IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

PulseRoute: Smart Traffic Managemnt And Emergency Response System

Ruchitha V. Basaraddi[1], Nesha Sarika M. Sholapure[1], Natasha B. Ravi[1], Manisha Patil[1], Prof.Basavaraj Malapur[2*]

B.E. Student, Department of Computer Science and Engineering, Basaveshwar Engineering College (Autonomous) affiliated to Visvesvaraya Technological University, Bagalkot, Karnataka, India[1]

Assistant Professor, Department of Computer Science and Engineering, Basaveshwar Engineering College (Autonomous) affiliated to Visvesvaraya Technological University, Bagalkot, Karnataka, India[2]

Abstract: Delays in ambulance journeys can arise from congestion, road closures, intersection queues, and limited coordination between ambulance operators and destination hospitals. This paper presents PulseRoute, a software-first framework for coordinating emergency dispatch, route visualization, simulated ambulance movement, advance notification of the upcoming junction, software-based junction-priority demonstration, alternative-route handling, and hospital prearrival communication.The system is designed as a web application with a React frontend, Fast API backend, Postgre SQL database, integrated map/routing service, software GP Semulator, anda PulseAI assistant connected through the Gemini API. After an operator creates a journey, the application displays a selected route and advances an ambulance marker through sequential latitude/longitude coordinates. Backend logic compares the current simulated position and travel direction with configured junction locations to determine the next junction and generate an advance alert when an approach-zone or distance condition is satisfied. A software junction-state simulator records and displays the priority state, while the hospital dashboard presents available journey and patient information. Alternative routing can be demonstrated when a roadblock or construction scenario is introduced. Sample heart-rate, SpO_2, temperature, and respiratory-rate values are displayed only as simulated readings. The contribution is an integrated, demonstrable application workflow rather than a deployed trafficcontrol system. As measured implementation results are not yet available, quantitative performance claims are intentionally withheld and a test protocol is provided for subsequent validation.

Keywords: Ambulance Coordination, Emergency Response, GPS Emulator, Intelligent Transportation Systems, Junction Detection,Route Management, Hospital Pre-Alert, Software Prototype

I. INTRODUCTION

Emergency medical response is a time-sensitive process involving dispatch, travel to the pickup location, patient transfer, and arrival at a destination hospital. In urban environments, the ambulance journey can be affected by traffic congestion, road closures, delays at intersections, and incomplete sharing of patient and journey information. The performance of emergency services therefore depends on both route-level decisions and coordination among the stakeholders involved in a journey.

Research on emergency-vehicle routing and traffic control has examined route optimization, signal pre-emption, lane reservation, and combined strategies [1], [3]. Route selection and pre-emption are closely related: an effective route must be selected, and the approaching intersections must be handled at appropriate times. However, infrastructure-connected traffic pre-emption requires authorized interfaces and deployment considerations that are outside the scope of the current PulseRoute prototype.

PulseRoute addresses this scope through a software-first architecture. It provides a central application for creating ambulance journeys, displaying a selected route, simulating movement through coordinate updates, identifying the next configured junction, generating advance alerts, demonstrating junction priority as an application state, and notifying the selected hospital through its dashboard. It also records journey and alert history and provides a patient-monitoring interface populated with explicitly simulated readings.

The motivation of this work is to demonstrate a connected emergency-journey workflow without requiring physical RFID readers, roadside traffic-light hardware, or a live GPS device. The software emulator makes it possible to repeat a route

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511219

scenario, control coordinate progression, test junction approach thresholds, and inspect the resulting alert and event history. This supports functional validation of the application workflow, while leaving field deployment and infrastructure actuation as future work. The main contributions of this paper are:

1. An integrated web-based workflow for emergency dispatch and ambulance journey management.
2. A map-oriented route representation combined with sequential simulated latitude/longitude updates.
3. A backend method for identifying the next configured junction using simulated position and travel direction, withadvance alert generation.
4. A software-only junction-priority state demonstration and a roadblock/alternative-route workflow.
5. A hospital dashboard for pre-arrival journey and patient information, together with simulated vital readings and eventhistory.
6. A defined evaluation protocol for testing route progression, junction alerts, rerouting, dashboard notification, andrecord persistence.

