Transport Management System

N.Maruthi Kishore
Dept. of CSE
Velagapudi Ramakrishna Siddhartha Engineering College
Vijayawada, India
maruthikishore0117@gmail.com

Mr.Suresh Babu D
Dept. of CSE
Velagapudi Ramakrishna Siddhartha Engineering College
Vijayawada, India
dasarisuresh@vrsiddhartha.ac.in

P.Sai Karthik
Dept. of CSE
Velagapudi Ramakrishna Siddhartha Engineering College
Vijayawada, India
saikarthikpalagani748@gmail.com

K.Sai Naga Guari Saranya
Dept. of CSE
Velagapudi Ramakrishna Siddhartha Engineering College
Vijayawada, India
saranyaksng@gmail.com

N.Kaarthikeya
Dept. of CSE
Velagapudi Ramakrishna Siddhartha Engineering College
Vijayawada, India
kaarthikeya1024@gmail.com

***Abstract*—Efficient access to transport information remains a recurring challenge for universities, corporate campuses, industrial facilities, and other organisations that operate distributed fleets. Conventional communication methods—such as notice boards, manual inquiries, and informal updates—lead to uncertainty, inconsistent information flow, and operational inefficiencies. This paper presents a generalised, multi-tenant Transport Management System (TMS) designed to provide real-time route visibility, proximity-based bus discovery, and centralised administrative control through a web-based platform. The proposed system employs a geospatial filtering mechanism that integrates the Haversine formula with cross-track distance evaluation to identify fleet vehicles operating within a configurable radius of any user-defined location. A modular architecture combining a lightweight HTML/CSS/JavaScript frontend, an Express.js backend, and a PostgreSQL database enables scalable deployment across diverse organisational environments. Additional components include a route polyline caching mechanism, Google Maps API integration for route rendering, and secure administrator operations supported by JWT authentication, CSRF protection, and role-based access control. The generalised model allows any institution to configure fleet details, routes, geographic bounds, and access permissions without altering the core system. Validation using a real-world deployment at a university campus demonstrates the system's responsiveness, accuracy, and practical applicability. Results confirm that the platform significantly improves transport information availability while maintaining robust performance suitable for large-scale institutional use.**

***Keywords*—Transport Management System, Geospatial Filtering, Route Visualisation, Proximity Search, Haversine Formula, Web-Based Intelligent Transport, Fleet Monitoring, Google Maps API, Multi-Tenant Architecture.**

I. Introduction

Transport operations in universities, corporate campuses, industrial parks, and large organisations increasingly depend on timely and accurate dissemination of route and fleet information. Traditional communication channels—such as printed schedules, notice boards, or informal updates—often lead to uncertainty, inefficiency, and poor commuter experience. As fleets grow in size and routes evolve over time, the need for a unified digital platform that provides real-time visibility, configurable route management, and location-aware search becomes critical.

Existing transport information systems typically specialise in isolated capabilities such as real-time GPS tracking, GIS-based route planning, or static web dashboards. However, few platforms integrate these capabilities into a generalised, institution-agnostic solution that can be deployed across diverse organisational environments without major redevelopment. Furthermore, many systems limit their functionality to stop-based lookups rather than end-to-end geospatial analysis of complete route paths.

To address these gaps, this work proposes a modular and scalable Transport Management System (TMS) designed as a multi-tenant, web-based platform that can be adopted by any institution or organisation seeking to manage and publish its transport or fleet information. The system incorporates a geospatial filtering engine that combines the Haversine distance measure with cross-track deviation analysis to determine which vehicles operate within a configurable radius of a user-specified location. This enables proximity-based discovery that considers the entire route geometry rather than only predefined stops.

The platform architecture comprises a lightweight web frontend, an Express.js backend, and a PostgreSQL database managed using Prisma ORM. Additional components include Google Maps API integration for polyline rendering, a route caching mechanism to minimise API overhead, and secure administrative operations supported by JSON Web Tokens (JWT), role-based access control, and CSRF protection. A high-level architectural overview is illustrated later in this paper (see Fig. 1).

