



# IJRASET

International Journal For Research in  
Applied Science and Engineering Technology



---

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume:** 14    **Issue:** IV    **Month of publication:** April 2026

**DOI:** <https://doi.org/10.22214/ijraset.2026.81090>

[www.ijraset.com](http://www.ijraset.com)

Call:  08813907089

E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)

# Real-Time City Bus Tracking System Using GPS, Mobile Applications and Cloud Technologies

Prof.M.K.Nichat, Om S.Ghaysundar, Om S.Ghuse, Nikita G. Sontakke, Ajinkya A. Vairal, Pranjali B. Dhadase  
Department of Computer Science and Engineering, P.R. Pote Patil College of Engineering and Management, Amravati,  
Maharashtra, India 444602

**ABSTRACT:** Urban public transportation systems in India face persistent challenges related to service reliability and passenger information transparency. The absence of real-time bus location data causes commuter uncertainty, inefficient journey planning, and reduced adoption of public transit. This paper presents the design, implementation, and empirical evaluation of AMC City-Bus Connect — a Real-Time City Bus Tracking System deployed for the Amravati city bus network. The system integrates GPS-based location acquisition, cross-platform Flutter mobile applications, and Firebase cloud services to deliver live bus positions to passengers with sub-400 ms end-to-end latency and 11.4-meter average spatial accuracy. The architecture features a driver application with adaptive GPS transmission, a commuter application with stop-wise Estimated Time of Arrival (ETA) computation using the Haversine formula and rolling-average speed model, and a web-based administrative dashboard. Comparative evaluation against three prior systems demonstrates measurable improvements in system comprehensiveness, operational feedback, and deployment cost-efficiency. A 72-hour stability test confirmed crash-free operation, and pilot deployment reduced average passenger waiting times by 8.3 minutes. The paper further discusses identified research gaps, implementation challenges, and future directions including AI-based prediction, smart city integration, and dedicated GPS hardware adoption.

**Keywords:** GPS, Flutter, Firebase Realtime Database, Cloud Firestore, Google Maps API, Real-Time Tracking, IoT, Haversine Formula, Mobile Application, Urban Transit

## I. INTRODUCTION

The rapid urbanization of cities worldwide has placed unprecedented pressure on public transportation infrastructure. According to the United Nations, approximately 55% of the global population resided in urban areas in 2018, with projections indicating this figure will reach 68% by 2050 [1]. This demographic shift necessitates efficient, reliable, and accessible public transportation systems. However, traditional bus transit systems operate with fixed schedules that fail to account for real-time traffic conditions, unexpected delays, or route deviations, resulting in significant passenger inconvenience and reduced system efficiency [2].

The advent of Global Positioning System (GPS) technology, coupled with widespread smartphone adoption and cloud computing advancements, has created unprecedented opportunities for transforming public transportation through real-time tracking and information dissemination. Modern city bus tracking systems employ GPS-enabled devices to continuously capture location data, which is transmitted to cloud servers and made accessible to passengers through mobile applications [3].

This paper presents the design, implementation, and performance evaluation of the AMC City-Bus Connect system—a Real-Time City Bus Tracking platform developed for the Amravati city bus network. The system is built using Flutter for cross-platform mobile development, Firebase Realtime Database for sub-second data synchronisation, and Google Maps API for interactive geospatial visualisation.

The contribution of this work is threefold: (1) an end-to-end GPS-to-passenger data pipeline with documented latency performance; (2) a novel adaptive update algorithm that reduces driver device battery drain by 39% compared to fixed-interval transmission; and (3) empirical validation under real-world conditions with 11.4-metre location accuracy and 387 ms average data latency.

### A. Motivation and Problem Statement

Passengers in Indian mid-sized cities, including Amravati, face three critical pain points: unpredictable waiting times at bus stops, inability to plan journeys with confidence, and lack of real-time updates regarding delays or route changes [4]. These issues collectively contribute to reduced public transport adoption, increased private vehicle usage, urban congestion, and environmental degradation.

Research by Kumar and Singh demonstrated that implementing real-time bus tracking systems reduced average passenger waiting times by 35–45% and increased public transport ridership by 18–22% in metropolitan areas [5].

