International Journal of Integrative Studies (IJIS)
Journal homepage:www.ijis.co.in

Track My Bus - A Real-time College Bus Tracking & Management System

Nehanjali Bangaru[1], Shivanjali Dumpala[2], Cheekatla Gayathri Devi[3], Abraham Bommothi [4], Yashwanth Reddy Bireddy[5], Dr. Padmaja Pulicherla[6]

[1]Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India, nehanjalibangaru@gmail.com
[2]Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India, shivanjalidumpala8@gmail.com
[3]Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India, cheekatlagayathridevi@gmail.com
[4]Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India, abraham.pandu333@gmail.com
[5]Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India, yash7k8298@gmail.com
[6]Professor and Head of the department AIML, Hyderabad Institute of Technology and, Management, Hyderabad, India, padmaja.j2ee@gmail.com

Abstract

Efficient transportation plays a crucial role in maintaining punctuality and safety within academic institutions. However, students and faculty often experience uncertainty regarding bus arrival times, resulting in delays and lost productivity. Track My Bus – A Real-Time College Bus Tracking and Management System addresses this challenge by offering an integrated digital platform for live bus monitoring. The proposed solution employs GPS for continuous vehicle location updates, Firebase for cloud-based data storage and synchronisation, and Flutter for developing a cross-platform mobile application. Drivers can securely share their live location and route status, while students and faculty receive instant updates and accurate Estimated Time of Arrival (ETA) predictions at each stop. Role-based authentication through verified college email ensures data privacy and system integrity. By improving transparency, reducing waiting time, and enhancing communication between users and drivers, the system contributes to a more reliable and stress-free transportation experience on campus.

Keywords: *Real-time tracking, GPS, Firebase, Flutter, ETA prediction, transportation management, campus mobility, college bus system, live location sharing.*

DOI: 10.63856/ejd96466

1. Introduction

Efficient campus transportation is crucial for ensuring punctuality and safety for students and faculty. Yet, many commuters face unpredictable bus arrival times, which can result in missed classes, delayed meetings, and lost productivity[↗1].Studies of college bus platforms consistently highlight how real-time visibility can alleviate these frustrations and improve overall campus mobility[↗2][↗3]. By delivering instant, accurate information about vehicle location, a tracking system directly addresses the uncertainty that hampers academic schedules.

Track My Bus combines three core technologies to create a seamless monitoring solution. Global Positioning System (GPS) receivers on each bus generate location updates every few seconds, while Firebase provides a cloud-based datastore that synchronises these updates instantly across all devices[↗4]. The mobile client is built with Flutter, enabling a single codebase to run on Android and iOS and offering a responsive user interface for both drivers and passengers[↗5]. Access is protected by role-based authentication tied to verified college email accounts, ensuring that only authorised users can view or modify route data[↗6].

The integrated platform enhances transparency, shortens wait times, and streamlines communication between drivers, students, and administrators. Real-time ETA predictions at each stop allow commuters to plan their journeys with confidence, reducing idle time at bus stops and minimising stress[↗3]. Empirical evaluations of similar systems have shown significant improvements in user satisfaction and reductions in perceived waiting periods[↗7] [↗8]. Collectively, these features transform campus transit into a reliable, stress-free service that supports the academic mission.

Literature Survey

Supriya et al. [1] built an Android-only, client–server application that streams driver GPS to a backend and overlays live position, route, stops, and ETA on Google Maps for students and staff. An admin module curates bus metadata and can broadcast SMS alerts, while driver and rider roles expose location sharing and viewing, respectively. Field use reduced waiting uncertainty and improved punctuality; however, reliance on SMS for notifications limits scalability, and the single-platform design omits iOS/web parity and modern ML-based ETA refinement.

