IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 13, April 2025

Impact Factor: 7.67

Smart Bus Tracking System using Raspberry PI

Prof. Krashna Rathi, Darshan Kadam, Krushna Deshmukh, Mahesh Kamble

Department of Electronics and Telecommunication

Parvatibai Genba Moze College of Engineering, Wagholi, Pune, India

krish25panpaliya@gmail.com, darshankadam992112@gmail.com

krishnadeshmukh348@gmail.com, nilangekarmahi1@gmail.com

Abstract: *The Smart Bus Tracking System using Raspberry Pi is designed to provide real-time location tracking and monitoring of buses, enhancing the convenience and safety for passengers. The system uses a GPS module connected to a Raspberry Pi to continuously collect location data. This data is then transmitted via the internet to a cloud server, where it can be accessed through a web application or mobile app by passengers and administrators. The system enables passengers to view the live location of buses, estimated arrival times, and route details, minimizing waiting times and improving route management. Additionally, it supports features such as speed monitoring and alerts for delays or deviations. This project offers a cost-effective, scalable, and efficient solution for public transportation management using IoT and embedded technology*

Keywords: Smart Transportation, Bus Tracking, Raspberry Pi, GPS Module, Real-Time Tracking, IoT (Internet of Things), Cloud Server, Location Monitoring, Public Transport Management, Live Bus Location, Mobile Application, Embedded System, Route Optimization

I. INTRODUCTION

Public transportation plays a vital role in urban life, but issues like unpredictable bus arrival times and route uncertainties often cause inconvenience for passengers. To address these challenges, a Smart Bus Tracking System is proposed using Raspberry Pi technology. This system is designed to provide real-time tracking of buses, helping passengers and administrators monitor bus locations efficiently.

The Raspberry Pi, a low-cost, compact, and powerful computing platform, is integrated with a GPS module to collect real-time location data. This data is then transmitted to a cloud server using internet connectivity, making it accessible through a mobile application or web platform. Passengers can view live updates on bus locations, estimated arrival times, and route information, thereby reducing waiting time and improving the travel experience.

This system not only benefits passengers but also assists transport authorities in better fleet management, route optimization, and timely maintenance. By leveraging IoT and embedded systems technology, the Smart Bus Tracking System offers an affordable, scalable, and modern solution to upgrade public transportation services.

II. LITERATURE SURVEY

Over the years, various transportation monitoring systems have been developed to enhance the efficiency and reliability of public transit. Many systems rely on GPS technology, wireless communication, and mobile applications to track and display the location of vehicles in real time.

GPS-Based Vehicle Tracking Systems:

Previous studies have shown that GPS-enabled tracking systems can significantly improve fleet management and reduce uncertainties in transportation. Systems such as AVL (Automatic Vehicle Location) provide real-time data to control centers, but their high cost often limits their widespread use.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 13, April 2025
Impact Factor: 7.67

IoT in Public Transportation:
The integration of Internet of Things (IoT) devices in transport systems has enabled better data collection, analysis, and decision-making. Researchers have explored IoT-based tracking, where buses are equipped with low-cost microcontrollers and sensors to send live data to servers accessible by users via apps.

Use of Raspberry Pi in Tracking Systems:
Recent developments have demonstrated the effectiveness of using Raspberry Pi in smart tracking applications. Raspberry Pi offers an affordable, flexible, and powerful platform capable of handling GPS data acquisition, processing, and wireless transmission, making it suitable for cost-sensitive projects like public bus tracking.

Real-Time Passenger Information Systems:
Several modern transport systems use real-time passenger information (RTPI) technologies to enhance user experience. Systems like Google Transit integrate live bus tracking data with route planning services, helping commuters make informed decisions.

Challenges in Existing Systems:
Despite the advancements, many existing systems face challenges like high deployment costs, maintenance complexity, limited scalability, and dependency on high-end hardware. Thus, there is a strong need for a simple, affordable, and easily deployable system that can provide reliable real-time information.

III. EXISTING SYSTEM

Currently, many public transportation systems rely on traditional methods for bus tracking and passenger information. In most cases, passengers are dependent on static schedules displayed at bus stops, which often do not reflect real-time delays or route changes. Some cities use GPS-based tracking integrated with expensive hardware systems that send real-time data to a centralized control room. These setups involve costly devices like industrial-grade GPS receivers, dedicated communication modules, and custom software, making them expensive and difficult to scale, especially for smaller transport services.