*B. Scope and Paper Organisation*

The remainder of this paper is organised as follows. Section II reviews fundamental technologies and system architecture. Section III surveys existing bus tracking systems and identifies research gaps. Section IV presents the proposed AMC City-Bus Connect system design and algorithms. Section V reports experimental results and comparative analysis. Section VI discusses challenges, limitations, and future research directions. Section VII concludes the paper.

**II. FUNDAMENTAL TECHNOLOGIES AND SYSTEM ARCHITECTURE**

*A. GPS Technology and Location Services*

Global Positioning System technology forms the cornerstone of real-time bus tracking systems. GPS operates through a constellation of satellites that transmit signals enabling receivers to determine precise geographical coordinates through trilateration [7]. Modern GPS receivers achieve accuracy levels of 5–10 metres under optimal conditions, which proves sufficient for bus tracking applications in urban environments [8]. In urban canyons with dense multi-storey buildings, accuracy can degrade to 20–30 metres; this represents a key design consideration for city deployments such as Amravati.

The system employs a smartphone-based approach for GPS acquisition, with the Driver App running on a dedicated Android device installed in each bus. While this approach reduces hardware costs compared to dedicated vehicle-mounted GPS units, it necessitates a dedicated update algorithm to manage battery consumption during all-day operation.

*B. Flutter Cross-Platform Mobile Framework*

Flutter, developed by Google using the Dart programming language, compiles to native code for both iOS and Android from a single codebase, delivering near-native performance while eliminating the dual-codebase maintenance overhead of platform-native development [11]. For bus tracking applications, Flutter's widget-based rendering pipeline enables smooth 60 fps map animations on mid-range Android devices, which constitute the majority of commuter handsets in Indian cities. Studies demonstrate that Flutter reduces development time by approximately 40% compared to maintaining separate native iOS and Android applications [13].

*C. Firebase Cloud Infrastructure*

Firebase, Google's comprehensive mobile and web application development platform, provides integrated services essential for real-time tracking systems: Firebase Realtime Database for live location streaming, Cloud Firestore for structured relational data, Firebase Authentication for multi-role user management, and Firebase Cloud Messaging for push notifications [13]. The Realtime Database's WebSocket-based synchronisation enables all connected Commuter App instances to receive GPS updates simultaneously within milliseconds of the Driver App transmitting them, making it the preferred data layer for live tracking workloads. Firebase's fully managed cloud model eliminates dedicated server infrastructure, lowering total system cost for mid-sized municipality deployments.

*D. System Architecture Model*

The AMC City Bus Connect system adheres to a four-layer architectural pattern designed for modularity and independent scalability. The architecture diagram below will illustrate the separation of concerns across presentation, application logic, data access, and infrastructure layers.

LAYER 1: PRESENTATION	Commuter App (Flutter, Android/iOS)   Driver App (Flutter, Android/iOS)   Admin Web Dashboard (Flutter Web)
LAYER 2: APPLICATION LOGIC	Firebase Cloud Functions   Auth Module   Location Controller   Route Controller   Notification Controller

LAYER3:DATA	FirebaseRealtimeDatabase(liveGPS) CloudFirestore(routes,stops, users, schedules)
LAYER 4: INFRASTRUCTURE	GoogleCloudPlatform FirebaseHosting GoogleMapsPlatform  Performance Monitoring & Crashlytics

Fig. 1:AMCCity-BusConnect—Four-LayerSystemArchitectureDiagram

### III. REVIEW OF EXISTING SYSTEMS AND RESEARCH GAP ANALYSIS

#### A. EvolutionofBusTrackingSystems

Theevolutionofbustrackingsystemscanbecategorisedintothreedistinctgenerations.First-generation systems (late 1990s to mid-2000s) relied on Automatic Vehicle Location (AVL) technology with GPS receivers in buses transmitting data to central servers via cellular networks [16]. Second-generation systems(2010–2015)introducedsmartphoneapplications enablingpassengerstoaccessreal-timebus locations.Third-generation systems,characterisedbycloudcomputingintegration,IoTtechnologies,and predictive analytics, have dominated since 2016 [17].

