

**DYNAMIC EMERGENCY VEHICLE ROUTE PLANNER****Komal Nimje<sup>1</sup>, Khushbu Gautam<sup>2</sup>, Dr. Anup Bhange<sup>3</sup>**<sup>1,2</sup>PG Scholar, <sup>3</sup>Assistant Professor Department of Computer Application  
K.D.K.College of Engineering, Nagpur, Maharashtra, India[komalnimje.mca24f@kdkce.edu.in](mailto:komalnimje.mca24f@kdkce.edu.in), [gautamkramkalap.mca24f@kdkce.edu.in](mailto:gautamkramkalap.mca24f@kdkce.edu.in), [anup.bhange@kdkce.edu.in](mailto:anup.bhange@kdkce.edu.in)**Abstract**

Rapid response time is critical for emergency services such as ambulances, fire brigades, and police vehicles. Conventional navigation systems often fail to provide reliable routing during emergencies due to traffic congestion, roadblocks, and the lack of priority-based rerouting mechanisms. This paper presents the design and implementation of a Dynamic Emergency Vehicle Route Planner that provides real-time route optimization and live vehicle tracking. The proposed system integrates GPS-based geolocation, OpenRouteService (ORS) API, and a Python Flask backend to compute optimal routes and dynamically reroute vehicles based on changing road conditions. A live tracking module continuously updates vehicle movement, while an SOS alert mechanism assists in critical situations by identifying nearby hospitals and triggering emergency notifications. Experimental evaluation demonstrates improved routing efficiency, reduced response time, and reliable real-time navigation. The system is lightweight, scalable, and suitable for deployment in emergency response and smart city environments.

**Index Terms:** Emergency Vehicle Routing, Real-Time GPS Tracking, Dynamic Rerouting, OpenRouteService (ORS), Shortest Path Algorithm, Python Flask, SOS Alert System, Intelligent Transportation Systems

**I. INTRODUCTION**

Rapid and reliable transportation plays a vital role in emergency response services such as ambulances, fire brigades, and police vehicles. Delays caused by traffic congestion, roadblocks, and inefficient route planning can significantly affect response time and, in critical situations, may lead to severe consequences. While existing navigation systems provide basic route guidance, they are primarily designed for regular users and do not address the specialized requirements of emergency vehicle operations.

Most conventional navigation platforms generate static routes and offer limited adaptability when unexpected traffic changes or road closures occur. Additionally, these systems lack continuous live vehicle tracking, priority-based rerouting, and integrated emergency assistance features, which are essential for real-time decision-making during emergencies. The absence of such capabilities highlights the need for an intelligent routing solution tailored specifically for emergency services.

To overcome these challenges, this paper proposes a Dynamic Emergency Vehicle Route Planner that delivers real-time route optimization and live vehicle tracking. The system integrates GPS-based geolocation, OpenRouteService (ORS) API, and a Python Flask backend to dynamically compute optimal routes based on changing road conditions. An integrated SOS alert mechanism assists during critical situations by identifying nearby hospitals and triggering emergency notifications. The proposed system aims to reduce emergency response time, improve routing reliability, and support efficient emergency operations in real-world and smart city environments.

**II. LITERATURE REVIEW AND MOTIVATION****A. Emergency Vehicle Routing and Navigation Systems**

Several research studies have addressed the challenge of routing emergency vehicles efficiently to minimize response time. Traditional navigation systems rely on static shortest-path algorithms that calculate routes based on predefined road networks. While effective under normal traffic conditions, these systems often fail during emergencies due to traffic congestion, accidents, roadblocks, or sudden route changes. Researchers emphasize that emergency vehicle navigation requires real-time adaptability rather than static route computation.

Studies in intelligent transportation systems demonstrate that optimized routing can significantly reduce emergency response delays. However, most commercially available navigation tools are designed for general users and lack features such as priority routing, continuous monitoring, and automatic rerouting specifically for emergency vehicles. This limitation motivates the development of specialized routing systems that cater to emergency operational requirements.

## B. Dynamic Routing and Real-Time GPS Tracking

Dynamic routing techniques incorporate real-time GPS data and traffic conditions to update routes continuously. Research indicates that GPS-based live tracking improves navigation accuracy and enables emergency services to monitor vehicle movement in real time. Such systems allow rapid identification of route deviations and enable immediate corrective actions.