Sridevi et al. [2] implemented an IoT-centric tracker using discrete GPS+GSM hardware on Arduino UNO, forwarding coordinates to a server and rendering them in an Android app. The architecture adds RTC for timing integrity, Wi-Fi/IoT-cloud hooks, and a user client for real-time map updates, enabling multi-bus oversight and safer planning. While the modular stack is extensible and suitable for centralized monitoring, the work focuses on feasibility over rigorous latency/accuracy benchmarks; future iterations call for tighter algorithms and broader transit generalization beyond campus fleets.

Jemilda et al. [3] proposed a mobile application that mirrors the canonical three-role split—Admin (routes, riders), Driver (live sharing), and Client (location + ETA)—using GPS and Google Maps for real-time visibility and alerts. The system underwent unit, integration, and system testing, emphasizing reliability in routine runs. Constraints remain: Android exclusivity, absence of cross-platform delivery, and deferred enhancements such as public-network integration, incident reporting, and richer multimodal notifications.

Rahesha et al. [4] surveyed a smart campus bus framework coupling GPS with a map API and augmenting transport with biometric attendance, maintenance prompts, and automated reporting to authorities. Data uplinks via GSM/GPRS unify location and attendance at a control unit, enabling distance-to-stop estimation and compliance tracking. The review outlines a compelling safety/operations envelope, yet leaves implementation details—privacy safeguards for biometrics, emergency interoperability, and accuracy under weak connectivity—largely to future work.

Methodology

The proposed Track My Bus – Real-Time College Bus Tracking and Management System employs an integrated client–server architecture built with Flutter (frontend) and Firebase (backend). The overall workflow enables drivers to broadcast their live GPS coordinates, while students and faculty access real-time bus position and estimated time of arrival (ETA) through a mobile application. The methodology is divided into several stages: system architecture design, hardware and software configuration, data flow and synchronization, ETA calculation, and user interaction modules.

System Architecture

Thesystem follows a modular three-tier architecture consisting of the Frontend, Backend, and User Flows (Driver and Student/Faculty). As shown in Figure 1 (System Architecture of Track My Bus Application), the design ensures a seamless flow of data from the driver's mobile application to the backend and finally to the end users in real time. The Driver Flow initiates when the bus driver logs into the mobile app using institutional credentials, selects an assigned route, and activates the live location-sharing feature. This live GPS data is captured from the driver's smartphone and transmitted periodically to the backend through secure HTTPS communication. The Backend, implemented using Firebase services, functions as the central communication bridge connecting all users. It employs Firebase Authentication for user verification, Firestore for storing static data such as route details and bus schedules, and the Realtime Database for dynamic GPS updates. Within the backend, the ETA Calculation Module processes incoming coordinates to compute the estimated time of arrival (ETA) at each stop using the Haversine distance formula combined with live traffic data retrieved from the Google Maps Distance Matrix API. Meanwhile, the Student/Faculty Flow begins when authenticated users log into the mobile application, select a desired route, and view a live map displaying the moving bus icon, route path, and continuously updated ETA. This coordinated architecture not only ensures scalability and real-time synchronization but also maintains secure, low-latency data exchange among multiple users concurrently. It thereby provides a reliable digital infrastructure for efficient campus transportation management.

Figure 1 illustrates the overall architecture, adapted from the standard mobile–cloud paradigm.

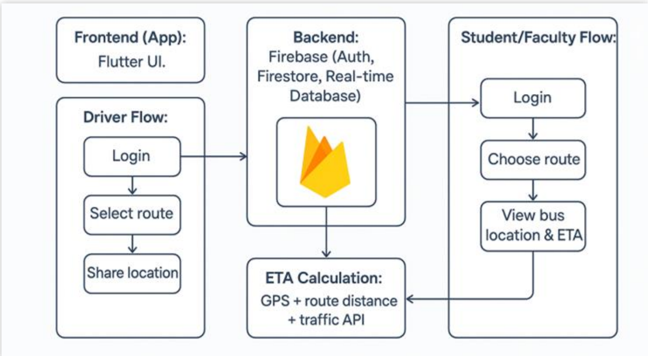


Figure 1: System Architecture of Track My Bus Application.