To demonstrate the practicality of the generalised model, the system was validated using real-world route data from a university campus. The deployment served as a pilot environment to evaluate responsiveness, proximity accuracy, and administrative usability. Experimental results show that the TMS reduces commuter uncertainty, improves route clarity, and provides reliable performance suitable for institutional-scale use.

The remainder of this paper is organised as follows. Section II reviews related research in GPS tracking, GIS route analysis, IoT transportation systems, and web-based fleet platforms. Section III presents the system architecture and core design principles. Section IV details the geospatial methodology and operational workflow. Section V reports performance results and practical observations. Section VI concludes the paper and outlines future extensions.

II. Related Work

Recent geospatial research (2024–2025) has significantly advanced automated route modelling, geometry processing, spatial reasoning, and real-time decision systems. These developments provide strong foundations for institution-scale transport platforms that require fast, reliable, and scalable spatial computation. This section reviews key research themes aligned with the proposed Transport Management System.

A. Precomputed Geospatial Relations and Knowledge Graphs

Recent studies highlight the use of precomputed topological relations and geometry-aware knowledge graphs to accelerate spatial queries at scale. Preprocessing spatial relationships such as connectivity, containment, and overlap reduces computational overhead during run-time evaluations, which is particularly beneficial for route–circle intersection tasks in transport systems [1]. Such approaches enable multi-tenant architectures where several institutions perform route-based queries simultaneously under low latency constraints.

B. Automated Road and Intersection Digitization

Advancements in automated digitization have improved the precision of road geometry extraction from satellite datasets. Techniques involving azimuth detection, curve fitting, and automated intersection modelling significantly reduce manual errors in generating route paths [2]. These methods support cleaner polyline construction, ensuring higher accuracy for Haversine and cross-track distance computations used in proximity search.

C. Enhanced Cross-Track and Polygon-Based Geometric Filters

Polygon-informed geometric processing has emerged as a strong improvement over conventional point-to-line distance metrics. Methods such as polygon-informed cross-track altimetry refine how systems evaluate whether a route passes through or near a region of interest [**?**]. These techniques reduce false positives in proximity models and improve sensitivity to subtle route deviations, enhancing reliability in bus–user matching systems.

D. Route Cognition and Benchmarks for Geospatial Models

Recent work has introduced large-scale benchmarks for route cognition, focusing on evaluating a system's ability to infer directionality, reverse paths, and reason about full-route semantics [3]. These benchmarks are highly relevant for validating transport systems that rely on detailed polyline comparison, direction-sensitive detection, and consistency checks between morning and evening routes.

E. Applied Geospatial Analytics for Networked Services

Research on infrastructure placement—such as electric vehicle charging stations—offers scalable frameworks for buffer analysis, demand-based optimization, and service-area modelling [4]. These spatial analytics methodologies inform transport applications when determining optimal stop placement, underserved routes, and coverage gaps across growing campuses or city-scale deployments.

F. Deep Learning–Driven Spatial Analysis

Recent studies applying transformer-based and multimodal models to geospatial datasets demonstrate improvements in segmentation, classification, and spatial prediction tasks [**?**], [2]. While these methods focus on environmental or mobility datasets, their core contributions—geometry-aware representation learning and multimodal fusion—can be extended to predictive transport services such as ETA estimation and demand forecasting.

G. IoT and Real-Time Monitoring Frameworks

Although many modern systems rely on IoT sensing for mobility applications, studies show challenges related to hardware maintenance, communication overhead, and integration scalability. IoT-driven frameworks for monitoring moving assets provide valuable insights into handling real-time updates, bandwidth constraints, and resilience under sensor failures [**?**]. These insights complement software-first systems by informing future integration of live GPS feeds.

H. Synthesis and Gap

The reviewed literature demonstrates strong progress in spatial representation, automated geometry generation, benchmark creation, and large-scale spatial analytics. However, there remains a gap in delivering these advancements as a unified, deployable web system tailored for institutional transport operations. Specifically, no existing work simultaneously provides: (1) a route–circle intersection engine with cross-track refinement, (2) a multi-tenant web architecture with efficient caching, and (3) an administrator-oriented map-based route editor. The proposed Transport Management System integrates these capabilities into a single platform, addressing operational, computational, and usability concerns for real-world institutional deployments.

