Bus Tracking System using GPS

S.A. Bhavani
Department of Computer Science & Engineering
Anil Neerukonda Institute of Technology & Sciences
Andhra Pradesh, India

J. Phani Kumar
Department of Computer Science & Engineering
Anil Neerukonda Institute of Technology & Sciences
Andhra Pradesh, India

N. Christina Preethi
Department of Computer Science & Engineering
Anil Neerukonda Institute of Technology & Sciences
Andhra Pradesh, India

M. Lohith
Department of Computer Science & Engineering
Anil Neerukonda Institute of Technology & Sciences
Andhra Pradesh, India

G. Sanjusha
Department of Computer Science & Engineering
Anil Neerukonda Institute of Technology & Sciences
Andhra Pradesh, India

ABSTRACT
The main goal is to create an application that tracks the bus's dynamic location using a GPS-enabled device that retrieves the current coordinates of the bus as well as the timestamp. These coordinates are then saved into a database. The operation of retrieving and plotting these stored coordinates will be performed by a client process. This allows the users to locate the bus on the map.

The developed system will use a producer-consumer model, with the GPS equipped device fetching live coordinates acting as the producer and the application plotting these received coordinates on a map acting as the consumer. The platform that is created uses API calls which has an advantage to be integrated with other applications. A web-application which plots the dynamic location of the bus on the map.

This design uses low-latency, near-real-time data that allows the proposed system to track coordinates in very short periods (sub-second), improving the tracking system's overall precision. As this system employs a NoSQL-like database it is based on

BASE characteristics. This permits data values to alter over time due to its Soft State characteristic. As a result, the data might be updated amidst hundreds of data reads, resulting in reduced latency and near-real-time results.

This paper proposes the web application architecture for tracking our college bus service which lets the students and faculty to track the bus with ease.

General Terms
Bus Tracking System

Keywords
Global Positioning System (GPS), Raspberry Pi4, Amazon Elasticsearch, OpenStreetMap, Leaflet Library

1. INTRODUCTION
The developed system uses a producer-consumer model. The producer is modeled with the GPS equipped device which is connected to the Raspberry Pi and placed in the bus which communicates with the available satellites and fetches live coordinates and stores into the database. While the consumer is

2. EXISTING SYSTEM
The rise of technology in public transport is impending, but above all, the structure of the bus network structure and intelligent bus monitoring system must be established. This research presents a cloud-based bus tracking system based on IoT for reducing waiting time, human intervention and energy consumption. To provide a better and more efficient bus service, the actual location and arrival time of the bus can be monitored dynamically using a smartphone application. Passengers may also purchase tickets without queueing and reserve available seats through online payments. In terms of time loss, the suggested approach offers greater flexibility and client satisfaction. The primary objective is to reduce passengers unnecessary delays and uncertainty with respect to wait times.The existing IoT-based bus tracking system can contribute to ensure the long-term sustainability of public transportation services by providing significant advantages to passengers.[1]

3. PROPOSED SYSTEM
The proposed system shall contain two modules, the producer and the consumer. Where the GPS enabled device that fetches the live coordinates acts as the producer and the application that plots these fetched coordinates on the map acts as the consumer. The system shall use modern tools and technologies which results in tracking the coordinates in very small intervals (sub-second) which in turn improves the precision of the tracking system as a whole.

3.1 Functionalities of Proposed System
- Bus Selection: This application lets you track the location of the required bus from a pool of available buses.
- Real-time tracking: Using the Global Positioning System, you may track any specific bus in real time (GPS). Also, the NoSQL like database allows alteration of data values which results in near-real-time tracking.
- Route Tracing: This feature allows you to know the route the bus has traveled. It also acts as a safety concern as parents can also find the route of the bus.
- API Integration: The system’s API calls allow any application to make use of these API calls.

4. ARCHITECTURE DESIGN

The entire system is based on two modules: the producer and the consumer which are interconnected with the Amazon ElasticSearch Database. (Fig.2)

4.1 Producer Process

The main functionality of this module is connecting and fetching. The components of this module are GPS Module and Raspberry Pi. The Raspberry Pi is connected to the GPS module with the Vcc, Gnd, Transmitter and Receiver Pins respectively. This gives the power supply to the GPS module which in turn starts searching for satellites. It requires a minimum of four satellites to find the latitude and longitude. Thus, the connected GPS module collects the latitude and longitudes.The Raspberry Pi is configured in such a way that it receives the data from the GPS module. Using the "sudo raspi-config" command we need to enable the serial interface port of the Raspberry Pi [2]. This is enabled to receive the information from the GPS Module. The information is received at serial port zero of Raspberry Pi. The received data is the NMEA statements (Fig 3). These can be viewed by using the command "cat /dev/serial0".