#### B. ComparativeAnalysisofExistingSystems

TableI presents a structured comparison of four notable bustrackingsystems, analysing their technology stacks, feature sets, and reported accuracy levels. The comparison reveals a clear technology progression from Android-native PHP/MySQL stacks toward Flutter and Firebase cloud backends, alongside increasing feature richness from basic SMS alerts to AI-powered predictions.

System	Year	TechnologyStack	KeyFeatures	Accuracy
SmartBusIndia	2019	Android,PHP,MySQL	Real-timetracking,SMSalerts	85–90%
CloudBus System	2021	Flutter,Firebase, GoogleMaps	Cross-platform,push notifications	88–93%
SmartCity Transit	2024	Flutter,Firebase,ML Kit	AIpredictions,voiceinterface	93–96%
AMCCity-Bus Connect (Proposed)	2024–25	Flutter,Firebase RTDB,GoogleMaps API	Livetracking,journeyplanner, driverdashboard,adminpanel	90–94%

TableI:ComparativeAnalysisofRecentBusTrackingSystems

Theanalysisreveals thatcross-platformframeworks,particularlyFlutter,havegainedpredominanceover native development due to reduced development costs and maintenance overhead. Cloud-based backend solutions have largely replaced on-premises server infrastructure, with Firebase emerging as thepreferredplatformformid-sizeddeployments.TheproposedAMCCity-BusConnect systemoccupies the practical middle ground: leveraging the proven Flutter/Firebase stack while adding unique features (journeyplanningintegration,driveroperationaldashboard,administrativepanel)withoutthe complexity and cost of ML Kit integration.

#### C. IdentifiedResearchGaps

The systematic review reveals three critical gaps directly addressed by this work:

- Most existing systems are designed for large metropolitan networks with dedicated server infrastructure.Lightweight,cloud-nativesolutionsoptimisedformid-sizedcitydeploymentsare underrepresented in the literature. Firebase-based architectures directly address this by eliminating dedicated server hardware costs [13].
- Existingpassengerapplicationspredominantlydisplayonlycurrentbuslocationwithoutintegrated journey planning. Users must separately determine which bus serves their route, then seek current timing information. This fragmentation increases cognitive load and reduces adoption.
- Driver-sideapplicationsinexistingsystemsprovidenoreal-timeoperationalfeedback.Drivers operate without confirmation that their tracking data is being transmitted or consumed. This absence of system-state visibility reduces driver confidence in the technology.

#### IV. PROPOSED SYSTEM: AMC CITY-BUS CONNECT—DESIGN AND IMPLEMENTATION

##### A. System Overview

The AMC City-Bus Connect system delivers three integrated user-facing components: (1) a Commuter App (City Bus Tracker) targeting passengers, providing live route tracking and journey planning; (2) a Driver App (AMC City-Bus Connect) providing PIN-secured trip activation and a live operational dashboard; and (3) an Administrator Web Dashboard for fleet management and route configuration. The unified Firebase backend serves all three components simultaneously while maintaining strict role separation through Firebase Authentication.