II. LITERATURE REVIEW

[1] S. Humagain, R. Sinha, E. Lai, and P. Ranjitkar (2020) in their paper "A Systematic Review of Route Optimisation and Pre-emption Methods for Emergency Vehicles" (published in Transportation Reviews) provided a systematic review of emergency-vehicle route optimization and pre-emption methods. Their review highlights the role of travel-time reduction and identifies gaps related to dynamic traffic information, travel-time-aware routing, effects on other traffic, and realworld implementation. This study provides a broad basis for understanding the coordination requirements addressed by the proposed system.

[2] K. Shaaban, M. A. Khan, R. Hamila, et al. (2019) in their paper "A Strategy for Emergency Vehicle Preemption and Route Selection" (published in Arabian Journal for Science and Engineering) formulated a joint strategy for emergencyvehicle path selection and signal pre-emption. Their work emphasizes selecting a suitable path and initiating pre-emption at appropriate points along the route, while considering the effect on non-priority traffic. This reinforces the importance of treating the route as a connected corridor rather than a sequence of unrelated intersections.

[3] W. Yu, W. Bai, W. Luan, and L. Qi (2022) in their paper "State-of-the-Art Review on Traffic Control Strategies for Emergency Vehicles" (published in IEEE Access) classified emergency-vehicle traffic-control strategies into route optimization, signal pre-emption, lane reservation, and mixed strategies. This classification supports the coordination requirements of emergency mobility and motivates the route and junction-alert modules of the proposed system.

[4] R. J. Fleischman et al. (2013) in their paper "Predicting Ambulance Time of Arrival to the Emergency Department Using Global Positioning System and Google Maps" (published in Prehospital Emergency Care) investigated ambulance arrival-time prediction using GPS and mapping information. Their research shows that location and arrival-time information can support hospital preparation, which motivates the destination-hospital pre-alert feature of the proposed platform.

[5] S. N. Shetab-Boushehri, P. Rajabi, and R. Mahmoudi (2022) in their paper "Modeling Location-Allocation of Emergency Medical Service Stations and Ambulance Routing Problems Considering the Variability of Events and Recurrent Traffic Congestion: A Real Case Study" (published in Healthcare Analytics) considered ambulance routing under uncertainty in emergency events and recurrent traffic congestion. This work motivates the inclusion of route representation, coordinate updates, and hospital pre-alert information in a coordinated platform.

The PulseRoute prototype is not intended to reproduce the infrastructure-connected mechanisms evaluated in the above studies. In particular, it does not issue live signal-controller commands or derive its coordinate stream from a physical GPS receiver. Its focus is the application-level integration of dispatch, route visualization, coordinate simulation, junction alerting, alternative-route demonstration, and hospital dashboard communication.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

III. BLOCK DIAGRAM

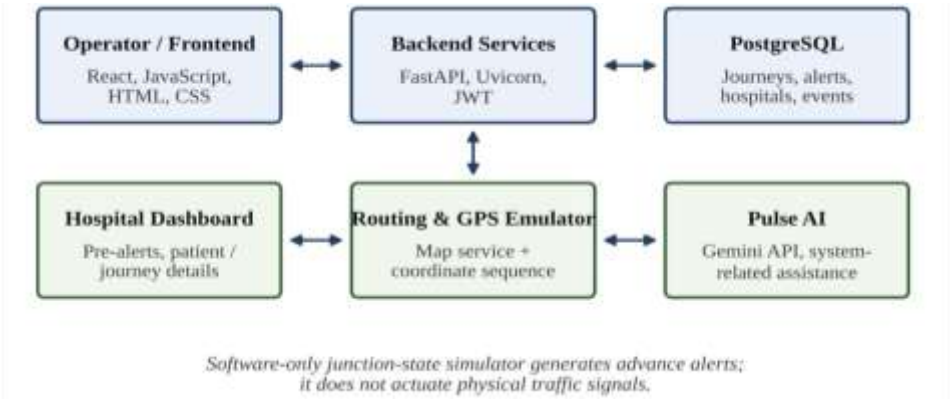

Fig. 1 Block Diagram: High-Level PulseRoute Architecture and Application Components

The frontend provides operator-facing forms, map views, journey status, hospital dashboard views, a patient-monitoring display, and Pulse AI access. The FastAPI backend exposes REST endpoints and applies journey, routing, junction-alert, notification, and history logic. PostgreSQL stores persistent application records, with SQLAlchemy used for objectrelational mapping and Pydantic for request/response validation. JWT-based authentication protects user access and APIs.