The NMEA sentences contain information like timestamp, latitude, longitude, speed, datestamp, no. of satellites available, no. of satellites used for fetching data etc. [5]

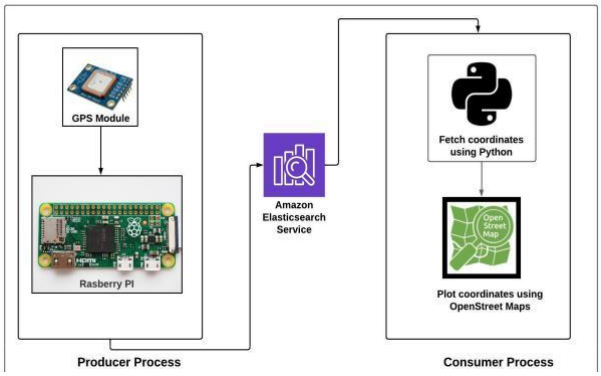

Figure 1. Architecture Design

From the available NMEA logs, we need to find the 'GPRMC' sentence as it contains the time, date, position, speed data provided by the GPS navigation receiver. These parameters are useful in our application.

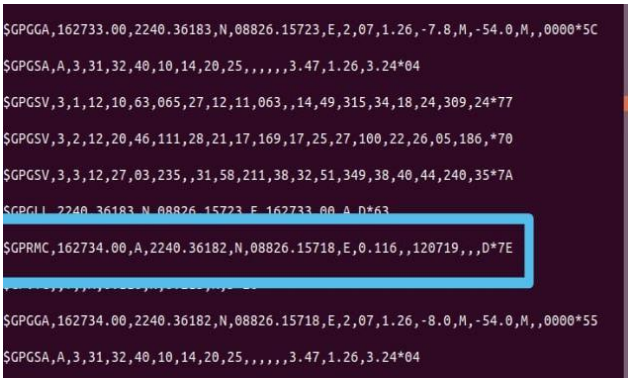

Figure 2. NMEA Sentences

Now, the Raspberry Pi contains a python code which parses through these NMEA sentences and fetches the required information in very short intervals. The information includes the latitudes, the longitudes and the timestamp. These are then immediately updated and stored into the database.

4.2 Consumer Process

The consumer process is a web-application. The main functionality of consumer application is to fetch the stored data from the database and plot the coordinates on the map. The main components in this process are the software modules. The users of this application can select a bus whose tracking information they want to have(Fig 4). It is programmed in such a way that the lastly stored coordinates are fetched along with timestamp. The fetched data is then plotted on the map which is the user interface. Along with plotting the current location, this application also plots the previously traveled locations which shows the route from the starting point to the destination[3].

API to store information. An API call occurs when a consumer process sends a request to an API, which then retrieves the desired data from the OpenSearch Database and returns it to the client.

4.3 Web-application

On the other hand, the software application is the only consumer for the APIs. It is developed using Flask, a python framework for backend and JavaScript's Leaflet Library integrated with OpenStreetMaps for the frontend. Leaflet is a JavaScript package that is used to create web mapping applications.

5. CONNECTIONS OF WORKING MODEL

Initially to configure the Raspberry Pi, we need external input devices (keyboard and mouse) and output devices(monitor). To connect the device to peripherals we need a set of interconnecting cables.

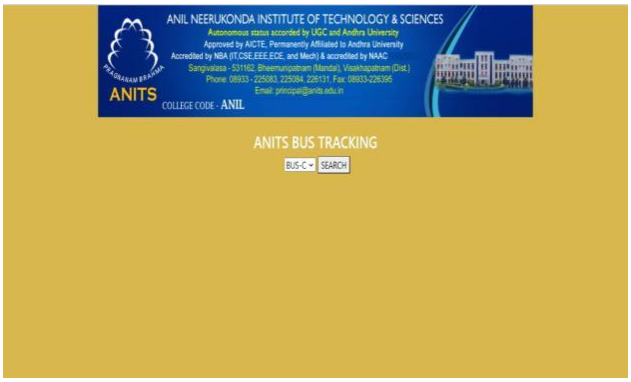

Figure 3. Web Application

6. DEVELOPING TOOLS AND TECHNOLOGIES

This application has been developed using modern hardware tools and software technologies that give the best and accurate information.