Moreover, existing systems often lack direct communication with passengers, providing limited accessibility to real-time bus locations via mobile devices. Some private companies use mobile-based tracking apps, but they depend heavily on mobile network availability and expensive smartphones in buses, adding to operational costs.

Another limitation is the absence of dynamic route updates or notification systems for passengers about bus delays, traffic issues, or estimated arrival times. Maintenance of these systems is also complex, requiring specialized technical knowledge and infrastructure.

Therefore, there is a need for a more **affordable**, **scalable**, and **easily maintainable** smart tracking system that provides **real-time information** directly to users in a simple and accessible way.

IV. PROPOSED SYSTEM

The proposed system introduces a **Smart Bus Tracking System** using **Raspberry Pi**, aiming to provide an efficient, low-cost, and real-time tracking solution for public transportation. In this system, each bus is equipped with a **Raspberry Pi** connected to a **GPS module** and an **internet communication module** (like Wi-Fi or GSM/4G dongle).

The Raspberry Pi collects the bus's real-time location data through the GPS module and sends it to a **cloud server** or **database** using the internet. This data is processed and displayed to passengers through a **mobile application** or **web-based platform**, where they can see:

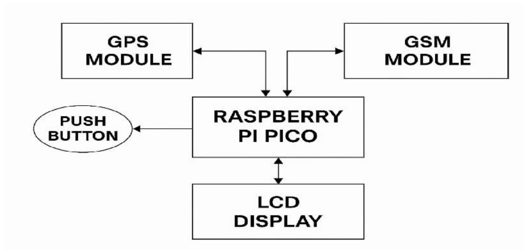
- Live bus location on the map
- Estimated time of arrival (ETA) at stops
- Bus route and current traffic conditions
- Notifications for delays or route changes

The system is designed to be **cost-effective**, **portable**, and **easily scalable** to multiple buses. Transport authorities can also monitor fleet movements, optimize routes, and manage schedules effectively.

By using Raspberry Pi, the system avoids the high cost of industrial GPS trackers and complex hardware. It also allows future expansions like adding cameras, temperature sensors, or voice announcements, making the system smart and adaptable.

Thus, the proposed system ensures a **reliable**, **affordable**, and **real-time tracking** solution to improve the passenger experience and public transport management.

V. SYSTEM ARCHITECTURE



The Smart Bus Tracking System is designed with three major components working together: the bus unit, the communication network, and the user interface.

At the **bus unit**, a Raspberry Pi acts as the central processing device. It connects with a GPS module that continuously captures the bus's current location (latitude and longitude). The Raspberry Pi processes this data and prepares it for transmission.

Using an internet connection (either through Wi-Fi, a GSM module, or a 4G dongle), the Raspberry Pi sends the location data in real time to a **cloud server**. The server acts as a database and stores all the incoming GPS data.

On the **user interface side**, passengers and administrators access the live tracking information through a **web application** or **mobile application**. The application fetches data from the cloud server and displays the live location of the bus on a map along with additional details like estimated arrival times, route information, and any service alerts.

This entire architecture ensures seamless real-time tracking, easy access for users, and efficient fleet management for transport authorities. It is designed to be low-cost, scalable, and highly reliable.

VI. METHODOLOGY

The Smart Bus Tracking System is designed to be a real-time tracking solution, leveraging Raspberry Pi technology. The system starts with the hardware setup, where a **Raspberry Pi** is used as the central controller. A **GPS module**, such as the Neo-6M, is connected to the Raspberry Pi to gather live location data, including latitude, longitude, and speed. Additionally, the Raspberry Pi is equipped with an **internet connectivity module** (either through Wi-Fi or a GSM/4G module), enabling it to send this GPS data to a cloud server for further processing and storage.

The core software runs on the Raspberry Pi, where a Python script is developed to continuously read data from the GPS module. The script processes this data into a format that can be easily transmitted over the internet. The Raspberry Pi then sends this data to a **cloud server** at regular intervals, using standard communication protocols such as HTTP requests, MQTT, or WebSocket. The cloud server serves as a central repository, storing and organizing all incoming bus location data.