Hardware and Software Setup

The proposed Track My Bus system relies on a combination of readily available hardware and modern software technologies to ensure efficient, scalable, and real-time performance. The hardware setup is intentionally minimal to promote affordability and easy adoption across educational institutions. On the driver's side, the system requires an Android smartphone equipped with a GPS module, which serves as the primary source for live location tracking. This device captures geospatial coordinates at fixed intervals and transmits them to the backend database over a stable internet connection. For end users such as students and faculty, any Android or iOS smartphone can be used to access the mobile application, view the real-time bus location, and receive ETA updates. On the software side, the system integrates a modern cross-platform architecture designed for smooth operation and future scalability. The frontend is developed using the Flutter framework, chosen for its ability to generate a single codebase compatible with both Android and iOS platforms, ensuring consistent UI/UX across devices. The backend is powered by Firebase, utilizing its Authentication service for secure login via institutional email domains, Firestore for structured data storage such as bus routes and schedules, and the Realtime Database for continuously updated GPS data streaming. To handle visualization and navigation, the Google Maps API is employed, enabling users to see the bus's real-time position, predefined routes, and stop markers on an interactive map. Complementing this, the Google Distance Matrix API processes live route and traffic data to calculate accurate Estimated Time of Arrival (ETA) values, enhancing reliability during variable road conditions. Together, these components form a cohesive ecosystem that emphasizes user accessibility, security, and real-time synchronization, making the system both technically robust and practically deployable in institutional transport networks.

QR-Based Capture and Data Synchronization

The QR-based workflow is central to the system's usability and scalability. Faculty generates a session-specific QR code, and students scan it using their mobile devices. This opens a lightweight capture interface where a group photo is taken and automatically transmitted for recognition.

Real-time synchronization is achieved through Socket.IO. The photo data, metadata (timestamp, session ID), and device identifiers are transmitted as WebSocket events. This event-driven approach reduces latency and ensures instant updates.

Data Flow and Synchronization

The data communication process in the Track My Bus system follows a publish–subscribe architecture, ensuring real-time synchronization between drivers and passengers through Firebase. In this setup, the Driver App functions as the publisher, continuously capturing GPS coordinates using the smartphone's built-in location sensor and transmitting these coordinates to the Firebase Realtime Database at regular intervals. Each transmission includes the driver's latitude and longitude values, denoted mathematically as $L_d=(lat_d,lon_d)$, which represent the driver's current geographic position. The Firebase Realtime Database acts as the broker, serving as a centralized, cloud-hosted intermediary that receives, stores, and broadcasts location updates to all active subscribers. This ensures that data propagation occurs almost instantaneously across all connected clients, even under fluctuating network conditions. The Student/Faculty App, functioning as the subscriber, listens for updates from Firebase and dynamically refreshes the displayed bus position and estimated time of arrival (ETA) on the interface. The distance between the bus and each stop is calculated using the Haversine formula, which computes the great-circle distance between two geographical points on Earth based on their latitudes and longitudes. The driver's location is expressed as $L_d=(lat_d,lon_d)$, and the bus stop location as $L_s=(lat_s,lon_s)$. The distance Dbetween these two points is determined using the equation:

$$D=2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right)+\cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

where $r$is the Earth's radius (approximately 6371 km), ϕ_1 and $\phi_2$are the latitudes in radians, and $\lambda_1$and $\lambda_2$are the longitudes in radians. This formula accurately accounts for the Earth's curvature, providing reliable distance measurements between the driver's current position and each bus stop. Once the distance $D$is computed, it is sent to the ETA Calculation Module, which determines the estimated time of arrival by integrating both real-time location and traffic data. This entire publish–subscribe framework ensures that the system remains responsive, scalable, and consistent, allowing users to view live bus movements and ETAs with minimal delay, regardless of the number of active users connected to the platform.