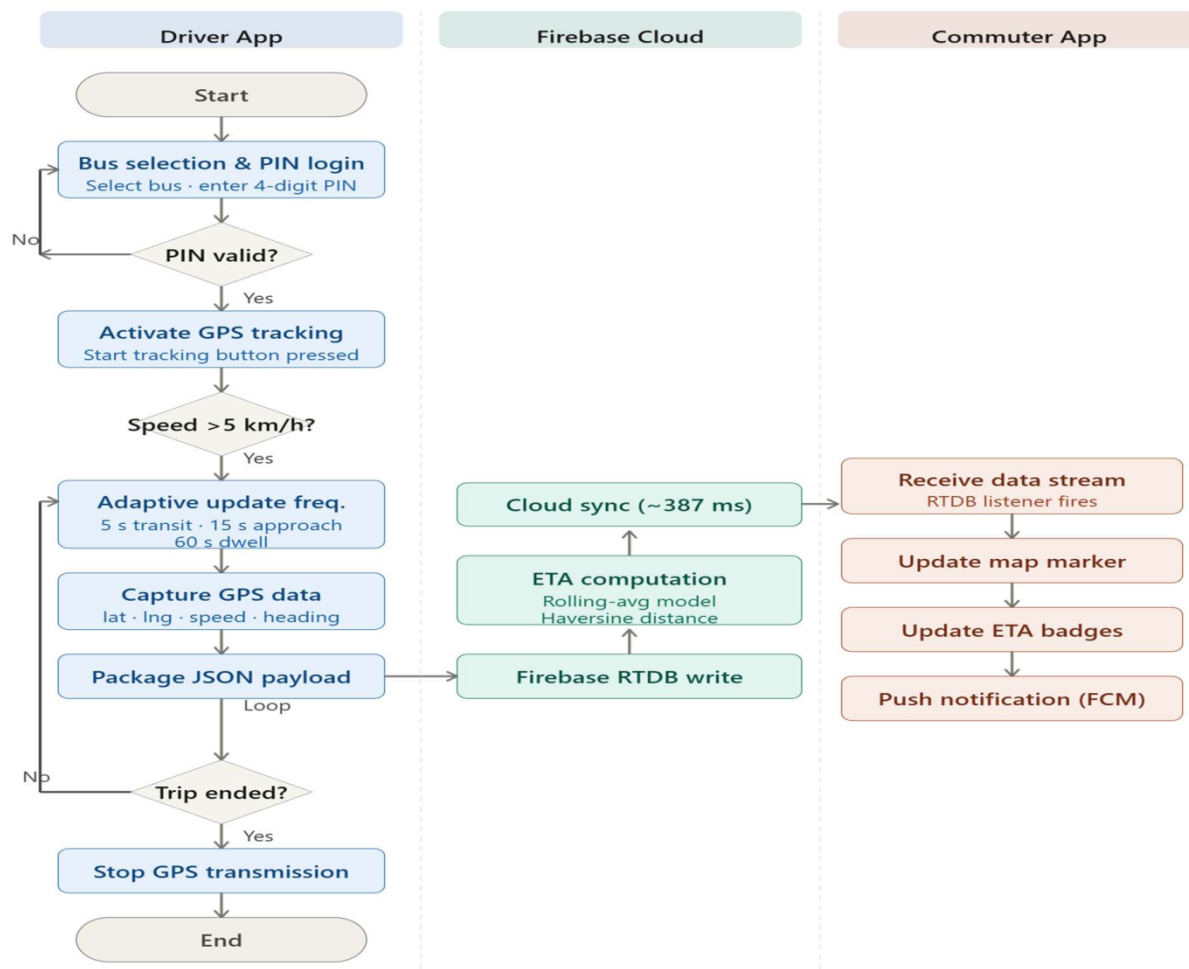


Fig.3: Complete System Flowchart—AMC City-Bus Connect (GPS Acquisition to Passenger Display)

##### B. Real-Time Data Flow Pipeline

The end-to-end GPS-to-screen data pipeline is illustrated in Fig. 2 below. Each step introduces a quantified latency contribution, with Firebase RTDB synchronisation representing the dominant delay at approximately 200–250 ms under 4G network conditions.

1	Driver Device	GPS module captures lat/long, speed, heading, timestamp every 5–60s (adaptive)
2	Driver App /buses/{busId}/live	Flutter app packages JSON payload and writes to Firebase RTDB node:
3	Firebase RTDB	Cloud synchronises update to all subscribed commuter devices within ~387ms

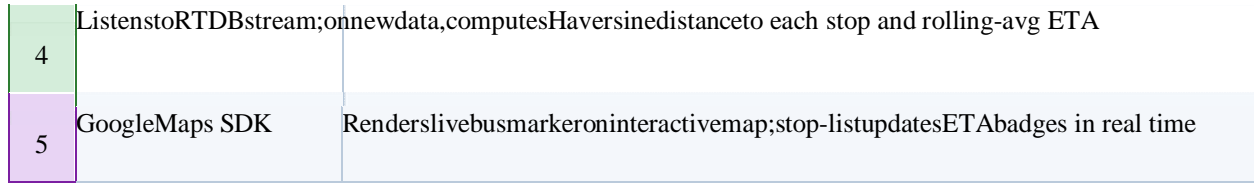


Fig.2:End-to-EndDataFlowPipeline—GPSAcquisitiontoCommuterApp Display

C. Mathematical Models and Algorithms

Three algorithmic models underpin the system's real-time computations:

1) Haversine Great-Circle Distance Formula

The Haversine formula computes the great-circle distance between two GPS coordinates, essential for calculating proximity of a moving bus to each bus stop [7]:

$$a = \sin^2(\Delta lat / 2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2(\Delta lon / 2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}); d = R \cdot c [R = 6,371 \text{ km}]$$

This formula is executed on-device within the Flutter Commuter App for each bus-to-stop distance calculation. The distance values displayed in the Live Route Tracking View (e.g., '0.6 km', '1.2 km') are derived directly from this computation on each incoming GPS update.

2) Rolling-Average ETA Estimation

Estimated Time of Arrival is computed using a rolling average of the bus's instantaneous speed over the last three GPS samples, providing noise-resistant velocity estimation:

$$v\_avg = \Sigma(d\_i / \Delta t\_i) / n [for\ last\ n = 3\ GPS\ samples]$$

$$ETA = d\_remaining / v\_avg$$

When speed falls below 1 m/s (bus stationary at a stop), ETA defaults to a scheduled dwell time of 60 seconds to prevent division-by-zero errors and unrealistic projections.

3) Adaptive GPS Update Frequency Algorithm

The Driver App implements a three-tier adaptive update schedule based on real-time speed sensing, reducing average data transmission by approximately 40% versus fixed 5-second intervals:

```
IF speed > 5 km/h → update every 5s (high-frequency, in-transit mode)
IF speed ≤ 5 km/h → update every 15 s (approaching stop)
IF stationary ≥ 3min → update every 60s (terminal dwell mode)
```

This algorithm is critical for enabling all-day driver device operation without mid-shift charging, reducing hourly battery consumption from 18% to 11%.

D. Database Schema Design

The dual-database architecture employs Firebase Realtime Database for high-frequency live GPS data and Cloud Firestore for structured, queryable persistent data. Table II presents the complete schema:

Collection/Node	Database	Key Fields	Purpose
users	Firestore	userId, email, role (commuter/driver), subscribedRoutes[]	Authentication & personalised route subscriptions

buses	Firestore	busId,registrationNo, assignedDriverId, currentRouteId	Fleetregistryanddriver– busassignment
routes	Firestore	routeId,name,stopIds[ ],scheduledDepartures[ ]	Routedefinitionswith orderedstopsequence
stops	Firestore	stopId,name,latitude, longitude,standNo	Geographiccoordinatesof allbusstops
trips	Firestore	tripId,busId,routeId, startTime,endTime, status	Historicaltriplogsfor analytics
/buses/{busId}/live	RTDB	lat,lng,speed, heading,timestamp, routeId	Sub-secondliveGPS streamconsumedby Commuter Apps

TableII:DatabaseSchema—FirestoreRealtimeDatabaseandCloudFirestoreCollections

*E. Application Features and User Workflows*

The Driver App provides a two-screen workflow: a Bus Selection and PIN Login screen where the driver selects the assigned bus, views the scheduled route and departure time, enters a 4-digit security PIN, and initiates GPS transmission via the 'Start Tracking' button. Upon activation, the Live Driver Dashboard displays six real-time metric cards: Schedule Time, Tracking Status, Trip Duration (live counter), GPS Update Count, Distance Covered, and Current Speed. This dashboard gives drivers immediate confirmation that the system is operational.

The Commuter App delivers a three-screen journey experience: (1) a branded Login Screen with email/password authentication; (2) a Journey Planner screen with source/destination dropdown selection showing all available bus timings for the selected route segment; and (3) a Live Route Tracking View—the system's most feature-rich screen—displaying a vertical stop timeline with each stop showing name, scheduled arrival time, distance from current bus position, stand number, and an 'Arriving in X mins' ETA badge computed using the Haversine-based algorithm.