The mapping/routing service is integrated by the application for route representation and the alternative-route workflow. The software GPS emulator advances the current ambulance position using a coordinate sequence associated with the selected route. A backend junction module evaluates the current simulated location against configured junctions. The hospital dashboard retrieves the pre-alert and available journey information from the same application data layer. Pulse AI is an assistant interface powered by the Gemini API; its role is system-related assistance rather than autonomous traffic control. The technologies used in each component are summarized in Table 1.

TABLE I IMPLEMENTATION TECHNOLOGY STACK OF PULSEROUTE

Component	Technology	Purpose in PulseRoute
Frontend	React, JavaScript, HTML, CSS	User interface, dashboards, map and journey views
Backend	Python, FastAPI, Uvicorn	REST APIs and application logic
Database	PostgreSQL	Persistent application records
ORM / validation	SQLAlchemy, Pydantic	Database models and request/response validation
Authentication	JWT-based authentication	Secure access to protected APIs
Mapping / routing	Integrated map and routing service	Route display and alternative-route workflow
GPS simulation	Software GPS emulator	Coordinate sequence and simulated movement
Junction simulation	Backend logic and software-based junction-state simulator	Advance alerts and green-corridor demonstration
AI assistance	Gemini API	Pulse AI assistant
Development / testing	VS Code, Postman, Swagger UI, GitHub	Development, API testing and version control

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

IV. METHODOLOGY

A. Problem Definition and System Requirements

Given a pickup location, a destination hospital, patient/journey information, a selected route, a sequence of simulated ambulance coordinates, and a set of configured junction locations, the system must maintain an emergency journey, display the route and current simulated position, identify the next junction in the direction of travel, generate an advance alert when an approach condition is satisfied, and expose relevant pre-alert information to the destination hospital dashboard. The functional requirements derived from this definition are listed in Table 2.

TABLE II FUNCTIONAL REQUIREMENTS OF PULSEROUTE

ID	Requirement	System behavior
FR-1	Journey creation	Accept pickup, patient, emergency and hospital details; create a journey record.
FR-2	Route visualization	Display a route between selected locations using the configured map/routing service.
FR-3	Coordinate simulation	Advance the ambulance position through a sequence of latitude/longitude points.
FR-4	Junction detection	Identify the next configured junction based on position and direction of travel.
FR-5	Advance alert	Trigger a notification when a configured approach-zone or distance threshold is reached.
FR-6	Priority simulation	Record/display a software junction-priority state without physical signal actuation.
FR-7	Alternative route	Update route display when a roadblock/construction scenario is introduced.
FR-8	Hospital dashboard	Display ambulance pre-alert and available patient/journey details.
FR-9	Monitoring/history	Display labelled simulated vitals and retain journey, alert and event history.
FR-10	AI assistance	Provide Pulse AI system-related assistance through the Gemini API.

The application should maintain consistent journey identifiers across dispatch, alerts, and hospital views; validate API inputs; protect authorized operations through JWT-based authentication; persist records in PostgreSQL; and provide traceable event history. Demonstration reliability also requires deterministic coordinate sequences and documented junction thresholds so that test runs can be repeated. These are design considerations to be verified during implementation testing, not claims of measured system performance.

B. Operational Workflow



Fig. 2 PulseRoute Operational Workflow from Dispatch to Hospital Pre-Alert

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

Emergency journey creation. The ambulance operator enters the pickup location, patient details, emergency information, and destination hospital. The backend validates the request, creates a journey record, and returns a journey reference that is used to associate subsequent coordinate updates, junction alerts, and event records.

Route generation and map display. The application obtains or represents a route between the selected origin and destination and displays it on a map. The routing service used in the implementation should be identified, including its routing profile and whether live traffic information is available. The application must not describe a route as trafficoptimal unless the configured service supplies and the implementation uses such information.