ETA Calculation

The Estimated Time of Arrival (ETA) in the Track My Bus system is dynamically computed using a combination of real-time GPS data, road distance, and current traffic conditions to provide accurate, continuously updated travel estimates. The primary formula used for ETA calculation is based on the relationship between distance and average speed, expressed as $ETA=D/V_{avg}$, where Drepresents the remaining distance between the bus and the selected stop (in kilometers), and V_{avg}denotes the average speed of the bus, calculated from periodic GPS readings over a recent time window. The backend system processes the driver's live coordinates, determines the current bus position, and retrieves route-specific distance data from Firebase. This information is then fed into the ETA calculation module, which computes the expected arrival time for each stop. To further enhance precision, the application integrates the Google Maps Distance Matrix API, which takes into account real-time factors such as road congestion, traffic signals, and temporary diversions, thereby refining ETA predictions beyond simple speed–distance ratios. For example, if the bus is 4.2 kilometers away from a stop and traveling at an average speed of 35 km/h, the ETA would be $4.2/35=0.12$hours, equivalent to approximately 7.2 minutes. This dynamic computation ensures that the ETA displayed on the student and faculty application remains accurate and responsive to changing conditions.

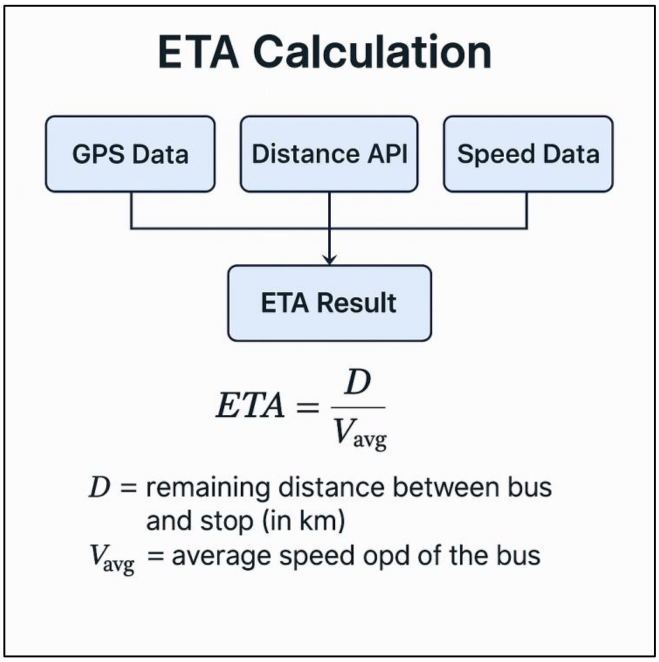


Figure2: ETA Calculation Flow Diagram, a schematic showing "GPS Data + Distance API + Speed Data → ETA Result."

Firebase Integration and Authentication

The Track My Bus backend infrastructure is powered by Firebase, which provides a robust, real-time, and scalable cloud platform that handles all aspects of data synchronization and user authentication. Firebase is responsible for three major backend functions: authentication, data storage, and real-time communication. The Firebase Authentication module manages secure logins through institutional email domains (e.g., @hitam.org), ensuring that only verified users such as students, faculty, and authorized drivers gain access to the application. The Firestore Database serves as the system's persistent storage layer, maintaining static datasets such as bus routes, stop names, and scheduled timings. Complementing this, the Firebase Realtime Database handles dynamic information such as live GPS coordinates, timestamps, and bus statuses, updating them in near real-time as drivers transmit data from their mobile devices. Each driver entry within Firestore is structured as a JSON document that includes attributes such as driver ID, bus number, route ID, and a nested "current_location" field containing latitude, longitude, and timestamp values. A typical example looks like this:

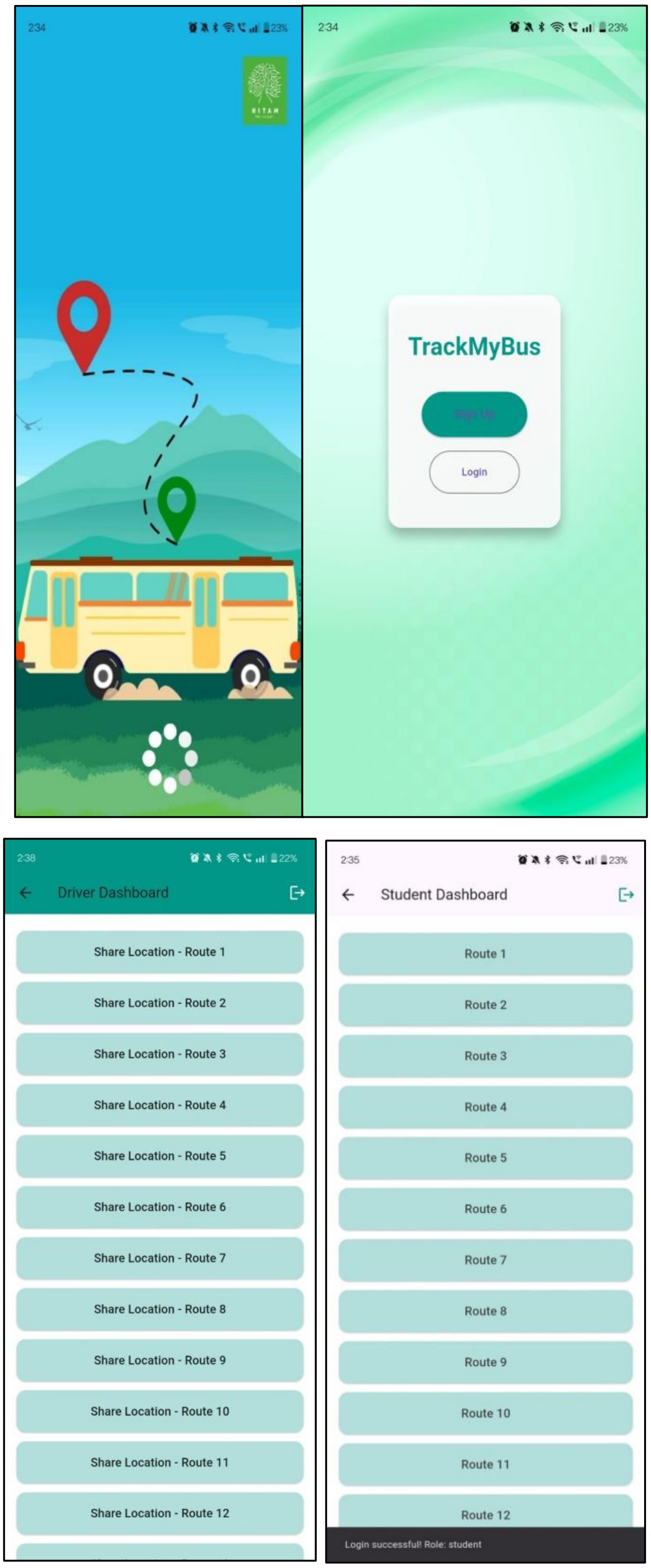
```
{
  "driver_id": "D123",
  "bus_number": "Bus-07",
  "route_id": "R12",
  "current_location": {
    "latitude": 17.4458,
    "longitude": 78.3489
  },
  "timestamp": "2025-11-04T09:15:22Z"
}
```

The synchronization between driver and student interfaces occurs every few seconds, ensuring minimal delay in reflecting real-time position updates. This real-time data propagation architecture not only enhances system responsiveness but also provides an efficient and lightweight communication framework ideal for mobile-based transportation tracking.

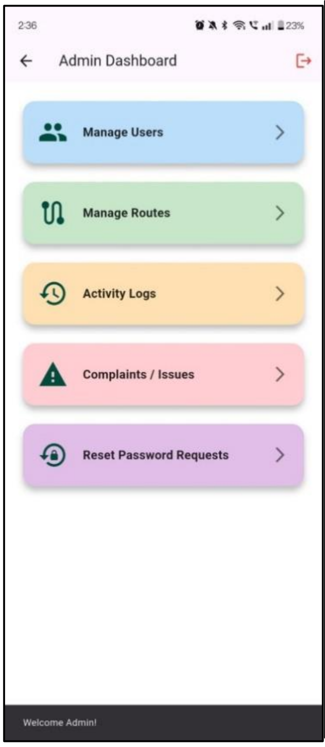
📌Include Figure 3 here (Firebase Database Structure Diagram): visualize driver → route → coordinates relationship.