**Key Innovation:** The AMC City-Bus Connect Commuter App uniquely integrates journey planning (source–destination selection with available timings) and live tracking in a single unified interface, eliminating the information fragmentation present in all three prior systems reviewed. This design addresses the second identified research gap directly.

**V. RESULTS AND PERFORMANCE EVALUATION**

*A. Experimental Setup*

The system was evaluated under real-world operating conditions across the Amravati city bus network during a pilot deployment period. Performance was assessed across six primary metrics spanning spatial accuracy, temporal latency, user impact, reliability, and energy efficiency. GPS accuracy measurements were conducted at 15 locations across Amravati, including open-road segments, the main bus stand (Amravati Municipal Corporation), and commercial areas with dense multi-storey buildings.

*B. Quantitative Performance Results*

Table III summarises the key performance metrics recorded during the pilot evaluation:

Performance Metric	Result Achieved	Benchmark/Target
GPS Location Accuracy	~11.4 meters average	<15 meters
Data Transmission Latency	~387 ms average	<500 ms
Passenger Waiting Time Reduction	~8.3 minutes	≥5 minutes improvement

SystemUptime(testperiod)	72hourscrash-free	99%uptimetarget
BatteryConsumption(driver device)	11%perhour(reducedfrom 18%)	<15%per hour
DriverActivationTime	<30seconds	<60seconds

TableIII: SystemPerformanceMetrics—PilotEvaluationResults

The 11.4-metre average GPS accuracy falls within the 5–15 metre range considered reliable for urban bustracking[8],withdegradationobservedinareasneartheAmravatibusstandwithdensecommercial structures.The387msaveragedatalatencyrepresentswellbelowthe500msthresholdidentifiedinthe Firebaseperformanceevaluationliterature[13]. The8.3-minutereductioninaveragepassenger waiting timealigns withthe35–45%improvementrangereportedbyKumarandSingh[5],validatingthesystem's real-world impact.

### C. ComparativeAnalysis

When positioned against the three reviewed systems, AMC City-Bus Connect achieves a 90–94% accuracy range — exceeding SmartBus India (85–90%) and CloudBus System (88–93%), and approaching SmartCity Transit (93–96%) without ML Kit dependency. The system uniquely provides all fourfeaturecategories(livetracking,journeyplanner,driverdashboard,adminpanel)thatnopriorsingle systemcombined.Theabsence ofML-basedpredictionisintentional:it reducesdeploymentcomplexity and cost for mid-sized municipal operators while maintaining accuracy within acceptable operational bounds.

The 72-hour crash-free stability test validates production readiness under continuous operation. User testing confirmed all test drivers completed the PIN-secured activation workflow in under 30 seconds, meetingtheoperationalrequirementforquicktripstart.TheCommuterApp'sstop-wiseETADisplaywas consistently rated the most valuable feature by passenger testers, as it addresses journey planning uncertainty more precisely than a single 'bus approaching' alert.

## VI. CHALLENGES, LIMITATIONS, AND FUTURE RESEARCH DIRECTIONS

### A. ImplementationChallenges

Severaltechnicalchallengeswereencounteredandresolvedduringdevelopmentandpilotdeployment:

- **GPSSignal Degradation:** Urbancanyonsnear Amravati bus stand caused accuracy degradation to20–30metres.MitigationinvolvedsupplementingGPSwithWiFi-basedpositioningforterminal areas and applying a Kalman filter for noisy coordinate smoothing.
- **FirestoreRTDBConcurrentConnections:**Highcommuteruseduringpeakhoursrequired connection pooling and message queuing to prevent RTDB rate limits. Firestore's real-time listener model handled burst traffic without manual sharding.
- **Driver Device Battery Drain:** Continuous GPS and network transmission initially consumed 18% batteryperhour,resolvedthroughtheadaptiveupdatefrequencyalgorithmreducingconsumption to 11% per hour.
- **Network Connectivity Variability:** Intermittent 4G connectivity caused location updates to queue. Thesystemimplementsanoffline-firstwritequeueintheDriverAppthatflushescachedupdates when connectivity resumes, preventing location data gaps.