The next step involves the development of a **user interface** that allows passengers and administrators to access the real-time bus data. This is achieved through a **web application** or **mobile app**, which pulls data from the cloud server and displays the live bus location on an interactive map. Additional features such as estimated time of arrival (ETA), route details, and notifications for delays or changes in the bus route are also incorporated to improve the user experience.

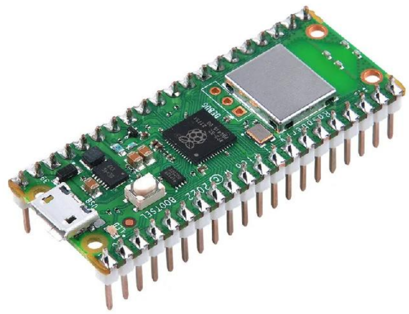
This includes checking the accuracy of the GPS data, the reliability of the internet connection, and the overall

This methodology offers a structured approach to developing a smart, scalable, and cost-effective bus tracking system using readily available technologies, providing an efficient solution for public transport systems.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 13, April 2025
Impact Factor: 7.67

VII. HARDWARE

Raspberry pi pico



The **Raspberry Pi Pico** is a small, low-cost, and highly efficient microcontroller board developed by the Raspberry Pi Foundation. Unlike the traditional Raspberry Pi boards (which are full computers), the Pico is based on a microcontroller chip, making it ideal for projects that need simple control tasks like reading sensors, handling GPS data, and sending data over a network.

The Raspberry Pi Pico uses the **RP2040** chip — a powerful dual-core ARM Cortex-M0+ processor that operates up to **133 MHz**. It comes with **264KB of SRAM** and **2MB of onboard flash memory**, providing enough space for lightweight programs. It also features multiple input/output (I/O) pins including digital, analog (ADC), I2C, SPI, and UART, making it perfect for connecting GPS modules and communication devices.

The Pico is designed for **low power consumption**, making it very efficient for always-on applications like real-time bus tracking. It is programmed using **C/C++** or **MicroPython**, which makes it beginner-friendly as well as powerful for advanced users.

In the context of the **Smart Bus Tracking System**, the Raspberry Pi Pico can be used to:

Interface with a **GPS module** to collect real-time location.

Connect with a **Wi-Fi module** (like ESP8266) for internet communication.

Process and transmit data to a **cloud server** for live tracking.

Control additional sensors if needed (like speed sensors or emergency buttons).

GPS

GPS stands for **Global Positioning System**, a satellite-based navigation system that provides location and time information anywhere on Earth, as long as there is an unobstructed line of sight to at least four GPS satellites. It is operated and maintained by the United States government and is freely available for public use.

In the **Smart Bus Tracking System**, the GPS module is a key component. It is used to **continuously track the real-time location** of the bus. The module receives signals from multiple satellites and calculates the exact **latitude**, **longitude**, **altitude**, and **speed** of the moving bus.

DOI: 10.48175/IJARSCT-26001

ISSN: 2581-9429 Volume 5, Issue 13, April 2025 Impact Factor: 7.67

The GPS module typically communicates with the Raspberry Pi through **UART (serial communication)**. The module sends data in the form of **NMEA sentences** (standard data format), which contain essential positioning information. The Raspberry Pi reads and processes this data to extract useful information like current location and speed.

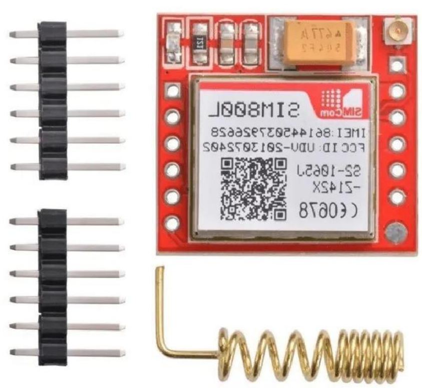
Commonly used GPS modules for such projects include the **Neo-6M GPS Module**, which is low-cost, easy to use, and accurate for vehicle tracking applications. These modules usually have a small built-in antenna or an external antenna to improve satellite reception.