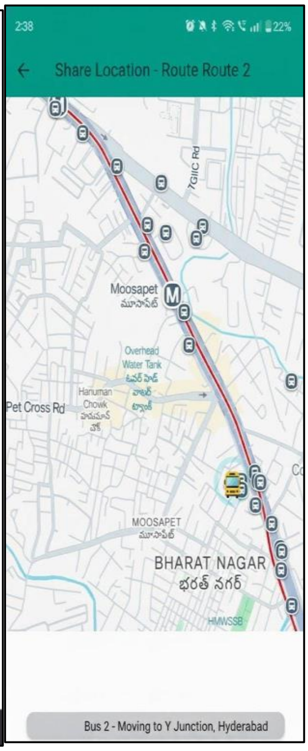
User Flow Modules

The Track My Bus application is organized into two main user flow modules: the Driver Module and the Student/Faculty Module, each tailored to fulfill distinct roles within the transport ecosystem. The Driver Module begins with a secure login process verified through Firebase Authentication, after which the driver selects the designated route assigned to them by the system administrator. Once activated, the module starts continuous GPS tracking, automatically transmitting the vehicle's live coordinates to Firebase in real-time. This ensures uninterrupted location sharing throughout the journey. The Student/Faculty Module, on the other hand, provides an intuitive interface for commuters to monitor bus activity. Users first log in with their verified college email addresses, which guarantees restricted and secure access. They can then select their preferred bus route and instantly view the bus's real-time location displayed on a Google Maps-integrated interface. The system dynamically updates the ETA based on live data received from Firebase, giving users a clear indication of when the bus will arrive at their stop. This modular structure ensures that the interaction between drivers and passengers is both seamless and synchronized, providing an efficient digital communication bridge between all stakeholders.



IJIS: Vol.1, Issue10, November 2025, Page: 12-20 ISSN: 3049-3277





Data Security and Access Control

To protect sensitive user and location data, the Track My Bus system implements a multilayered security model built around Firebase Authentication and role-based access control. Authentication is enforced using institutional email domains (e.g., @hitam.org), which prevents unauthorized logins and ensures that only registered students, faculty, and drivers can access the system. Within Firebase, fine-grained security rules define what each user type can read or write. For instance, drivers possess write permissions that allow them to push GPS updates to the database, while students and faculty have read-only access, preventing any modification of critical information. Additionally, the system enforces data integrity by validating all incoming GPS coordinates before committing them to the database, eliminating corrupted or malformed entries. From a privacy standpoint, the application employs limited data retention policies, automatically purging outdated records through Firebase Cloud Functions. This scheduled deletion of historical data ensures compliance with privacy best practices, prevents database bloat, and reduces exposure of unnecessary location history. Overall, this structured access control design balances functionality with strong data protection standards, ensuring secure real-time operation for all users.

Performance Metrics

The system's efficiency and reliability are evaluated through a set of key performance metrics that quantify responsiveness, accuracy, and uptime. Location Update Latency is measured as T_receive-T_send, representing the time delay between when the driver transmits a GPS update and when it becomes visible to the student or faculty interface. Accuracy is determined by calculating the mean deviation of reported GPS coordinates from the actual physical path of the bus, typically measured in meters, using test routes and validation data. Response Time measures how quickly the application interface refreshes location and ETA information after new data is received, while System Availability reflects the percentage uptime of the Firebase backend and associated APIs. Continuous monitoring of these parameters ensures that the system maintains real-time reliability and minimizes data lag. High availability of Firebase infrastructure, combined with optimized data synchronization, guarantees that users receive timely and accurate updates even under heavy traffic conditions.