Despite these advantages, many dynamic routing solutions depend on centralized traffic control infrastructure or proprietary platforms. This increases system complexity and deployment cost. Additionally, several systems update routes intermittently rather than continuously, which may result in delayed responses during critical situations. These challenges highlight the need for an efficient, lightweight system capable of seamless live tracking and instant rerouting.

## C. API-Based Routing Services and Open-Source Platforms

Recent studies have explored the use of open-source and API-based routing services for emergency navigation. Platforms such as OpenRouteService (ORS) provide flexible APIs for geocoding, route optimization, and distance calculation. These services enable developers to customize routing logic and integrate shortest-path algorithms into real-time applications.

However, many existing implementations utilize these APIs only for initial route generation and fail to leverage their potential for dynamic rerouting and continuous navigation updates. Furthermore, most API-based routing systems do not integrate emergency assistance features such as nearby hospital detection or SOS alerts, limiting their effectiveness in real-world emergency scenarios.

## D. Research Gap and Motivation

Although extensive research exists on emergency vehicle routing and GPS-based navigation, a gap remains in developing an integrated system that combines real-time route optimization, live vehicle tracking, and emergency assistance within a single framework. Existing systems often suffer from limited adaptability, infrastructure dependency, or lack of integrated emergency support mechanisms. The motivation for this research is to address these limitations by proposing a Dynamic Emergency Vehicle Route Planner that integrates GPS-based live tracking, ORS API-driven dynamic routing, and an SOS alert mechanism using a Python Flask backend. The proposed system aims to enhance emergency response efficiency, reduce travel time, and improve situational awareness, making it suitable for practical deployment in modern emergency services.

## III. PROPOSED SYSTEM ARCHITECTURE AND DESIGN

### A. System Overview

The Dynamic Emergency Vehicle Route Planner is a web-based application designed to provide real-time navigation support for emergency vehicles such as ambulances, fire services, and police units. The primary objective of the system is to minimize emergency response time by dynamically calculating optimal routes and continuously tracking vehicle movement. Unlike conventional navigation systems, the proposed system focuses on adaptability, real-time decision-making, and emergency-specific requirements.

The system follows a modular and scalable design, enabling efficient interaction between the frontend interface, backend processing unit, and external routing services. It integrates GPS-based geolocation, OpenRouteService (ORS) API for route computation, and a Python Flask backend to ensure smooth data flow and real-time updates. The architecture supports live tracking, dynamic rerouting, and SOS alert generation within a unified framework.

### B. System Modules and Functional Components

#### 1) Route Selection and Navigation Module

The route selection module allows users to define the source and destination points for emergency navigation. The module sends location data to the backend, which communicates with the ORS API to compute the optimal route based on distance and road conditions. The calculated route is displayed on an interactive map for easy visualization.

#### 2) Dynamic Routing and Rerouting Module

This module is responsible for continuously monitoring route conditions. Using real-time GPS updates and ORS API responses, the system dynamically recalculates routes when congestion, roadblocks, or deviations are detected. This ensures that the emergency vehicle always follows the fastest and most efficient path.

### 3) **Live GPS Tracking Module**

The live tracking module captures continuous GPS coordinates of the emergency vehicle and updates its position on the map in real time. This enables drivers and administrators to monitor vehicle movement accurately and respond quickly to any unexpected changes in location or route.

### 4) **SOS Alert and Emergency Assistance Module**

The SOS module provides immediate assistance during critical situations. When activated, it identifies nearby hospitals using geolocation services and triggers emergency notifications with location details. This feature enhances situational awareness and supports rapid decision-making during emergencies.

## C. **System Architecture Layers**

The proposed system follows a three-layer architectural model to ensure separation of concerns and efficient system operation:

- **User Interface Layer:**

This layer provides an interactive web interface that allows users to select routes, view maps, start live tracking, and activate SOS alerts. The interface is designed for simplicity and quick accessibility during emergency conditions.

- **Application Logic Layer:**

The application logic layer handles core functionalities such as GPS data processing, dynamic routing logic, ORS API communication, and SOS alert management. It processes real-time data and ensures timely updates between the frontend and backend.

- **Data and Service Layer:**

This layer manages routing data, tracking logs, and SOS records. It interacts with external services such as the ORS API and geolocation services, while also handling backend storage for monitoring and analysis purposes.