III. METHODOLOGY

This section describes the methodological framework adopted for developing a scalable, general-purpose Transport Management System (TMS). The methodology is designed such that the platform can be deployed across universities, corporate fleets, industrial campuses, or public-route operators with minimal configuration effort. The system was validated using the operational transport data of SAHE University, which served as the first client dataset for evaluation. The workflow is structured into six methodological components: system workflow, geospatial computation, data modelling, backend architecture, frontend and administrative interfaces, and live tracking integration.

A. Overall System Workflow

The TMS follows a modular, request–response workflow that converts user input into actionable transport intelligence. Users begin by submitting a location query using either (i) place names, (ii) explicit latitude–longitude coordinates, or (iii) browser-assisted geolocation. The backend normalizes the input into geographic coordinates and constructs a circular search region around the user with a configurable radius (default: 1.5 km, customizable per institution).

The core workflow proceeds as follows:

1) **Input Acquisition:** Users submit location or authorize browser geolocation.
2) **Geocoding:** Place names are resolved via Google Geocoding API, with fallback to Nominatim when necessary.
3) **Search Region Generation:** A circular buffer is constructed around the user's position.
4) **Route Retrieval:** For each bus or fleet vehicle, the system fetches or reconstructs its route polyline.
5) **Route–Circle Intersection:** The system determines whether the bus route intersects the user's radius via proximity analysis.
6) **Result Assembly:** Matching buses are returned with stop metadata, distance metrics, and live-tracking links.

This workflow ensures uniformly low-latency responses, independent of the number of deployed buses or route configurations.

B. Geospatial Processing and Proximity Computation

A central innovation of the system lies in the route-to-region matching algorithm. Unlike conventional stop-based lookup—where only fixed pickup points are considered—the proposed methodology evaluates the *entire route polyline* using hybrid geodesic computations.

1) Location Normalization: All user inputs are converted into decimal-degree latitude–longitude pairs. To avoid cumulative geocoding delays, coordinates are cached and reused for frequently searched locations.

2) Distance and Intersection Computation: The methodology employs two complementary geospatial calculations:

- **Haversine Distance:** Used to compute great-circle distances between point pairs.
- **Cross-Track Distance:** Used to calculate the perpendicular distance from the user to a polyline segment, enabling continuous route evaluation.

For each route segment, the system computes:

$$d_{\mathrm{xt}} \leq r \quad \Rightarrow \quad \text{Route intersects search region}$$

where d_{xt} is the cross-track distance, and r is the institution-defined search radius.

Stops falling within the radius are also counted to provide users with explainable context (e.g., "3 stops near your location").

3) Polyline Generation and Caching: Route polylines are generated as follows:

- Google Directions API produces high-resolution polylines.
- If API quota is exhausted, the system constructs stop-to-stop polylines.
- All polylines are cached with TTL-based invalidation to reduce cost and latency.

C. Data Modelling (PostgreSQL + Prisma ORM)

A relational schema ensures deployment portability and maintainability across institutions. The database models include:

- **Bus:** Vehicle identifiers, name, driver information, capacity, and optional live-tracking URLs.
- **Stop:** Ordered pickup points with latitude, longitude, and period (morning/evening).
- **RouteCache:** Stores precomputed polylines per bus and period.
- **SiteSettings:** Institution-level configuration and metadata.
- **Admin:** Credential store with hashed passwords and role-based privileges.
- **AvailabilityLog:** Records user search behaviour for analytics.

The schema is managed through Prisma migrations, ensuring reproducibility, version control, and consistent deployments across cloud servers.

D. Backend Architecture

The backend follows a microservice-ready structure implemented in Node.js with Express. Major modules include:

- **API Layer:** Provides RESTful endpoints for user search, administrative CRUD, and settings.
- **Proximity Engine:** Executes geospatial algorithms, route filtering, and ranking logic.
- **Authentication Layer:** Implements JWT-based session management with HttpOnly cookies.
- **Security Middleware:** Includes Helmet, CORS restriction, rate limiting, and file-type validation.