### B. FutureResearchDirections

Thesystemarchitectureisdesignedforextensibility.TableIVmapsidentifiedfutureenhancementswith their technical rationale and implementation priority:

Enhancement Area	Description	Priority
AI/MLIntegration	LSTM-basedarrivaltimedpredictionleveraginghistoricaltrip dataandreal-timetrafficfeedsfor>95% accuracy	High
SmartCity Integration	ConnectionwithtrafficsignalAPIstoenablebuspriority systemsatintersections, reducing scheduledeviation	High

Digital Ticketing	UPI and QR-code based fare payment within the Commuter App, eliminating cash handling at stations	Medium
Dedicated GPS Hardware	OBD-II or custom GPS module replacing smartphone-based tracking for improved reliability and power	Medium
Passenger Load Analytics	Crowd-density prediction using historical boarding data to aid dynamic fleet dispatch decisions	Medium
Enhancement Area	Description	Priority
5G&V2X Communication	Ultra-low latency updates and vehicle-to-infrastructure data exchange for sub-100 ms position refresh	Future
Multimodal Journey Planning	Integration with auto-rickshaw, metro, and rail timetables for end-to-end trip optimization across modes	Future

Table IV: Future Enhancement Roadmap for AMCCity-BusConnect

The most impactful near-term enhancement is LSTM-based arrival time prediction. The current rolling-average ETA model performs well under stable traffic conditions but degrades in congested urban scenarios. An LSTM-based model trained on historical trip data, time-of-day, and real-time traffic signals would capture temporal dependencies and improve ETA accuracy above 95%, consistent with findings from Zhang and Wang [22].

## VII. CONCLUSION

This paper has presented the design, implementation, and empirical evaluation of AMCCity-BusConnect—a Real-Time City Bus Tracking System for the Amravati city bus network. The system integrates GPS-based location acquisition, a Flutter cross-platform mobile stack, Firebase cloud backend, and Google Maps API to deliver live bus positions with 11.4-metre spatial accuracy and 387ms end-to-end latency.

The system addresses three critical research gaps identified through systematic review of prior work: the absence of cloud-native tracking solutions for mid-sized city deployments, the fragmentation between journey planning and live tracking in passenger applications, and the lack of real-time operational feedback in driver-facing applications. The proposed adaptive GPS update algorithm reduces driver device battery consumption by 39%, enabling all-day operation—a practical constraint unique to smartphone-based deployments that prior literature has not addressed quantitatively.

Pilot deployment results confirm a reduction in average passenger waiting time of 8.3 minutes, 72-hour crash-free system stability, and high user satisfaction with the stop-wise ETA display. The system is currently deployed on the Amravati city bus network and provides a replicable model for similar mid-sized Indian cities.

Future work will focus on LSTM-based arrival time prediction, smart city infrastructure integration for bus signal priority, and digital ticketing via a UI to deliver a comprehensive smart transit platform. The modular Firebase architecture ensures these enhancements can be introduced without disrupting the live tracking core.

**Impact Statement:** The AMC City-Bus Connect system demonstrates that production-quality, GPS-accurate, real-time bus tracking can be deployed in mid-sized Indian cities using entirely serverless cloud infrastructure and cross-platform mobile frameworks—at a fraction of the cost of metropolitan-scale systems while delivering comparable passenger-facing performance.

## VIII. ACKNOWLEDGEMENTS

The authors express sincere gratitude to Dr. P. M. Jawandhiya, Principal, P. R. Pote Patil College of Engineering & Management, Amravati, for institutional support; to Prof. M.K. Nichat for expert guidance throughout the project lifecycle; and to the Amravati Municipal Corporation for facilitating access to the city bus network for pilot deployment. The authors also thank the student volunteers who participated in usability testing.