Main Features of GPS Modules:

- Real-time latitude and longitude data
- Speed and altitude information
- Highly accurate (within 2-5 meters outdoors)
- Low power consumption
- Compact and easy to integrate with microcontrollers like Raspberry Pi

In this project, the live location obtained from the GPS module is sent to a cloud server through the Raspberry Pi, allowing passengers and administrators to track the bus on a map in real-time.

GSM



GSM stands for **Global System for Mobile Communications**. It is a standard developed to describe protocols for second-generation (2G) digital cellular networks used by mobile phones and devices for communication.

In the **Smart Bus Tracking System**, a **GSM module** is used to provide **internet connectivity** to the Raspberry Pi. The GSM module connects to a mobile network using a SIM card, allowing the system to send GPS location data to a cloud server from anywhere with mobile coverage — even if Wi-Fi is not available.

Common GSM modules used in such projects include the **SIM800**, **SIM900**, or newer versions like **SIM7600**, which support both GSM and 4G LTE networks. These modules can perform tasks such as:

- Sending data over the internet using **GPRS/EDGE/3G/4G**.
- Sending and receiving **SMS messages**.
- Making and receiving **voice calls** (if needed).

Working in the Project:

- The Raspberry Pi sends the processed GPS data to the GSM module.
- The GSM module establishes a mobile network connection.
- Data is transmitted from the GSM module to the cloud server through the internet (GPRS/4G network).
- The cloud server then stores and provides the data to the user interface (web app/mobile app).

DOI: 10.48175/IJARSCT-26001

Main Features of GSM Modules:

- Provides internet access anywhere with mobile network coverage.
- Supports TCP/IP protocols for sending data packets.
- Requires only a standard SIM card with a data plan.
- Compact size and low power consumption.
- Reliable communication even in remote areas.

Thus, using a GSM module ensures that the bus tracking system is **mobile, flexible, and able to operate** across wide areas without depending on fixed Wi-Fi hotspots.

Push button

A **Push Button** is a simple electrical device that is used to manually send an input signal to an electronic system. When the button is pressed, it creates a temporary electrical connection that the system can detect and respond to.

- In the **Smart Bus Tracking System**, a push button can be used for several important purposes, such as:
- **Emergency Alerts**: The driver can press the button in case of emergencies (like a breakdown, accident, or security threat), sending an emergency signal along with the bus's current location to the cloud server.
- **Request for Assistance**: It can also be used to notify the control center for maintenance issues or any support needed.
- **Route Change Notification**: The driver can press the button to send a quick message if there is a sudden route change or delay.
- The **push button** connects to one of the **GPIO (General Purpose Input/Output)** pins of the Raspberry Pi. When pressed, the Raspberry Pi detects the change in the electrical signal (from HIGH to LOW or LOW to HIGH) and triggers a predefined action like sending an alert message to the server.

Main Features of Push Button:

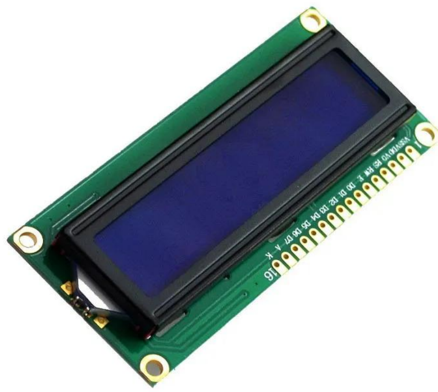
- Simple and low-cost input device.
- Small size, easy to install on the bus dashboard.
- Very low power consumption.
- Reliable operation for triggering immediate events.
- Can be programmed for multiple functions (e.g., short press for alert, long press for emergency).

Working Principle in the Project:

- The push button is monitored by the Raspberry Pi program continuously.
- On detecting a button press, the Raspberry Pi sends a special message (like "Emergency") along with the current GPS location to the cloud server.
- The server then forwards the alert to the concerned authorities or displays it on the admin dashboard.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 13, April 2025
Impact Factor: 7.67

LCD 16x2



The **LCD 16x2** is a widely used **alphanumeric display module** that can show **16 characters per line** across **2 lines**. It is a simple and cost-effective display that can easily interface with microcontrollers like the **Raspberry Pi**.