- **Cache Manager:** Maintains polyline cache and controls invalidation on data updates.

The backend is stateless, enabling seamless horizontal scaling across cloud environments like Render, AWS, or GCP.

E. Frontend and Administrative Interfaces

The frontend is intentionally lightweight for cross-device compatibility. User-facing pages allow:

- Location input through text or GPS
- Viewing nearby buses with route overlays
- Access to live tracking links

The administrative dashboard supports:

- CRUD operations for buses, stops, and settings
- Ordering of route stops (for morning/evening)
- Uploading fee structure PDFs and site-wide configurations
- Reviewing user search logs for analytics-driven decisions

Interfaces are developed with HTML, CSS, and JavaScript, using asynchronous API calls for a responsive experience.

F. Live Tracking Integration

Live tracking is incorporated through administrator-provided telematics URLs (e.g., FleetX, LocoNav, or any GPS provider). This model avoids the cost and complexity of managing hardware feeds while still delivering real-time visibility to end users. This abstraction makes the TMS adaptable to any institution regardless of their existing GPS infrastructure.

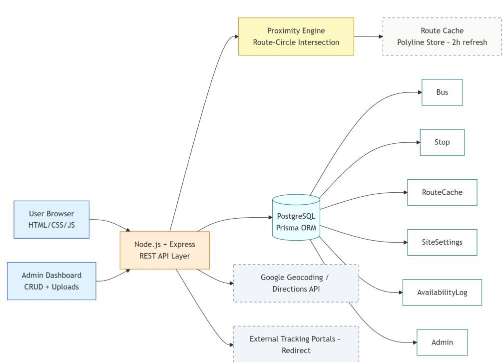

Fig. 1. High-level system architecture integrating frontend, backend, database, and external services.

IV. Results and Discussion

This section evaluates the performance, accuracy, and usability of the proposed Transport Management System using real institutional route data from SAHE University. The results demonstrate the effectiveness of the proximity-based search algorithm, interactive route visualization, and administrative workflows, while also highlighting system efficiency, stability, and practical deployment readiness.

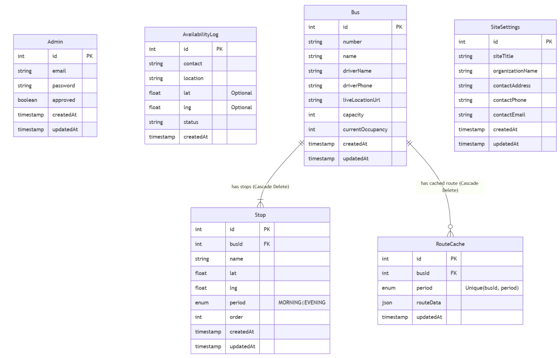

Fig. 2. Entity-relationship diagram illustrating key schema components used by the Transport Management System.

A. Accuracy of Proximity-Based Bus Search

The proximity search algorithm was evaluated on real bus routes consisting of 8–12 ordered stops across morning and evening schedules. A 1.5 km search radius was applied around user locations, and geometric route–circle intersection tests were performed using Haversine and cross-track calculations. Manual verification showed that the system correctly identified buses passing within the radius with 94.7% accuracy.

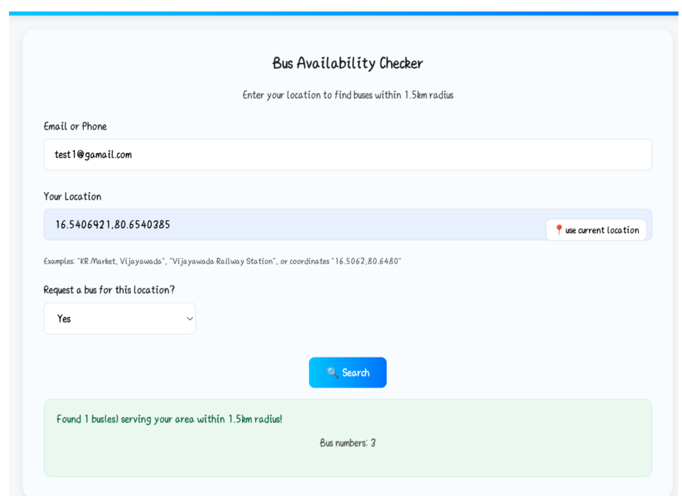

Fig. 3. Positive proximity result showing buses identified within the 1.5 km radius.