GPS emulator and movement updates. A software emulator supplies an ordered set of latitude/longitude coordinates representing the ambulance's movement along the selected route. During the demonstration, the current coordinate is advanced through the sequence, enabling controlled testing of movement and junction-alert logic. The emulator does not represent a live GPS receiver or a continuously connected ambulance device.

Next-junction detection. The backend compares the current simulated position and travel direction with configured junction locations along the planned route. The module selects the next junction in the direction of travel and checks whether the ambulance has entered a configured approach zone or reached a specified distance threshold. When the condition is satisfied, the system generates an advance pre-alert containing the journey reference, junction identity, and available approach information. For reproducible testing, the junction list, route order, approach threshold, and coordinate sequence should be stored or configured explicitly; the geospatial distance formula, threshold value, and junctionmatching algorithm are implementation-specific and should be documented from the final code.

Junction-priority demonstration. After a junction alert is generated, the application records and displays the corresponding junction-priority state and event notification. This is a software simulation intended to demonstrate how a traffic-management layer could be informed before ambulance arrival. It does not physically change a signal phase, connect to a municipal controller, or guarantee a cleared corridor.

Alternative-route handling. When a roadblock or construction scenario is introduced, the application requests or demonstrates an alternative route, subject to the capabilities of the configured routing service. The updated route is then shown in the map view. Test records should identify the blocked segment, the original route, the alternative route, and whether the routing service successfully returned a replacement path.

Hospital pre-alert and patient monitoring. The application creates a pre-arrival alert for the selected hospital dashboard, which displays available patient and journey details so that hospital staff can prepare before arrival. A patientmonitoring interface displays sample heart rate, SpO_2, temperature, and respiratory-rate readings. All such values are clearly labelled as simulated and are not represented as live clinical measurements.

Data management and Pulse AI. Backend APIs manage journey records, junction alerts, hospital information, patient records, and event history. Pulse AI provides an interface for system-related queries using the Gemini API. The AI component is an assistance feature and is not described as making clinical decisions, controlling signals, or autonomously dispatching ambulances.

C. Data Flow and Module Design

PulseRoute relies on a shared journey reference to connect the different application modules. The dispatch form creates the journey and stores the pickup, destination, emergency, and available patient information. Route generation associates a route representation with that journey. The GPS emulator then advances the simulated position, and each update is evaluated against the configured junction information. Alerts, reroute events, and hospital pre-alert information retain the same journey reference so that the operator view, hospital dashboard, and event history remain consistent.

The current route and the current simulated position are treated as separate pieces of state. A route describes the path selected by the mapping/routing service, whereas the simulated position describes the emulator's latest coordinate. A route change may invalidate a previously selected next junction; therefore, after an alternative route is accepted, the system should recalculate the next-junction candidate using the active route and avoid presenting an alert tied only to the superseded path. Event records provide traceability across these state changes and can include an event identifier, journey reference, event type, timestamp, and a related junction or hospital identifier. The modules and the information handled by each are summarized in Table 3; actual field names and database relationships should match the implemented database models.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

TABLE III DATA AND MODULE DESIGN OF PULSEROUTE

Module	Principal information handled	Purpose
Journey management	Journey ID, pickup, destination, patient and emergency details	Create and track emergency journeys
Route management	Origin, destination, route representation, roadblock scenario	Display route and alternative route
GPS emulator	Ordered latitude/longitude sequence, current index/position	Simulate ambulance movement
Junction alert	Configured junctions, direction, approach threshold, alert status	Identify next junction and issue advance alert
Priority simulator	Junction reference, priority state, event timestamp	Demonstrate software-only junction priority
Hospital dashboard	Hospital, journey and available patient details	Show pre-arrival information
Patient monitoring	Sample heart rate, SpO2, temperature, respiratory rate	Display clearly labelled simulated readings
Event history	Journey, alert, reroute and notification events	Support traceability and demonstration
Pulse AI	User query and assistant response	Provide system-related assistance

D. Junction Alert Decision Logic

The junction-alert process is a sequence of checks. The backend obtains the active journey and route, reads the latest simulated coordinate, and identifies the direction of movement from the ordered coordinate updates. It then considers configured junctions that remain ahead along the active route and selects the next eligible junction. The approach condition is evaluated using the configured approach zone or distance threshold. When the condition is satisfied, an alert event is created and made available to the application views. The next coordinate update repeats the evaluation until the junction has been passed or the route state changes.

