is then given to a remote user who wants to know the real-time bus information. The system provides the relevant information regarding real-time location, route details, driver’s contact number, average waiting time, and expected time to reach. The mobile application is developed using the flutter framework with the integration of firebase real-time database, which is connected to Node MCU.

Keywords:Mobile Application, Real Time Bus Tracking, Flutter, Firebase, Realtime Database, Node MCU.

Introduction

Cities frequently experience severe traffic problems and related commuter concerns, such as delays in bus arrival at bus stops and a lack of knowledge about various bus routes, stops, and schedules. Sometimes college employees and students will barely make it to the bus in such situations. This will cause a number of issues, such as being late for both classes and occasionally tests. We have put in place a smart bus tracking system to get around this problem. The college buses are tracked using this system.

The technology makes use of a smartphone app. Buses have Global Positioning System (GPS) tracking sensors installed in them with integrated IOT devices, and the smartphone app uses Mapbox API to show the location of the bus on a map. It displays the exact location of the bus on a map and updates the user's knowledge at various intervals. Additionally, the estimated arrival time is shown, which aids the user in determining precisely when the bus will arrive at their stop

literature survey

"Development of An Android Based Real Time Bus Tracking System" by Mohammad Nazmul Hasan and Md. Sharif Hossen [1], Everyone is rushing to get where they're going in this fast-paced world. Waiting for the bus in this situation is not an option. People who use public transit should be aware of where the bus they need is located. With the advancement of technology, Android smartphones have become global and affordable for all. Smartphones have ample potential to provide a rich user experience with interactive features.[2]The system in the bus uses Area Trace Algorithm and Haversine distance. Users will send an SMS in a specified format to a toll-free number and will receive a reply SMS with bus location, seat vacancy and estimated time of arrival.

System Architecture

"The first step is to create the front end of the application using the flutter framework. We need to set up the IoT with node MCU and GPS module and connect the IoT with the firebase real-time database. Now we need to connect our application to the firebase database. Now the application is integrated with firebase to fetch coordinates given by node MCU and using Mapbox API we need to display the maps and navigation in the application.

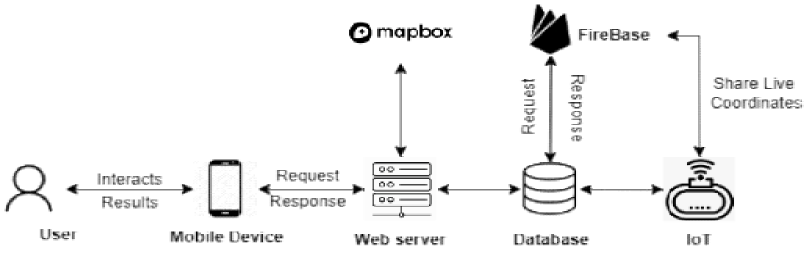


Figure 1. System Architecture

Technology Used

Flutter Framework

Flutter is a Google open source framework that allows you to create beautiful, natively built, multi-platform apps from a single codebase.
Flutter is made up of two main components:
An SDK (Software Development Kit) is a tool that allows you to create software. A set of tools to assist you in the development of your applications. Tools for compiling your code into native machine code are included (code for iOS and Android).
A Framework (Widget-based User Interface Library) is a collection of reusable user interface elements (buttons, text inputs, sliders, and so on) that you may customize to fit your needs.You'll utilize the Dart programming language to create Flutter apps. Google established the language in October 2011, but it has come a long way in the last several years.Dart is a front-end development language, and you'll be using it.

Mapbox API

Mapbox created and contributed to several open-source mapping projects, including the Mapbox GL-JS JavaScript library, MBTiles specification, TileMill cartography IDE, Leaflet JavaScript library, and CartoCSS map styling language and parser. Open data sources such as OpenStreetMap, Microsoft Open Maps, Wikidata, and data providers for specific geographic areas are used to compile the data. OpenStreetMap layers and features can be modified and added directly in OpenStreetMap, and your changes will eventually show up on your Mapbox map

Firebase Database

A toolkit called Firebase is designed to help you "create, improve, and expand your app," and the tools it provides you with cover a sizable amount of the services that developers would typically be required to construct themselves but don't really want to do because they'd rather concentrate on the app experience. Analytics, authentication, databases, configuration, file storage, push messaging, and a long list of other things fall under this category. The services scale with little to no development effort and are hosted in the cloud.