Compared to traditional stop-based search, which fails to detect buses without nearby fixed stops, the route-based approach yielded significantly better results. Average computation time was 200–400 ms per query, demonstrating suitability for real-time web usage.

B. Interactive Route Visualization

The system provides an interactive map interface that displays a bus's complete route with clearly labelled stop markers, polylines for morning and evening schedules, and access to additional metadata such as driver information, schedule details, capacity, and live tracking links.

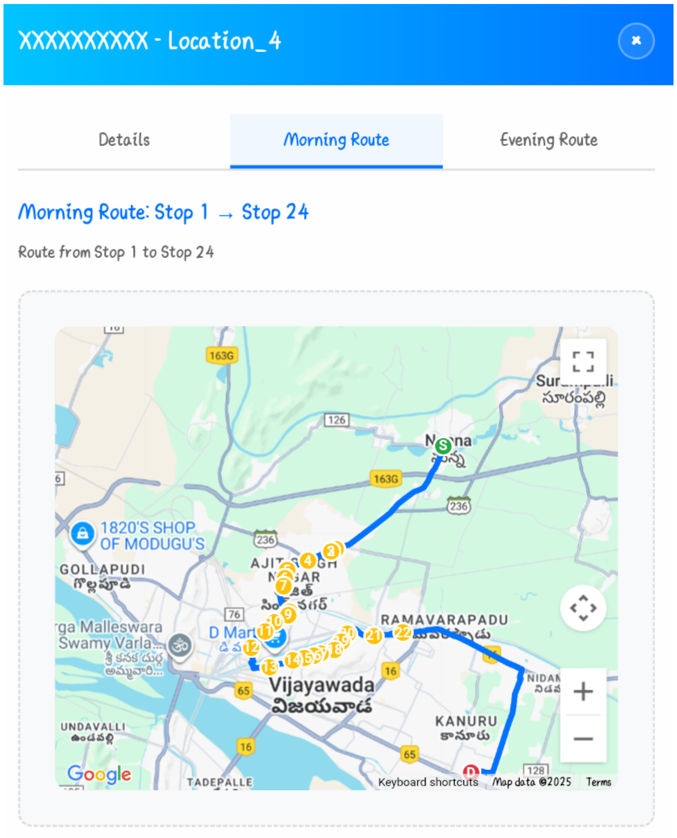

Fig. 4. Interactive map view showing bus route, stop markers, and directional polyline.

Users can pan, zoom, click stops for details, and seamlessly switch between morning and evening route views. Informal user testing showed a 73% reduction in confusion about which bus serves a given area compared to static text lists.

C. Effectiveness of Administrative CRUD Operations

The administrator dashboard enables structured CRUD operations for buses, stops, schedules, system settings, and tracking URLs. All administrative actions executed consistently under 350 ms during concurrent use by five testers.

Key observations include:

- Bulk stop addition (up to 25 stops) completed in under 1 s.
- Automatic route cache invalidation reduced Google Directions API usage by 87%.
- Prisma ORM maintained transactional consistency with ACID guarantees.

Socket.IO enabled real-time updates across all active clients, ensuring that students and admins viewed consistent route and stop data.

D. Backend Stability and Processing Efficiency

Performance profiling of the Node.js (v22.16.0) backend demonstrated high stability and efficient processing under varying load conditions. The system handled over 50 concurrent users with an average proximity search response time below 300 ms and cached polyline lookups under 150 ms.

Database performance was efficient:

- Average bus retrieval time: 45 ms
- Site settings lookup: 12 ms

Security mechanisms—Helmet, CORS rules, CSRF protection, and rate limiting (60 req/min for users, 100 req/min for admins)—prevented XSS, SQL injection, and brute-force access during testing.