Several implementation details affect reliability. The route order should be considered so that a junction already passed is not selected again; repeated alerts for the same journey and junction should be avoided while the ambulance remains within one approach zone; the selection rule should be deterministic when two junctions are close together; and, if the emulator restarts or the route changes, the backend should re-evaluate the active position and route rather than rely on stale state. Alert lead time is influenced by the approach threshold, coordinate spacing, emulator update interval, simulated speed, and route geometry, so a threshold that works for one demonstration route may not suit another.

E. Mathematical Model

The junction-alert logic of PulseRoute is formalized below. The exact threshold value, distance formula, and duplicatesuppression mechanism are implementation-specific and should be confirmed against the final code. 1. Input Parameters

Let,

· i = Journey reference (identifier of the active emergency journey)

· $P_t = (\varphi_t, \lambda_t)$ = Current simulated ambulance position at update step t · $J_j = (\varphi_j, \lambda_j)$ = Configured junction j, with route-order index s_j along the active route · s_t = Route-order index of the current simulated position · θ = Configured approach-zone / distance threshold

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

V. DISTANCE COMPUTATION

$$d(P_t, J_j) = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_t \cos\varphi_j \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \quad (1)$$

Where r is the mean Earth radius, and Δφ and Δλ are the latitude and longitude differences between P_t and J_j. 3. Next-Junction Selection

$$j^* = \min\{\, j \mid s_j > s_t \,\} \quad (2)$$

Where j* is the nearest configured junction that remains ahead of the ambulance along the active route; a junction already passed is never re-selected.

VI. ADVANCE-ALERT CONDITION

$$A_t = 1 \text{ if } d(P_t, J_{j^*}) \le \theta \text{ and no alert exists for } (i, j^*); \quad A_t = 0 \text{ otherwise} \quad (3)$$

Where $A_t = 1$ triggers an advance pre-alert containing the journey reference, junction identity, and approach information. 5. Alert Lead Time (5)

$$v_{sim} = \frac{\Delta s}{\Delta t}$$

$$T_{lead} \approx \frac{d(P_{alert}, J_{j^*})}{v_{sim}} \quad (4)$$

Where Δs is the coordinate spacing, Δt is the emulator update interval, v_{sim} is the simulated speed, and T_{lead} is the time between the alert and simulated arrival at the junction.

VII. EVALUATION METRIC

$$SR = \frac{N_{success}}{N_{total}} \times 100\% \quad (6)$$

Where SR is the success rate of a test scenario (for example, route generation, alert delivery, or reroute), $N_{success}$ is the number of successful runs, and N_{total} is the number of repeated runs.

VIII. RESULTS AND DISCUSSION

The evaluation uses a predefined route in Bengaluru and scenarios covering dispatch, coordinate progression, nextjunction detection, pre-alert timing, roadblock/alternative-route handling, hospital notification, and event logging. A repeatable evaluation uses a documented route, fixed coordinate sequence, configured junction locations, and recorded threshold values, and each scenario is repeated to identify intermittent failures.

Reproducibility controls. The evaluation dataset preserves the route coordinate sequence, configured junction identifiers, approach thresholds, simulation update interval, and roadblock scenario definition. A fixed scenario configuration enables the same functional test to be replayed after code changes. Each run is assigned a unique run identifier and linked to the journey record, event log, and any screenshots collected as evidence.

Separation of test layers. Unit-level checks verify coordinate ordering, route-index progression, junction selection, and alert-state transitions. API-level checks verify journey creation, protected endpoint behavior, event persistence, and dashboard retrieval. End-to-end checks exercise the operator workflow from dispatch through simulated movement to hospital pre-alert. Reporting these layers separately helps identify whether a failure originates in the interface, backend

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

logic, database, or an external map/routing service. The scenarios and the evidence collected for each are listed in Table 4.