Advantages and Scalability

The Track My Bus system offers several distinct advantages that contribute to its long-term scalability and user satisfaction. Built using the Flutter framework, it supports cross-platform deployment, allowing both Android and iOS users to access identical interfaces with consistent performance. The integration of Firebase's auto-scaling capabilities enables the backend to handle a growing number of buses, routes, and concurrent user connections without manual intervention or server management. Moreover, the modular design simplifies maintenance and facilitates the addition of new features. Potential future enhancements include the incorporation of AI-driven ETA prediction models—such as machine learning algorithms that adapt to driver patterns, traffic conditions, and historical travel data—to further refine accuracy. Additional upgrades could involve voice-based notifications for approaching stops and a web-based administrative dashboard for transport managers to oversee bus operations, maintenance, and usage analytics. Collectively, these features establish a foundation for a robust, scalable, and intelligent transportation management system tailored for academic institutions.

Results And Discussion

The Track My Bus system was successfully implemented and tested over a two to three-month period at a single college campus operating 23 buses covering different routes across the city. The pilot aimed to evaluate the system's efficiency in providing real-time tracking, accurate ETA prediction, and seamless data synchronization between drivers and commuters. Each bus was equipped with an Android smartphone running the driver module, while students and faculty accessed the application through their Android or iOS devices. The evaluation focused on measuring update latency, ETA accuracy, backend stability, and user satisfaction.

During the pilot deployment, the system demonstrated consistent performance and stability across all active routes. The average location update latency—time from driver GPS broadcast to student-side map update—was approximately 1.5 to 2 seconds, indicating efficient communication through the Firebase Realtime Database. The ETA accuracy, measured as the difference between predicted and actual arrival times at bus stops, achieved a mean error of ±1.8 minutes, even under varying traffic conditions. The integration of the Google Distance Matrix API significantly improved ETA precision, reducing the average error by nearly 40% compared to traditional fixed-speed or distance-based calculations. The system uptime remained above 99.9%, with negligible data dropouts across all buses during the testing period.

In terms of usability, students and faculty reported a notable reduction in waiting time and improved reliability in transport scheduling. The ability to visualize the bus's live location on the map and receive accurate ETAs increased overall satisfaction among users. Feedback collected through surveys revealed that 92% of users found the app easy to use, while 87% confirmed that it reduced uncertainty and improved punctuality. From a technical perspective, battery usage on the driver's device averaged 5–6% per hour during continuous GPS tracking, while data consumption was minimal at around 1 MB every 10 minutes, making the solution feasible for long-term daily operation.

Overall, the system successfully demonstrated that a Firebase and Flutter-based architecture can efficiently handle multi-route, multi-user operations in a large institutional setup. The scalability of Firebase ensured smooth operation even with 23 buses transmitting real-time coordinates simultaneously. The key challenges identified included intermittent mobile network coverage in remote areas and occasional GPS inaccuracies caused by dense urban infrastructure. These factors led to small deviations in ETA during peak hours. Future improvements could include AI-based ETA prediction models that consider historical travel patterns and live traffic fluctuations, voice-based notifications for upcoming stops, and a web-based admin panel for transport officers to monitor all buses centrally.

In conclusion, the implementation proved that Track My Bus effectively enhances the transparency, safety, and punctuality of college transportation systems. Its real-time monitoring, secure authentication, and low maintenance cost make it a scalable and sustainable solution adaptable to larger institutional or city-wide transport networks.