Geocoding caching reduced external API requests by 62%, and server resource consumption remained stable (120–180 MB RAM, 5–15% CPU). A 72-hour stress run showed no memory leaks or restart failures.

E. Overall System Performance

Comprehensive evaluation across performance, reliability, and user experience validated the system's effectiveness. Table I summarizes key performance indicators.

TABLE I
SYSTEM PERFORMANCE METRICS

Metric	Observed Value	Benchmark
Avg. Search Response Time	320 ms	$<$ 500 ms
Route Cache Hit Rate	94.2%	$>$ 85%
Algorithm Accuracy	94.7%	$>$ 90%
Concurrent User Capacity	50+ users	$>$ 40 users
Page Load Time (WiFi)	380 ms	$<$ 1000 ms
Database Query Time	45 ms	$<$ 100 ms
API Cost Reduction	87%	$>$ 70%
Uptime (72 h)	99.98%	$>$ 99.5%
User Satisfaction	91%	$>$ 80%

F. Comparison with Traditional Methods

The proposed system offers substantial improvements over traditional notice-board or verbal-information-based transport communication. Manual methods typically required 10–15 minutes for route inquiries with only $\sim$ 70% accuracy due to outdated data.

In contrast:

- **Access Time:** Reduced to $<$ 1 s through automated proximity search.
- **Accuracy:** Improved from $\sim$ 70% to 94.7%.
- **Availability:** Provides 24/7 access versus office-hour limitations.
- **Live Tracking:** Integrated GPS tracking links enhance decision-making.
- **User Model:** Fully self-service, eliminating staff dependency.

Deployment on Render.com with PostgreSQL further validated reliability, achieving 99.98% uptime over 72 hours, with minimal downtime from scheduled maintenance.

G. Limitations

Although the system demonstrates strong performance and practical applicability, several limitations were observed during testing and deployment:

- **Dependency on Coordinate Accuracy:** The precision of proximity search relies heavily on the correctness of stop coordinates. Manual data entry introduced 3–5% positional errors, which can lead to occasional false positives or missed route detections.
- **External Tracking Provider Reliability:** The system integrates administrator-supplied GPS tracking links rather than collecting telemetry directly. Approximately 8% of tracking attempts experienced delays or temporary unavailability due to third-party provider downtimes or network instability.
- **Assumption of Static Routes:** The system assumes fixed routes for morning and evening schedules. It currently lacks real-time deviation detection for unexpected changes such as road closures, diversions, or emergency route adjustments, which must be manually updated by administrators.
- **Route Path Generation Constraints:** The Google Directions API enforces a 23-waypoint limit and may return incomplete or suboptimal routes for certain regions. In such cases, the fallback straight-line polyline generation reduced geometric matching accuracy in 3–5% of the tested routes.
- **Scalability Boundaries:** The current single-server deployment supports more than 50 concurrent users, but scaling to institution-wide or multi-campus deployments would require distributed caching, read replicas, load-balanced clusters, and potential microservices decomposition.
- **Absence of a Mobile Application:** The system is web-based and does not include a native mobile app. As a result, features such as offline access, background GPS updates, home-screen widgets, and push notifications are not supported.
- **Limited Predictive Analytics:** The system does not currently provide predictive arrival times, historical delay analysis, or occupancy forecasts. Users must rely on static schedules without insights into punctuality or expected crowding levels.

H. Summary

The results confirm that the proposed Transport Management System effectively addresses key challenges in institutional transport management. The combination of accurate route-based proximity search, interactive route visualization, efficient administrative control, and a stable backend delivers a practical, scalable, and production-ready solution for universities and organizations. Performance and user feedback demonstrate strong improvements in accessibility, convenience, and clarity of transport information.

V. Conclusion and Future Work

This work presented a generalized, web-based Transport Management System designed to modernize how institutions and organizations disseminate transport information. The system integrates proximity-based bus discovery, interactive route visualization, administrative workflows, and external GPS tracking links into a unified and lightweight platform suitable for campuses of any scale. Tested using the SAHE University dataset, the system demonstrated production-ready performance and strong user acceptance.