Arduino

Arduino is an open source hardware and software company, initiative, and user community that creates single-board microcontrollers and microcontroller kits for creating digital devices. In this case we are using an microcontroller called Node MCU which is programmable using arduino language.
Arduino programming language:
The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming.
Arduino code is written in C++ which includes special methods and functions used by the user based on the requirement.

Internet of Things

1. NodeMCU ESP8266

The NodeMCU ESP8266 development board comes with an ESP-12E module that includes an ESP8266 chip with a Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at adjustable clock frequencies from 80 MHz to 160 MHz. The NodeMCU has 128KB of RAM and 4MB of flash memory for storing data and programs. With built-in Wi-Fi / Bluetooth and deep sleep operation, its high computing power is ideal for IoT projects. The NodeMCU Dev Kit has an Arduino-like analog (ie A0) pin and a digital (D0-D8) pin on the board. Supports serial communication protocols. UART, SPI, I2C, etc. Using such a serial protocol, you can connect to serial devices such as I2C compatible LCD displays, HMC5883 magnetometers, MPU-6050 gyro meters + accelerometers, RTC chips, GPS modules, touch screen displays and SD cards.

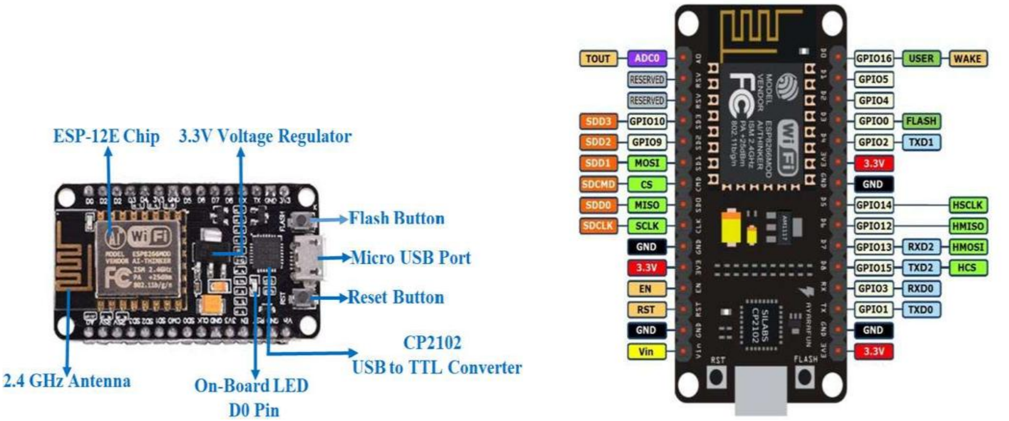


Figure 2. Node MCU pin description.

2. ***NEO-6MV2***

The NEO-6MV2 is a GPS (Global Positioning System) module and is used for navigation. The module simply checks its location on earth and provides output data which is the longitude and latitude of its position.NEO-6 modules are ideal for battery-operated mobile devices with severe cost and space requirements because of their tiny architecture, power, and memory options. Even in the most challenging circumstances, the NEO-6MV2's innovative design offers great navigation performance.

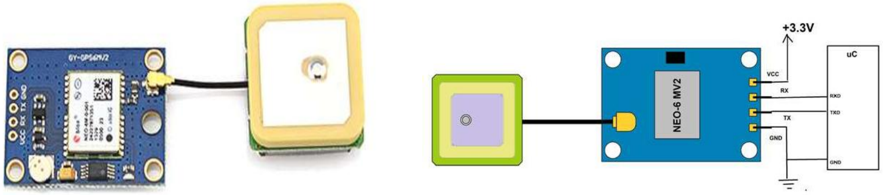


Figure 3. NEO-6MV2 and pin diagrams

Integration of All the Components.