TABLE IV EVALUATION SCENARIOS, PROCEDURES AND EVIDENCE / METRICS

Scenario	Procedure	Evidence / metric
Dispatch validation	Submit valid and invalid journey details	Successful journey creation, validation behavior, API response time
Route visualization	Select pickup and destination	Displayed route, route-service response, route-generation success
Coordinate progression	Run a fixed coordinate sequence	Marker position and update interval; compare against sequence
Junction detection	Approach each configured junction	Correct next-junction identity and detection outcome
Advance notification	Vary approach threshold and coordinate spacing	Alert lead time and missed/duplicate alert count
Roadblock response	Introduce a blocked segment	Alternative route returned and displayed; reroute latency
Hospital pre-alert	Create and progress an emergency journey	Dashboard delivery success and notification latency
Event persistence	Review journey and alert history	Record completeness and retrieval success
Patient display	Load sample vital readings	Correct display and simulated-data labelling
Access control	Test authenticated and unauthenticated API requests	Authorized/unauthorized behavior

Metrics. The following metrics are suitable for reporting after implementation: journey-creation success rate; routegeneration success rate; coordinate-update interval; junction-detection correctness; advance-alert lead time; missed and duplicate alert counts; alternative-route success rate and response time; hospital pre-alert delivery success and latency; event-record completeness; and API response time. Where ground-truth GPS or traffic data are unavailable, only functional outcomes from the emulator are reported, and claims about realworld location accuracy or traffic-delay reduction are explicitly avoided.

Results status. This manuscript defines expected outcomes and an evaluation scope but does not include measured test results, screenshots, sample logs, timing measurements, or statistical comparisons. Therefore, it does not report numerical performance gains, detection accuracy, response-time reduction, or clinical benefit. The prototype can establish whether the configured software workflow triggers an alert before simulated arrival, but this alone does not demonstrate a realworld reduction in ambulance response time.

Comparison with related approaches. Table 5 contrasts the implementation boundary of PulseRoute with categories of capabilities found in related emergency-vehicle research. It is not a claim that all prior studies contain every capability listed, nor is it a benchmark ranking, because the referenced studies use different network models, assumptions, datasets, and evaluation methods [1]–[3]. A quantitative comparison would require a shared route network, traffic assumptions, baseline routing strategy, and repeatable test conditions.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511219

TABLE V COMPARISON OF PULSEROUTE WITH INFRASTRUCTURE-CONNECTED RESEARCH APPROACHES

Dimension	Infrastructure-connected research approaches	PulseRoute prototype scope
Vehicle location	May use live GPS/GNSS or vehicle communication	Software-emulated coordinate sequence
Route planning	May optimize using traffic, travel time and network conditions	Map-based route representation; providerdependent rerouting
Intersection handling	May communicate with signal controllers for preemption	Application-level junction alert and priority-state simulation
Hospital coordination	May integrate dispatch and medical-service workflows	In-application hospital pre-alert dashboard
Patient data	May use connected medical/vehicle data	Sample/simulated vital readings
Deployment	Can include field trials or infrastructure integration	Software prototype; no roadside hardware actuation

Design trade-offs. The software-first approach reduces the hardware and deployment requirements for an academic prototype and makes repeatable demonstrations possible. Its corresponding limitation is that the emulator cannot capture real positioning errors, mobile-network interruptions, driver behavior, or the complexity of an operational traffic-control environment. Using a routing service also creates an external dependency: route availability, road-network coverage, traffic data, and alternative-path behavior may vary by provider and configuration. The paper therefore refers to route generation and map display rather than claiming globally optimal or traffic-optimal routing.

Failure handling and human oversight. A demonstration should account for missing map responses, invalid coordinates, unavailable database connections, expired authentication tokens, and failed hospital-dashboard retrieval. The interface should distinguish a failed external service request from a valid route or a successfully delivered pre-alert, and if an alternative route cannot be obtained, the system should preserve the original journey record and make the failure visible to the operator. Pulse AI output is informational and should not override the operator, determine patient treatment, or issue traffic-control commands; any future expansion into operational decision support would require a separate validation process, clear responsibility boundaries, and safeguards against inaccurate or incomplete generated information.

Limitations, safety, and privacy considerations. Simulated coordinates are not live GPS readings, and simulated vital signs are not medical-device measurements. Junction alerts and priority states are software demonstrations and do not physically operate roadside signals or connect to a city traffic-control network. Hospital pre-alerts are displayed inside the application and are not claimed to be integrated with an external hospital information system, and alternative-route behavior depends on the configured mapping/routing service. Because the system handles patient and journey information, a real deployment would require appropriate access control, secure communication, data minimization, audit logging, retention policies, and review under applicable privacy and healthcare requirements. A software notification must not be represented as a guaranteed emergency traffic clearance.