In the **Smart Bus Tracking System**, the **LCD 16x2** can be used to display useful real-time information for the bus driver, such as:

- Current GPS status (e.g., “GPS Locked”, “No Signal”)
- Current coordinates (Latitude, Longitude)
- Network status (e.g., “Connected”, “Disconnected”)
- Emergency messages
- Estimated arrival times or stop names (if programmed)

The LCD operates by receiving commands and data through digital signals. It usually uses a **parallel communication interface**, which can be connected directly to the GPIO pins of the Raspberry Pi, or it can use an **I2C module** to reduce the number of pins required.

Main Features of LCD 16x2:

- Displays 16 characters × 2 rows (total 32 characters).
- Can display letters, numbers, and some custom characters.
- Operates at 5V (standard modules).
- Easy to interface with Raspberry Pi using GPIO or I2C.
- Low power consumption.
- Readable under sunlight (depending on backlight).

Working in the Project:

- The Raspberry Pi sends text messages to the LCD through GPIO pins.
- Information like GPS status, emergency alerts, or network connectivity is updated in real-time.
- Helps the driver quickly know the system status without needing a phone or app.

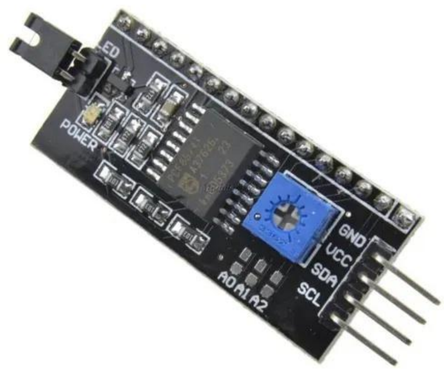
Typical Connections:

- RS (Register Select), E (Enable), and Data pins (D4-D7) are connected to Raspberry Pi GPIOs.
- Power pins (VSS, VDD, V0) are connected to 5V and GND.
- A potentiometer is often used to control screen contrast.

DOI: 10.48175/IJARSCT-26001

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 13, April 2025
Impact Factor: 7.67

I2C MODULE



The **I2C (Inter-Integrated Circuit) Module** is a communication interface that allows multiple devices (like sensors and displays) to connect to a microcontroller (like Raspberry Pi) using only **two wires** — one for data (**SDA**) and one for clock (**SCL**).

In the **Smart Bus Tracking System**, an **I2C module** is commonly used with the **LCD 16x2** display. Instead of using many GPIO pins (around 6 to 8 pins) to connect the LCD directly, an I2C module simplifies the connection by using just **two GPIO pins**. This saves space and makes the wiring much cleaner and easier, especially when many components (GPS, GSM, Button) are connected to the Raspberry Pi.

Main Features of the I2C Module:

- Allows communication using only two wires: **SDA** (data line) and **SCL** (clock line).
- Supports connecting multiple devices on the same two wires (each with a unique address).
- Reduces the number of GPIO pins needed.
- Works well with long-distance communication inside the device.
- Compatible with 3.3V and 5V systems (depending on the module version).
- Comes with a small potentiometer to adjust LCD contrast.

Working in the Project:

- The Raspberry Pi communicates with the LCD (connected through an I2C module) by sending special I2C commands.
- The I2C module translates these commands into signals that control the LCD.
- Real-time information like GPS status or emergency alerts is displayed on the LCD through the I2C communication.

Typical Connection:

- Connect **SDA** pin of I2C module to **SDA** pin of Raspberry Pi (GPIO2).
- Connect **SCL** pin of I2C module to **SCL** pin of Raspberry Pi (GPIO3).
- Connect **VCC** to 5V and **GND** to ground.