All the components (Node MCU, GPS module, 16*2 LCD screen, potentiometers, resistor, jump wire and bread board in the below circuit are integrated with each other to make an IOT device which is connected to the internet with the help of WiFi module which is present in the NodeMCU micro processor.The purpose of this device is to fetch the live coordinates (Location) of the Bus using GPS module and to send these live coordinates to the Real time database (Firebase).This whole IoT device is installed in the Bus with 3-9 V power supply and internet source.

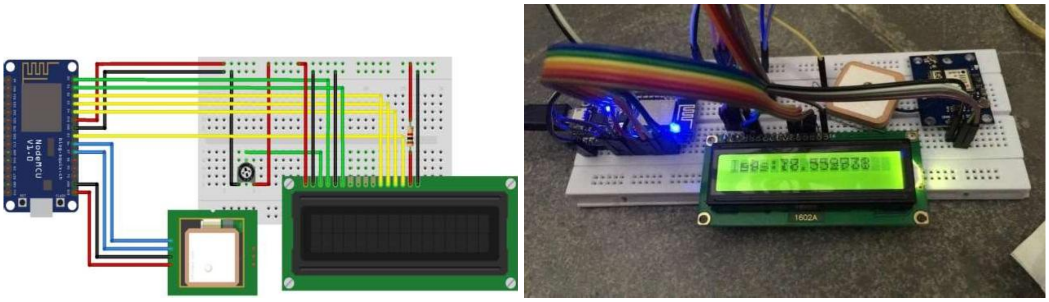


Figure 4. Circuit diagram of IoT device and real device

The above picture is the final complete Iot device that is integrated with all individual components which is connected to the internet and is in the working stage to find the live location coordinates of the current place and shows the coordinates in the display. The Iot device provides 6 responses of geo-coordinates of current location in a single second and that is reflected in the screen continuously and it directly uploads the same data in the real-time database firebase .

Results

When the IoT device gets power on it'll connect to the network/wifi and initialize the gps module and then the gps module will get coordinates to upload to the firebase real time database. Now to the application side the device fetch the coordinates from firebase realtime database, that will be passed to mapbox api to show in a map and then the application, based on both Bus tracker coordinates and the device coordinates it will calculate distance and time to arrive from bus location to user location.

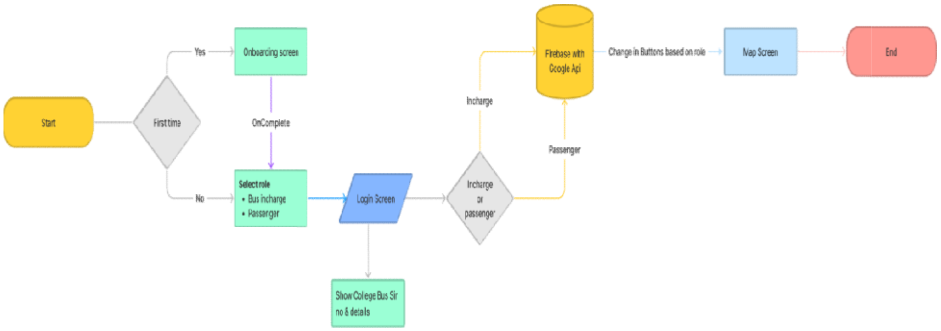
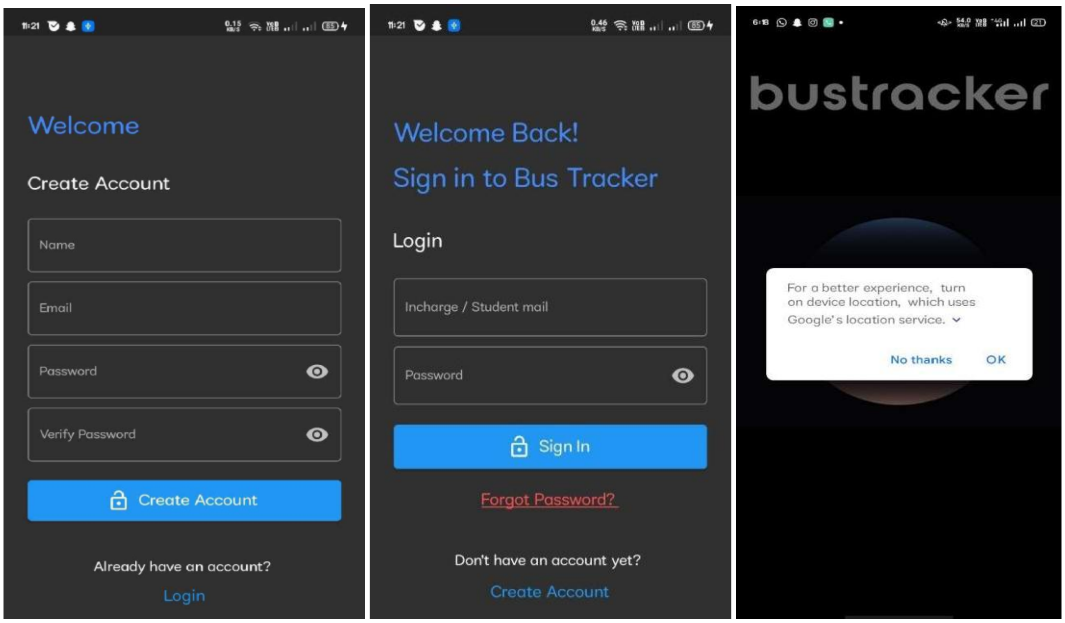