6.1 GPS Module

This application was built using a Neo 6M GPS Module which is economical and most suitable. The Global Positioning System (GPS) is a satellite-based navigation system. This device employs cutting-edge technology in order to provide the most accurate positional data available.

6.2 Raspberry Pi

The GPS Module is integrated with the Raspberry Pi 4 Model B which is a single-board computer with high computational capacity. The Raspberry Pi 4 Model B was released in June 2019, features a 1.5 GHz quad-core ARM Cortex-A72 processor, on-board 802.11ac Wi-Fi, Bluetooth 5, full gigabit Ethernet, two USB 2.0 connections, two USB 3.0 ports, 1–8 GB of RAM, and dual-monitor support for up to 4K resolution.

6.3 Power Supply

To power up the Raspberry Pi, a power supply of 3A and USB-C type is required. For this system we have used a Mi power bank of 20000 mAh with the required specifications as it is portable..

6.4 API

The API is driven on top of the Flask framework based on Python Language. Amazon ElasticSearch is used for database

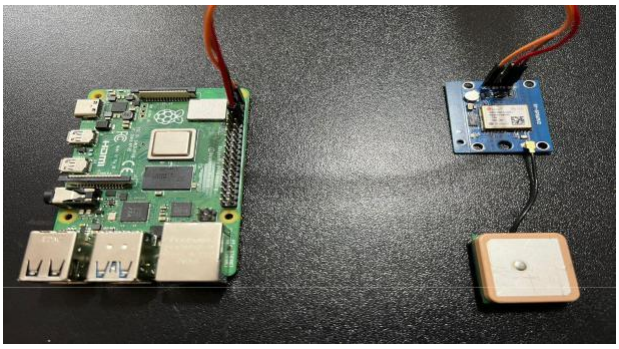

Figure 4. Connections

A HDMI to Micro HDMI cable is required to connect it to the monitor. A mouse and keyboard that are USB/Bluetooth compatible can be used as input devices. A microSD card is essential for file storage and the Raspberry Pi OS. To power up the Raspberry Pi, USB-C type power supply with 3A is necessary. (Fig 5).

After configuration is completed, we need a set of Jumper Cables to connect the GPS Module's VCC, Gnd, Rx, Tx to the respective pins of the Raspberry Pi.[4]

7. OUTPUT

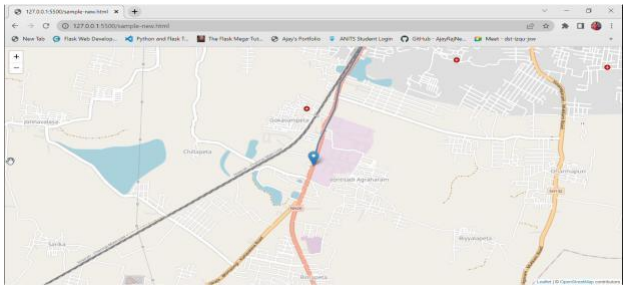

Figure 5. Output

The output is the coordinates that are depicted on the map along with the representation of the route which gives the user an idea of the movements of the bus.(Fig 6).

8. CONCLUSION

Our college's staff and students benefit from the proposed system since it allows them to maintain track of the present location of the desired bus and also allows them to check the route which the bus has traveled from the starting point. The developed system has the potential to deliver a highly ccurate, near real-time tracking service that is both efficient and user-friendly.

9. ACKNOWLEDGMENT

Our thanks to our guide who have contributed towards the completion of our project

10. REFERENCES

[1] https://ieeexplore-ieee-org-anits.knimbus.com/document/8952037

[2] https://maker.pro/raspberry-pi/tutorial/how-to-use-a-gps-receiver-with-raspberry-pi-4

[3] https://leafletjs.com/reference.html

[4] https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/1

[5] http://aprs.gids.nl/nmea/