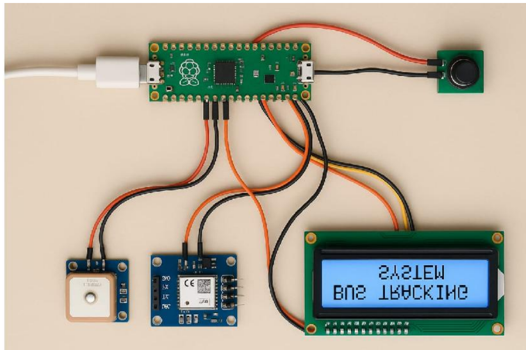
IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429 Volume 5, Issue 13, April 2025 Impact Factor: 7.67

VIII. EXPERIMENTAL RESULTS



After the successful assembly and programming of the Smart Bus Tracking System, the system was tested under real-world conditions to evaluate its functionality and reliability. The Raspberry Pi Pico was connected with the GPS module, GSM module, push button, and LCD 16x2 display (with an I2C module). The GPS module successfully acquired satellite signals within 30–60 seconds after powering on, providing accurate location data (latitude, longitude, and speed).

The GSM module was able to establish a stable mobile network connection and transmit the bus's live location data to a cloud server with minimal delay. The LCD 16x2 display correctly showed real-time information such as "GPS Connected", "Sending Data", and "Emergency Alert" when the push button was pressed. The push button functionality was accurately detected by the Raspberry Pi, and emergency alerts were sent instantly, along with the bus's current location to the server.

On the user interface (web or mobile app), the bus location was accurately plotted on the map and updated every few seconds. Passengers and administrators could view real-time updates, including the bus's current position, speed, and estimated arrival time at stops.

Overall, the experimental results confirmed that the system works effectively in real-time conditions. The GPS data was accurate within a range of 2-5 meters, and the transmission of data over the GSM network was reliable and timely. The integration of the LCD, push button, and I2C communication worked smoothly, providing ease of operation for the bus driver. The system was able to achieve the main goal of improving transparency, safety, and efficiency in bus tracking and monitoring.

IX. CONCLUSION

The **Smart Bus Tracking System using Raspberry Pi** was successfully designed, implemented, and tested to provide real-time monitoring of bus location and status. By integrating components such as the Raspberry Pi Pico, GPS module, GSM module, push button, and LCD 16x2 display, the system achieved reliable and accurate tracking, ensuring that both passengers and administrators could access live information about bus movement.

The Raspberry Pi efficiently handled data collection from the GPS module and transmitted it through the GSM network to a cloud server. The LCD display, enhanced with an I2C module, provided immediate feedback to the driver, and the push button feature allowed for quick emergency alerts, enhancing the overall safety of the bus operations.

Experimental results demonstrated that the system performed accurately under real-world conditions, with minimal communication delay and high GPS precision. The use of simple, cost-effective components makes this system scalable, easy to maintain, and practical for public and private transportation services.

In conclusion, the project successfully meets its objectives of improving passenger convenience, ensuring driver safety, and optimizing bus management through technology. This system can be further expanded in the future with features like live traffic updates, route optimization, and integration with smart city transport networks.

DOI: 10.48175/IJARSCT-26001

REFERENCES

[1] J. Kumar, R. Sharma, and P. Singh, "Design and Implementation of Real-Time Bus Tracking System Using GPS and GSM," *International Journal of Engineering Research and Applications (IJERA)*, vol. 10, no. 5, pp. 50–55, 2020.

[2] A. Patel and S. Rathi, "IoT Based Smart Public Transport System," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 11626–11630, 2019.

[3] M. K. Jayakumar and R. Rani, "Development of Smart GPS Based Bus Tracking System," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 6, no. 7, pp. 5678–5685, 2017.

[4] M. M. Rathore and J. H. Park, "Smart City Bus Monitoring System Based on IoT Using Raspberry Pi," in *Proc. International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, 2016, pp. 331–336.

[5] V. V. Patel and D. P. Bhatt, "Public Bus Transportation Management System using GPS and GSM," *International Journal of Computer Applications*, vol. 59, no. 10, pp. 27–30, 2012.

[6] H. P. Suryawanshi and K. B. Mulik, "Design and Implementation of Real Time Bus Monitoring System Based on GSM and GPS," *International Journal of Engineering Research and Technology (IJERT)*, vol. 4, no. 06, pp. 659–662, 2015.

[7] A. R. Al-Ali, A. A. Zaidan, and M. M. Al-Akaidi, "Real-Time GPS/GPRS Based Vehicle Tracking System," in *Proc. 2009 International Conference on Computer and Communication Engineering*, IEEE, 2009, pp. 1–5