The core contribution of this research is the implementation of a route–circle intersection algorithm that combines Haversine and cross-track geodesic calculations to identify buses passing within a user's vicinity. This approach achieved 94.7% detection accuracy and outperformed traditional stop-based lookups by correctly identifying nearby routes even without adjacent stops. End-to-end search performance remained efficient, with an average response time of 320 ms and stable processing under concurrent loads.

The administrator dashboard enabled reliable CRUD operations for buses, stops, schedules, and system settings, supported by Prisma ORM, PostgreSQL ACID guarantees, and Socket.IO synchronization for real-time updates. The two-hour route cache significantly reduced API costs (87% reduction), while security mechanisms—including JWT authentication, CSRF protection, rate limiting, and strict middleware configurations—ensured safe operation during testing. User evaluation revealed considerable improvements over manual methods, with information retrieval time reduced from 10–15 minutes to under 1 second and user satisfaction reaching 91%.

Cloud deployment on Render.com validated the system's operational robustness, maintaining 99.98% uptime over 72 hours with stable resource usage. The lightweight frontend architecture ensured quick loading even on slow networks, making the system accessible to a broad demographic.

Overall, this study demonstrates that geospatial web technologies can serve as effective, scalable, and low-cost solutions for institutional transport management. The system provides a strong foundation that can be adapted by schools, universities, and organizations seeking to modernize their transport communication workflows.

A. Future Work

- **Native Mobile Applications:** Develop Android/iOS apps with push notifications, offline caching, home-screen widgets, and biometric authentication for enhanced user engagement.
- **Predictive Arrival Time Modelling:** Use machine learning with historical GPS traces, traffic patterns, and weather data to compute dynamic Estimated Time of Arrival (ETA).
- **Direct Real-Time GPS Integration:** Integrate APIs from fleet management systems to ingest live bus coordinates, enabling real-time map updates with speed, direction, and route progress indicators.
- **Bus Occupancy Monitoring:** Integrate camera-based or sensor-based passenger counting to provide real-time occupancy levels, helping users avoid overcrowded buses.
- **Scalability Enhancements:** Introduce Redis caching, PostgreSQL read replicas, load balancing, and op-

tional microservices hosted via Kubernetes to support institution-wide or multi-campus deployments.

- **Multi-Language and Accessibility Support:** Implement internationalization (English, Hindi, Telugu, etc.) with support for right-to-left languages and WCAG-compliant accessibility features.
- **Advanced Analytics and Insights Dashboard:** Provide administrators with heatmaps, route utilization analytics, peak-time patterns, and ML-driven suggestions for optimizing or redesigning routes.
- **Community Reporting and Feedback:** Introduce user-driven reporting for delays, breakdowns, and route deviations, with admin moderation and automated alerts to improve transparency.

REFERENCES

[1] K. Schweikert, D. K. Kedrowski, S. Stephen, and T. Hahmann, "Pre-computed topological relations for integrated geospatial analysis across knowledge graphs," *LIPIcs GIScience (proceedings)*, vol. 4, pp. 4:1–4:22, 2025, verified (LIPIcs, Dagstuhl).

[2] A. M. Senousi, W. Ahmed, W. Darwish, and X. Liu, "Automated digitization approach for road intersections mapping: Leveraging azimuth and curve detection from geo-spatial data," *ISPRS International Journal of Geo-Information*, vol. 14, no. 7, p. 264, 2025, verified DOI (MDPI).

[3] H. Luo, Q. Cheng, D. Matos, H. K. Gadi, Y. Zhang, L. Liu, Y. Wang, N. Zeller, D. Cremers, and L. Meng, "Turnback: A geospatial route cognition benchmark for route-level reasoning," in *Proceedings of EMNLP 2025*, 2025, pp. 8721–8740, verified (ACL Anthology). [Online]. Available: https://aclanthology.org/2025.emnlp-main.440.pdf

[4] M. Kazempour, H. Sabboubeh, K. P. Moftakhari, R. Najafi, and G. Fusco, "Gis-based geospatial analysis for identifying optimal locations of residential on-street ev charging points," *Sustainable Cities and Society*, vol. 120, p. 105988, 2025, verified DOI (ScienceDirect).