Deployment considerations. Moving from a controlled prototype to an operational service would require additional validation beyond functional testing. A live location feed would need to be authenticated, monitored for stale or implausible coordinates, and handled safely during network interruptions. Traffic and road-closure data would need an identified provider, documented update frequency, and a fallback when the service is unavailable. Any connection to traffic-control infrastructure would require authorization from the relevant authority, a secure interface, fail-safe behavior, and human operational oversight. Hospital integration would require agreement on data formats, identity matching, access permissions, and secure communication, and any introduced medical sensors would need to be distinguished from the simulated values used in the current prototype. These are future deployment considerations and are not claimed as completed PulseRoute capabilities.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511219

IX. CONCLUSION

PulseRoute is a software-first framework for coordinating ambulance emergency journeys and demonstrating selected intelligent traffic-management functions. It integrates journey creation, map-based route visualization, simulated GPS movement, next-junction detection, advance alerts, software-only junction-priority states, alternative-route handling, hospital pre-alerts, simulated patient-vital display, event history, and Pulse AI assistance. The architecture is designed around React, FastAPI, PostgreSQL, an integrated map/routing service, and a coordinate-sequence emulator.

The manuscript deliberately distinguishes simulated application behavior from live GPS, clinical sensing, and physical traffic-signal control. Future evaluation should report reproducible functional tests and measured performance before any claims of operational benefit are made.

X. FUTURE SCOPE

Future enhancements include connecting a live GPS receiver or secure ambulance mobile application to stream real coordinates; validating junction detection and pre-alert timing across a larger set of junctions; integrating verified live traffic and road-closure feeds; connecting to authorized traffic-control infrastructure for signal actuation; integrating calibrated medical sensors; and establishing secure interoperability with hospital information systems. Additional work should include multi-ambulance conflict handling, service availability monitoring, security testing, and field validation under authorized conditions.

ACKNOWLEDGEMENT

We, the students, express our sincere gratitude to our project guide for the valuable mentorship, technical guidance, and constant support provided throughout the conceptualization, design, and implementation of PulseRoute. We also extend our appreciation to the Department of Computer Science and Engineering, Basaveshwar Engineering College (Autonomous), Bagalkot, for providing the facilities and academic environment necessary to carry out this work. This project reflects our collaborative effort towards a software-first, demonstrable framework for ambulance journey coordination and hospital pre-arrival communication.

REFERENCES

[1]. S. Humagain, R. Sinha, E. Lai, and P. Ranjitkar, “A systematic review of route optimisation and pre-emption methods for emergency vehicles,” Transportation Reviews, vol. 40, no. 1, pp. 35–53, 2020, doi: 10.1080/01441647.2019.1649319.

[2]. K. Shaaban, M. A. Khan, R. Hamila, et al., “A strategy for emergency vehicle preemption and route selection,” Arabian Journal for Science and Engineering, vol. 44, pp. 8905–8913, 2019, doi: 10.1007/s13369-019-03913-8.

[3]. W. Yu, W. Bai, W. Luan, and L. Qi, “State-of-the-art review on traffic control strategies for emergency vehicles,” IEEE Access, vol. 10, pp. 109729–109742, 2022, doi: 10.1109/ACCESS.2022.3213798.

[4]. R. J. Fleischman et al., “Predicting ambulance time of arrival to the emergency department using Global Positioning System and Google Maps,” Prehospital Emergency Care, vol. 17, no. 4, pp. 458–465, 2013, doi:3109/10903127.2013.811562.

[6]. S. N. Shetab-Boushehri, P. Rajabi, and R. Mahmoudi, “Modeling location–allocation of emergency medical service stations and ambulance routing problems considering the variability of events and recurrent traffic congestion: A real case study,” Healthcare Analytics, vol. 2, Art. no. 100048, 2022, doi: 10.1016/j.health.2022.100048.

 This work is licensed under a Creative Commons Attribution 4.0 International License