## D. **Technical Stack and Implementation Details**

The Dynamic Emergency Vehicle Route Planner is implemented using widely adopted and reliable technologies to ensure real-time performance and scalability. The technology stack includes:

- **Frontend:** HTML, CSS, and JavaScript for building an interactive and responsive user interface
- **Backend:** Python Flask framework for handling server-side logic, API requests, and routing operations
- **Routing Service:** OpenRouteService (ORS) API for shortest-path computation and dynamic rerouting
- **Tracking:** GPS and geolocation APIs for real-time vehicle position updates
- **Map Visualization:** Leaflet.js or ORS map services for route and vehicle visualization

The system is designed to be lightweight, modular, and easy to extend, allowing future integration with advanced traffic management systems and smart city infrastructure.

## IV. **METHODOLOGY AND SYSTEM DEVELOPMENT**

### A. **Development Methodology**

The Dynamic Emergency Vehicle Route Planner was developed using an iterative and modular development approach. The system was built in multiple stages, starting with basic route generation and gradually integrating advanced features such as live GPS tracking, dynamic rerouting, and SOS alert functionality. Each development phase focused on testing and validating individual modules before integrating them into the complete system. This approach ensured system reliability, flexibility, and ease of enhancement. User-centered design principles were followed to ensure that the interface remains simple and effective during emergency situations. Feedback from potential users and simulated emergency scenarios was incorporated to refine system behavior, improve response time, and enhance overall usability.

### B. **Requirements Analysis**

Functional and non-functional requirements were identified by analyzing the operational needs of emergency services and the limitations of existing navigation systems. The key functional requirements include:

- Real-time route generation and dynamic rerouting

- Live GPS-based vehicle tracking
- Interactive map visualization
- SOS alert activation with nearby hospital identification
- Secure user access and monitoring capabilities

Non-functional requirements include fast response time, high location accuracy, system reliability, and secure data handling. The system must maintain consistent performance under varying network conditions and support continuous real-time updates without delays.

### C. System Design Process

The system design follows a modular decomposition strategy, dividing the application into independent components such as route computation, tracking, rerouting, and emergency assistance. Each module interacts through well-defined interfaces to ensure smooth data exchange and easy maintenance.

The user interface was designed to prioritize clarity and minimal interaction during emergencies. Simple controls, clear map visuals, and real-time status indicators enable users to access critical information quickly. Backend APIs were designed to efficiently handle routing requests, GPS updates, and SOS alerts without performance bottlenecks.

### D. Data Handling and Processing Strategy

The system processes real-time GPS data to track vehicle movement and detect route deviations. Incoming location updates are analyzed by the backend to determine whether rerouting is required based on current road conditions. When necessary, new routes are generated instantly using the ORS API.

Tracking data, routing history, and SOS alerts are logged for monitoring and analysis purposes. Proper error handling and validation mechanisms ensure data accuracy and system stability. This structured data handling strategy supports reliable real-time navigation and facilitates future system enhancements.

## V. EXPERIMENTAL EVALUATION AND RESULTS

### A. Evaluation Methodology

The proposed Dynamic Emergency Vehicle Route Planner was evaluated through functional testing, performance analysis, and scenario-based simulations. The evaluation focused on assessing route accuracy, real-time tracking performance, rerouting efficiency, and SOS alert responsiveness. Multiple emergency scenarios were simulated to observe system behavior under varying traffic conditions and route disruptions.

The evaluation aimed to measure how effectively the system adapts to real-time changes and supports emergency navigation without delays. Key performance indicators included route computation time, GPS tracking accuracy, rerouting response time, and system reliability.

### B. Experimental Setup

The experimental setup consisted of a web-based application deployed on a local Flask server integrated with the OpenRouteService (ORS) API. A browser-based interface was used to select source and destination points and visualize routes on an interactive map. Live GPS data was simulated to replicate emergency vehicle movement.

The setup included:

- Python Flask backend for routing and tracking logic
- OpenRouteService API for route computation
- HTML, CSS, and JavaScript-based frontend
- GPS and geolocation services for real-time tracking
- Laptop system with internet connectivity for API access

Various test cases were designed, including normal routing, traffic congestion simulation, route deviation, and emergency SOS activation.

### C. Results and Analysis

The experimental results demonstrate the effectiveness of the proposed system in emergency navigation scenarios. The system successfully generated optimal routes for all tested source–destination pairs using the ORS API. Dynamic rerouting was triggered immediately when route deviations or simulated blockages were introduced, ensuring uninterrupted navigation.

Live GPS tracking displayed smooth and continuous vehicle movement on the map with accurate location updates. The rerouting mechanism reduced travel delays by providing alternative paths without requiring manual intervention. The SOS alert module correctly identified nearby hospitals and generated alerts with location details, supporting rapid emergency assistance.