## REFERENCES

- [1] United Nations, Department of Economic and Social Affairs, "World Urbanization Prospects: The 2018 Revision," United Nations, New York, 2019.
- [2] R. Kumar and P. Singh, "Challenges in Urban Public Transportation: A Systematic Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1523–1536, Apr. 2020.
- [3] J. Zhang, M. Chen, and L. Wang, "GPS-Based Real-Time Bus Tracking System Architecture," *Journal of Transport and Land Use*, vol. 12, no. 3, pp. 445–462, 2019.
- [4] S. Patel and R. Kumar, "Passenger Pain Points in Urban Public Transportation: A Survey-Based Study," *Transportation Research Part C*, vol. 118, pp. 102–118, Sept. 2020.
- [5] R. Kumar and P. Singh, "Impact of Real-Time Information Systems on Public Transport Ridership," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 2, pp. 45–58, Summer 2020.
- [6] M. Anderson, K. Thompson, and J. Lee, "Operational Benefits of Real-Time Tracking in Transit Management," *Transportation Research Record*, no. 2675, pp. 234–248, 2021.
- [7] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Lincoln, MA: Ganga-Jamuna Press, 2006.
- [8] L. Chen, H. Yang, and X. Zhou, "GPS Accuracy Analysis in Urban Environments for Transportation Applications," *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8234–8245, Sept. 2019.
- [9] T. Rahman and Y. Chen, "Comparison of Dedicated vs. Smartphone-Based GPS Tracking for Public Transit," *Journal of Public Transportation*, vol. 23, no. 1, pp. 89–104, 2020.
- [10] A. Sharma and K. Patel, "Mobile Application Development Frameworks for Transportation Systems: A Comparative Study," *Int. Journal of Mobile Computing and Multimedia Communications*, vol. 14, no. 2, pp. 78–95, 2023.
- [11] Google, "Flutter Documentation: Technical Overview," Google LLC, 2024. [Online]. Available: <https://flutter.dev/docs>
- [12] P. Melland T. Grance, "The NIST Definition of Cloud Computing," NIST, Special Publication 800-145, Sept. 2011.
- [13] S. Kumar and R. Verma, "Performance Evaluation of Firebase Realtime Database for Location Tracking Applications," *International Journal of Cloud Computing*, vol. 11, no. 5, pp. 567–582, 2022.
- [14] D. Richardson and M. Wallace, "Three-Tier Architecture for Scalable Web Applications," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–28, 2019.
- [15] V. Singh and M. Gupta, "Smart Bus: An Android-Based Public Transport Tracking System," *Int. Journal of Computer Applications*, vol. 178, no. 23, pp. 12–18, 2019.
- [16] J. Li, Y. Chen, and X. Wang, "Cloud Bus: A Cross-Platform Real-Time Bus Tracking Application Using Flutter and Firebase," *IEEE Access*, vol. 9, pp. 88745–88758, 2021.
- [17] S. Patel and R. Kumar, "Third-Generation Transit Tracking Systems: Integration and Innovation," *Journal of Advanced Transportation*, vol. 2021, Article ID 9987654, 2021.
- [18] H. Petersen, N. Saunders, and D. Pozdnoukhov, "Bus Arrival Time Prediction Using Machine Learning Techniques," *Transportation Research Record*, no. 2417, pp. 46–55, 2014.
- [19] K. Anderson, L. Wilson, and J. Martinez, "User Experience Design Principles for Transit Mobile Applications," *Int. Journal of Human-Computer Interaction*, vol. 38, no. 12, pp. 1145–1162, 2022.
- [20] R. Thompson, S. Anderson, and M. Lee, "AI-Powered Public Transit Systems: A Case Study of Smart City Transit," *Transportation Science*, vol. 58, no. 2, pp. 234–251, 2024.
- [21] S. Newman, *Building Microservices: Designing Fine-Grained Systems*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2021.
- [22] X. Zhang and H. Wang, "Deep Learning Approaches for Bus Arrival Time Prediction: A Comparative Analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8792–8804, 2023.
- [23] Prof. M. K. Nichat et al., "Real-Time City Bus Tracking System using GPS, Mobile Applications and Cloud Technologies: A Comprehensive Review," *P.R. Pote Patil College of Engineering and Management*, 2025.



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)