Figure 5. User flow

Screen-1: Login/Signup

When the user opens the mobile application for the first time, the user needs to sign up by filling in required details like username, E-mail, and password. If the user has an existing account, then he can login with his username and password.



For a better experience, turn on device location, which uses Google's location service.

Figure 6. Sign up and login page ,Location access page

Screen-2: Location access to application

After successful sign up/login of the user, the application needs the user's device location so that it is able to track distance between Iot device and user device, for that the application requests the device location which the user must accept.

Screen-3: Main screen

This is the main screen of the mobile application in which the user can access the real-time location of the bus and can see the movement in the map.In the screen you will find a section in the above part which has the critical information related to the journey,This widget shows result of the analysis of distance(in kilometer) left and eta(estimated time of arrival) of the bus to reach the boarding point of the student/user.

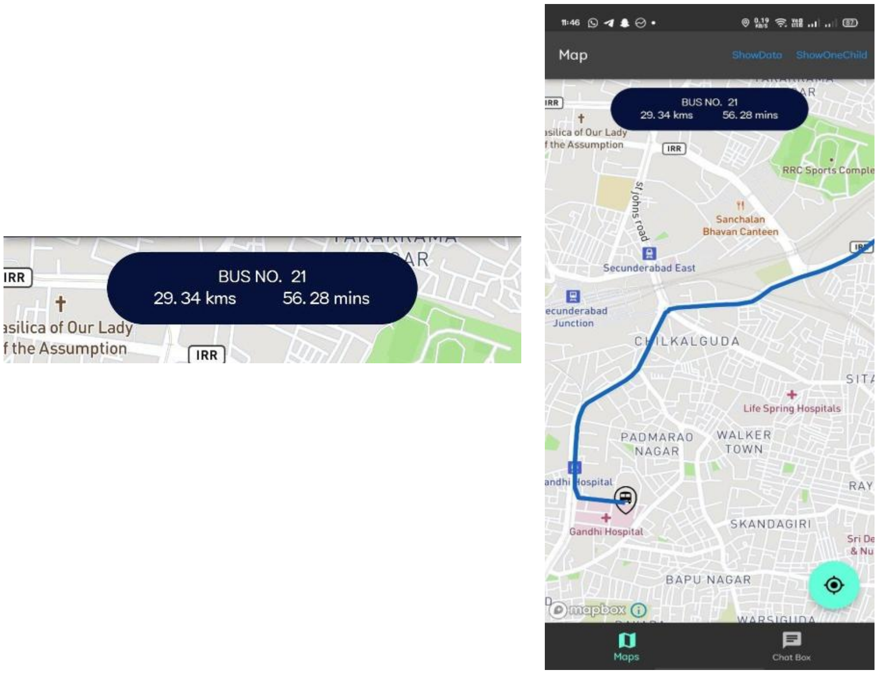
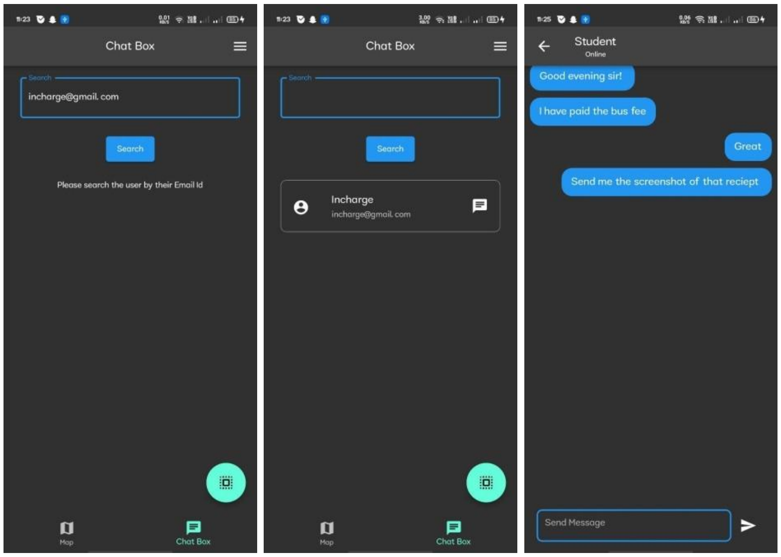