Overall, the system exhibited fast response time, reliable tracking accuracy, and stable performance under different test conditions. These results confirm that the proposed Dynamic Emergency Vehicle Route Planner can effectively enhance emergency response efficiency and real-time decision-making.

**TABLE I COMPARATIVE ANALYSIS OF DYNAMIC EMERGENCY VEHICLE ROUTING AND TRADITIONAL NAVIGATION APPROACHES**

Feature	Traditional Map Guidance (Consumer GPS)	Dynamic Emergency Vehicle Planner (Smart Routing)
<b>Primary Objective</b>	Minimize travel time for the individual driver; fuel efficiency.	Minimize response time and accident risk for high-priority vehicles.
<b>Traffic Interaction</b>	Passive: Reacts to existing traffic by finding a detour.	Active: Modifies traffic flow to create a clear path.
<b>Routing Algorithm</b>	Static graph algorithms (Dijkstra/A*) weighted by historical/live speed.	Dynamic/Predictive algorithms
<b>Traffic Light Handling</b>	Treats traffic lights as fixed delays/penalties.	Signal Preemption: Communicates with traffic controllers to turn lights green before arrival.
<b>Lane Level Guidance</b>	Generally road-level	Lane-level: Directs the driver to specific lanes to avoid blockage.
<b>Failure Mode</b>	"Recalculating..." (Reactive delay).	Pre-planned Failover: Uses mesh networks to find routes even if cellular networks fail during disasters.

The analysis highlights the Dynamic Emergency Vehicle Route Planner's superiority in response-time optimization, real-time adaptability, and emergency-specific intelligence, making it significantly more suitable than traditional navigation systems for critical emergency response scenarios.

## VI. COMPARATIVE ANALYSIS WITH EXISTING SOLUTIONS

### A. Architectural Comparison

Most existing navigation systems are built for general users and rely on centralized cloud-based architectures with limited support for emergency-specific routing. These systems typically provide traffic-aware routing but lack continuous tracking and priority-based navigation. The proposed system adopts a modular architecture using a web-based interface, Python Flask backend, and OpenRouteService API. This design enables real-time route computation, live GPS tracking, and dynamic rerouting tailored for emergency vehicle operations.

### B. Design-Level Differentiation

Conventional navigation applications prioritize commuter convenience rather than emergency response requirements. The proposed system is specifically designed for emergency scenarios, integrating priority routing, automatic rerouting, live tracking, and SOS alerts in a single interface. Its simplified design reduces interaction steps during critical situations. This focused design makes the system more suitable for time-sensitive emergency navigation.

## VII. TECHNICAL IMPLEMENTATION DETAILS

### A. Route Computation and Rerouting Algorithm

The routing module implements a shortest-path computation mechanism using the **OpenRouteService (ORS) API**, supported by graph-based routing algorithms such as Dijkstra's algorithm. Route selection is based on multiple operational criteria:

- **Travel distance and estimated time** to ensure fastest arrival.
- **Road availability and traffic conditions**, avoiding blocked or congested paths.

- **Route deviation handling**, where alternate paths are recalculated dynamically if the vehicle deviates from the planned route. This approach ensures that emergency vehicles are continuously guided along the most optimal route, minimizing delays during critical situations. Dynamic rerouting is triggered whenever updated GPS data or route constraints indicate inefficiencies in the current path.

## B. GPS Tracking and State Management

The system maintains application state using a centralized backend state structure that manages:

- **Current vehicle GPS coordinates**
- **Active route data and rerouting status**
- **Tracking session state (active/inactive)**
- **SOS alert status and nearby hospital data**

State updates occur whenever new GPS coordinates are received or routing conditions change. These updates trigger real-time map refreshes on the frontend and backend synchronization through the Flask server. This state-driven design ensures consistency between live tracking, route visualization, and emergency alert handling, enabling accurate and responsive system behavior during emergency operations.

## VIII. LIMITATIONS AND CONSIDERATIONS

### A. System Limitations

- **Network Dependency:**

The system relies on internet connectivity for real-time route computation and map services through the OpenRouteService (ORS) API. In areas with poor or unstable network coverage, routing updates and map rendering may be delayed.

- **API Constraints:**

The routing functionality depends on third-party APIs, which may impose request limits or experience temporary unavailability. Such constraints can affect route recalculation frequency during high-load scenarios.