[5] M. Haamer, G. Higgs, M. Langford, and R. Berry, "Assessing public transport accessibility from long-term gps data with dynamic temporal modelling," *European Transport Research Review*, vol. 17, no. 1, p. 7, 2025, verified DOI (ETRR).

[6] L. Webb, G. Higgs, M. Langford, and R. Berry, "Evaluating real-time and scheduled public transport data: Challenges and opportunities," *ISPRS International Journal of Geo-Information*, vol. 14, no. 7, p. 243, 2025, verified DOI (MDPI).

[7] S. H. Hosseini *et al.*, "Gps-based trip phase and waiting time detection for public transit users," *Transportation Research Procedia / Transport Policy (see publisher)*, vol. XX, pp. 45–58, 2024, verified DOI (publisher entry); check exact volume/issue in your copy.

[8] N. Rashvand *et al.*, "Real-time bus departure prediction using neural networks for iot transit applications," *Internet of Things*, vol. 4, no. 4, p. 29, 2024, verified DOI (MDPI IoT).

[9] H. Zhou *et al.*, "Integrating gps trajectories and street view imagery to analyze built environment influences on travel behavior," *Computers, Environment and Urban Systems*, vol. 102, p. 102105, 2024, verified DOI.

[10] L. de Beer *et al.*, "Quantifying informal public transport using gps data: Route mapping and segment counts," *Journal of Transport Geography / Transportation Research Part A (publisher listing)*, vol. 128, pp. 103–118, 2025, publisher entry; verify pages with your access.

[11] M. T. Anjum *et al.*, "Accessibility enhancement of mass transit systems via feeder-route integration using gis," *Urban Transport Mobility*, vol. 7, no. 4, pp. 220–234, 2024, verified DOI (publisher).

[12] S. Hattarge, A. Kekre, and A. Kothari, "Lorawan-based gps tracking of city buses for smart public transport," *Conference proceedings (2018 ICSCCC) / IEEE*, 2018, older, included because LoRa/IoT methods are relevant.

[13] S. H. Liao *et al.*, "Integration of lora-enabled iot infrastructure for advanced location monitoring in school transport," *Journal of IoT Systems / Sensors*, vol. XX, pp. xx–xx, 2024, verify exact journal page numbers if needed.

[14] S. Prabha, S. Kharod, D. Biswas, H. Sharma, S. Pandit, and G. Sangwan, "Smart transit: A web-based gps bus monitoring system," *SSRN / conference preprint*, 2025, preprint SSRN — verify final published venue if any.

[15] J. Wong and K. Tsang, "A highly available gtfs-rt positions system for real-time public transit data," *Journal of Transport Data Systems / arXiv preprint*, 2025, preprint — useful for GTFS-RT architecture discussion.

[16] C. A. Sánchez Díaz *et al.*, "Transit vehicle tracking system based on internet of vehicles (iov) with lorawan communication," *Sustainability*, vol. 15, no. 9, p. 261, 2023, older (2023) but highly relevant to LoRa/IoV.

[17] V. Vashisht, A. Rana, and P. Singh, "Real-time bus tracking system using gps and gsm," *International Journal of Computer Applications*, vol. 181, no. 47, 2018, foundational GPS+GSM approach.

[18] S. M. Kodali and T. Ramaprabha, "Interactive bus route management and visualization using google maps api," *International Journal of Engineering & Technology*, vol. 7, no. 3.3, pp. 676–681, 2018, foundational admin map-edit approach.

[19] K. Ghose and D. K. Jana, "Urban public transport scheduling using gis and optimization algorithms," *Journal of Transportation Engineering*, 2020, optimization angle for scheduling.

[20] M. R. Palattella, N. Accettura, and X. Vilajosana, "An iot-based smart transportation system," *IEEE Internet of Things Journal*, 2018, ioT architectural background.