Figure 7. Widget showing Bus no, total distance and ETA and Home Screen of the application.

The mobile application also alerts the user or student by sending a notification when the bus is near the boarding point or when the estimation time of arrival is less than 15 minutes.

The application shows both locations of mobile phone and tracker in real time. By accessing both devices location the mapbox api can calculate distance and according to the speed tracker is moving we can also know the eta (estimated time of arrival).Also when a user is willing to send a message to the incharge, they can access the chat box and by entering their id they can send.



incharge@gmail.com

Please search the user by their Email Id

Incharge
incharge@gmail.com

Good evening sir!

I have paid the bus fee

Great

Send me the screenshot of that reciept

Figure 11. Chat box with search field and chat window

The above pictures show the chat box implementations.In this a user/student/incharge searches for a particular person by their email id and they can initiate their personal chat. A user can also view their recents chats in the chat log by clicking on the floating action button which is in the bottom right side of the chat screen.

Conclusion:

The main objective of this project is to help the student to get real-time location of the college bus using the map provided in our mobile application such that the student will have a high probability of boarding the bus on time to reach the college. The student/user will get a notification when the bus is near or 15 minutes away from the boarding point. The student/user can also use the chat box feature to post any question and get their queries resolved by the bus faculty.

As of now, the application supports location tracking, for getting real time bus location, and chat box for query resolving. In further enhancement each student/user record is stored in the database for further use by the bus faculty. The student record includes the status of fee payment, address of his home, emergency contact details in case of any emergency which will only be accessible to the bus Incharge.

Acknowledgment

We would like to thank our guide **Mr. A. Ramesh** and **Mrs. Soppari Kavitha** for their continuous support and guidance. Due to their guidance, we completed our project successfully. Also, we are extremely grateful to **Dr. M. V. VIJAYA SARADHI**, Head of the Department of Computer Science and Engineering, Ace Engineering College for his support and valuable time.

References:

[1]. Mohammad Nazmul Hasan; Md. Sharif Hossen, "Development of An Android Based Real Time Bus Tracking System", 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) 10.1109/ICASERT.2019.8934621

[2] Mr Darshan Ingle "Experimental Estimates of Low-Cost Bus Tracking System Using Area-Trace Algorithm" 2015 Fifth International Conference on Communication Systems and Network Technologies.

[3] Flutter documentation - https://docs.flutter.dev/

[4] Map box documentation - https://docs.mapbox.com

[5] Node MCU documentation - https://nodemcu.readthedocs.io/en/release/

[6] Arudino programming IDE documentation - https://www.circuito.io/blog/arduino-code/

[7] Firebase documentation - https://firebase.google.com/docs