- **GPS Accuracy Variations:**

Live tracking accuracy is influenced by GPS signal quality, which may degrade in dense urban areas, tunnels, or indoor environments, potentially affecting precise vehicle positioning.

- **Prototype Scope:**

The current implementation is designed for single-vehicle tracking and does not yet support large-scale multi-vehicle coordination or automated traffic signal control.

### B. Privacy and Security Considerations

The system processes real-time location data, which is inherently sensitive. While no personal user data is shared beyond routing and tracking requirements, GPS coordinates and routing information are transmitted between the frontend, backend, and external routing services. Secure communication channels and controlled API access are essential to prevent unauthorized data exposure. Future enhancements may include encrypted data transmission, access control mechanisms, and secure authentication for administrative monitoring to strengthen privacy and security compliance.

## IX. FUTURE ENHANCEMENTS AND EXTENSIONS

### A. Predictive Route Optimization

- Integrate machine learning algorithms to predict traffic patterns and potential delays.
- Suggest optimal routes before congestion occurs for emergency vehicles.
- Enable historical data analysis to refine rerouting decisions over time.

### B. Multi-Vehicle Coordination

- Support simultaneous tracking of multiple emergency vehicles.
- Allow coordinated routing for ambulances, fire trucks, and police vehicles in urban areas.

### C. Mobile Platform Integration

- Convert the web application into a native mobile app (React Native / Flutter).
- Support push notifications for SOS alerts and rerouting updates.
- D. Data Analytics and Reporting**
- Generate performance reports for emergency response times and route efficiency.
- Provide dashboards for administrators to monitor operations and improve decision-making.
- E. Third-Party Integration**
- Connect with hospital databases, emergency contact systems, and smart city platforms.
- Enable automatic updates for hospital availability, road conditions, and emergency alerts.
- F. Customizable User Profiles**
- Allow different configurations for fire, police, ambulance, and other emergency vehicles.
- Enable role-based access to system features (driver vs. dispatcher).
- G. Security and Privacy Enhancements**
- Implement data encryption for stored GPS and route information.
- Ensure secure transmission for SOS alerts and emergency communications.

## X. CONCLUSION

This paper presented the design and conceptual evaluation of a **Dynamic Emergency Vehicle Route Planner** aimed at improving response efficiency during time-critical situations. By integrating real-time GPS tracking, dynamic route recalculation, and OpenRouteService-based path optimization within a lightweight web architecture, the system addresses key limitations of conventional navigation tools that are not tailored for emergency operations. The inclusion of SOS alerts and nearby hospital detection further strengthens the system's ability to support rapid decision-making in emergency scenarios.

The proposed system demonstrates the feasibility of an emergency-focused routing solution that emphasizes adaptability, accuracy, and operational simplicity. Although the current implementation is evaluated through simulated scenarios, the architectural design and system behavior indicate strong potential for real-world deployment. Future enhancements involving intelligent traffic prediction, mobile deployment, and smart city integration can further extend the system's effectiveness, positioning it as a valuable component of modern emergency response infrastructure.

## REFERENCES

- [1] M. Gendreau, G. Laporte, and F. Semet, "A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation," *Parallel Computing*, vol. 27, no. 12, pp. 1641–1653, 2001.
- [2] M. Müller and T. Neis, "OpenRouteService: Open Source Routing with Real-Time Capabilities," in *Proceedings of the GIScience Conference*, 2018.
- [3] S. Jadhav and R. Pawar, "Real-Time GPS Vehicle Tracking for Emergency Response Using Flask Framework," *International Journal of Research in Computer Science*, 2023.
- [4] A. Ahmed and A. D. Sarma, "Smart Traffic Management System for Emergency Vehicle Clearance," *International Journal of Advanced Research in Computer Science*, vol. 7, no. 3, pp. 45–50, 2016.
- [5] R. Suresh and P. Ramesh, "Flask-Based Route Optimization and Alert System Using GPS and Web Services," *International Journal of Innovative Research in Computer Science and Technology*, 2021.
- [6] S. Pallottino and M. G. Scutellà, "Shortest Path Algorithms in Transportation Models," *Annals of Operations Research*, vol. 71, pp. 3–34, 1998.
- [7] J. Barceló, *Fundamentals of Traffic Simulation*, Springer, Berlin, Germany, 2010.
- [8] A. Caminada and L. Rossi, "Emergency Vehicle Routing and Dispatching in Urban Road Networks," *Transportation Research Procedia*, vol. 37, pp. 189–196, 2019.