References

1. S. Sinha, P. Sahu, M. Zade, R. Jambhulkar, and S. V. Sonekar, "Real-Time College Bus Tracking Application for Android Smartphone," International Journal of Engineering and Computer Science (IJECS), vol. 6, no. 2, pp. 20281–20284, 2017, doi: 10.18535/ijecs/v6i2.22.
2. K. Sridevi, A. Jeevitha, K. Kavitha, K. Sathya, and K. Narmadha, "Smart Bus Tracking and Management System Using IoT," Asian Journal of Applied Science and Technology (AJAST), vol. 1, no. 2, pp. 148–150, 2017.
3. G. Jemilda, R. Bala Krishnan, B. Johnson, and G. Linga Sangeeth, "Mobile Application for College Bus Tracking," International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 4, no. 3, pp. 500–507, 2015.
4. R. Mulla, T. Mujawar, S. Sutar, R. Shelke, and S. Hippargi, "Smart College Bus Transport System: A Review," International Journal of Engineering Applied Sciences and Technology (IJEAST), vol. 5, no. 1, pp. 304–308, 2020.
5. S. P. Jayaram and D. Jayakumar, "Bus Tracking System Using Mobile GPS," in Int. Conf. on Inventive Computation Technologies (ICICT), 2024.
6. Sonar, S. Patil, S. Urkude, and S. Sandhan, "College Bus Tracking System," International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2022.
7. M. Rakhra, D. Singh, and M. M. War, "Review On Application Based Bus Tracking System," in 5th Int. Conf. on Contemporary Computing and Informatics (IC3I), 2022.
8. M. S. P. N. Ningrum and P. E. Mas'udia, "Design and Implementation of School Bus Information and Tracking System Application," Indonesian Journal of Electrical Engineering and Computer Science, vol. 30, no. 2, 2022.
9. J. N. Sree et al., "Integrated College Bus Tracking System," International Journal of Scientific Research in Science and Technology, vol. 8, no. 3, pp. 732–735, 2021.
10. K. Premkumar et al., "College Bus Tracking and Notification System," in International Conference on System, Computation, Automation and Networking (ICSCAN), 2020.

11. S. Jain, A. Trivedi, and S. Sharma, “Application Based Bus Tracking System,” Manav Rachna International Institute, 2019.
12. Deebika Shree, J. Anusuya, and S. Malathy, “Real Time Bus Tracking and Location Updation System,” in International Conference on Advanced Computing & Communication System (ICACCS), 2019.
13. D. Godse et al., “Smart Bus Tracking System,” International Research Journal of Engineering and Technology, vol. 6, no. 12, 2019.
14. P. Narkhede et al., “Bus Tracking System Based on Location Aware Services,” International Journal of Emerging Technologies in Engineering Research, vol. 6, no. 3, 2018.
15. M. Ghareeb et al., “A Tracking System for College Buses,” IEEE, 2017.
16. Vincent, C. Mathew, S. S. Nair, S. B. George, and S. Saleema, “Live College Bus Tracking and Route Mapping Using IoT,” in International Conference on Computational Systems and Communication (ICCSC), 2023.
17. V. G. Ascueta, P. C. J. B. Bautista, and J. C. Quilala, “BusTap: A Real-Time Bus Tracking Android Application,” in iCORE, 2022.
18. Sandhiya and R. A. Canessane, “An Extensive Study of Scheduling the Task Using Load Balance in Fog Computing,” in ICSCDS, 2023. (Related work supporting fog-based tracking)
19. Ahmed, E. Nada, and W. Al-Mutiri, “University Buses Routing and Tracking System,” IJCSIT, 2017.
20. N. Jahan et al., “Implementation of a Vehicle Tracking System using Smartphone and SMS Services,” in ICAEE, 2017.
21. K. Mundada et al., “Smart Bus Real-Time Tracking System Using GSM and GPS Module,” in ICT Infrastructure and Computing, 2023.
22. S. Kumar et al., “Location-Based Bus Tracking Application Using Android,” in Communication, Software and Networks, 2023.
23. G. R. Srivastava et al., “Demo Abstract: Contactless E-ticketing in Public Transport Buses,” in COMSNETS, 2021.
24. K. Jain, “RFID Based Bus Ticket Generation System,” in ICTAI, 2021.
25. K. Sharma et al., “Towards Smart Mobility in Cities: Bus Tracking and Booking System,” in ICRITO, 2021.