International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Smart Emergency Response System (SERS)

Savita Masali, Anil K, Anjana B R, Asad Fathima, Ganesh Prasad K
Department of Computer Science, Sambhram Institute of Technology, Bengaluru, India
ORCID (Anil K): https://orcid.org/0009-0000-0026-4047

Abstract - During a medical emergency, the people reporting the incident, ambulance staff, and hospitals need to exchange correct information quickly. Delays may occur if the location is unclear, an ambulance is not available, or a hospital cannot provide the required care. This paper presents the Smart Emergency Response System (SERS), a web-based prototype intended to bring these steps into one workflow. A user can submit incident details and GPS coordinates, with optional time- and location-tagged evidence. Ambulance selection considers emergency priority, availability, and distance. Hospital options are ranked using estimated travel time, distance, doctor availability, ICU facilities, emergency beds, and capacity; a separate readiness check is applied before confirmation. The design also includes live status sharing and a simulated green corridor to demonstrate traffic priority. The corridor does not control real traffic lights. The prototype will be assessed with test scenarios using dispatch time, arrival time, hospital confirmation time, ETA error, communication delay, and manual communication steps. Results must be reported only after testing is completed.

Index Terms - Smart Emergency Response System, ambulance dispatch, GPS tracking, hospital recommendation, emergency coordination, traffic simulation.

I. INTRODUCTION

Emergency services often need information from several groups at the same time. A citizen reports an incident, an ambulance team needs the correct location, and a hospital must know whether it can receive the patient. Police and traffic authorities may also be involved. When these groups use separate communication channels, updates can be missed or repeated. That can make it harder to decide which ambulance to send and which hospital to contact.

SERS is proposed as a shared platform for the main coordination tasks. A user enters the incident type and details and shares a GPS location. The system checks the request and looks for an available ambulance based on priority, availability, and distance. Once an ambulance is assigned, its location and status can be shared with authorized users. Hospital suggestions consider travel time and available medical facilities rather than distance alone. The selected hospital is checked against the recorded readiness information before confirmation.

The project also contains a simulated green corridor. It is included to show where traffic priority could fit into the workflow; it does not change real traffic signals. AI-assisted functions are intended for user interaction and communication. Dispatch selection, hospital ranking, and readiness checks use defined rules and scoring, so the paper does not describe every part of the system as an AI decision. The present work describes a prototype design and its planned evaluation, not a proven improvement in real emergency outcomes.

II. RELATED WORK

Laksono et al. use the Analytic Hierarchy Process and a dynamic shortest-path method to help choose an ambulance and plan a route [1]. Their work shows that dispatch can be treated as a decision involving more than one factor. SERS follows this general idea and adds a shared workflow for location updates, hospital suggestions, and readiness checks.

Dar et al. describe an accident-detection and response approach that uses smartphone sensors, GPS, and fog computing to send emergency notifications while limiting communication delay [2]. That work mainly addresses detection and notification. SERS is intended to support later steps as well, including ambulance assignment, tracking, and hospital selection.

Poulton et al. study ambulance movement using historical GPS information and road travel speeds [3]. Their work draws attention to road conditions when estimating arrival times. SERS also uses location updates and route information to support tracking and ETA estimation, although its estimates will need to be checked against recorded test cases.

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

The NAJDA system supports emergency reporting, location sharing, ambulance tracking, and hospital notification [4]. The study in [5] also considers patient severity, ambulance assignment, estimated arrival time, and hospital availability. These approaches show that ambulance movement and hospital coordination are linked parts of an emergency response.

Other work considers edge machine learning for accident detection and dispatch [6], as well as ambulance routing and traffic coordination using a simulated green corridor [7]. SERS uses AI-assisted functions for communication and user support, while dispatch and hospital decisions follow defined rules. Its green corridor is a demonstration only. The system therefore combines ideas from prior work in a single proposed workflow, but its usefulness must be established through testing rather than assumed from the design.

III. PROBLEM STATEMENT AND OBJECTIVES

The main problem addressed by SERS is the difficulty of coordinating emergency information across citizens, ambulance teams, hospitals, and supporting authorities. Four practical issues are considered. First, a report must include a usable incident location. Second, the ambulance selected should be available and suitable for the reported priority. Third, the closest hospital may not have the doctor, ICU facility, emergency bed, or capacity that is needed. Finally, people involved in the response need updated location and status information.

SERS is designed around the following objectives:
1. Record an emergency report with GPS location and optional location-tagged evidence.
2. Select an available ambulance using incident priority, ambulance availability, and distance.
3. Share ambulance location and status updates with authorized users.
4. Rank hospitals using travel time, distance, medical staff, facilities, beds, and capacity.
5. Check the recorded readiness information before confirming a hospital.
6. Demonstrate the role of traffic priority using a simulated green corridor.
7. Provide user-facing communication support, including voice guidance where available.
8. Evaluate dispatch time, arrival time, hospital confirmation time, ETA error, and communication delay.

IV. PROPOSED SYSTEM AND METHODOLOGY

The process begins when a citizen opens an emergency report, enters the incident details, chooses an emergency category, and shares the current GPS location. Photos or other evidence can be attached when needed. The backend validates the submitted information and checks the reported priority. It then compares the incident with available ambulances and selects a suitable option using the configured rules. The assigned driver receives the incident location and relevant details.

After dispatch, GPS updates are used to show the ambulance's current position and status. Route information supports an estimated arrival time. Hospital options are ranked using distance, estimated journey time, doctor availability, ICU facilities, emergency beds, and overall capacity. Before the destination is confirmed, the readiness module checks the stored information for the required services. If a hospital does not meet the recorded requirements, another option can be reviewed.

The traffic-coordination feature represents a green corridor in the interface to show how priority could be included in the process. It is a simulation and is not connected to live traffic lights. The prototype uses React and Vite for the interface, Node.js and Express for backend operations, MongoDB with Mongoose for stored records, and Socket.IO for live status updates. Planned tests will use multiple emergency scenarios. Any outcomes produced by simulated cases will be labelled as simulation results.

V. SYSTEM ARCHITECTURE

The architecture has three main parts: the frontend, backend services, and database. Citizens and ambulance drivers use the interface to submit reports and follow ambulance progress. Hospitals, police, traffic authorities, and administrators use role-specific functions. The backend processes reports, applies ambulance dispatch rules, ranks hospitals, and checks readiness. MongoDB stores user details, emergency reports, ambulance information, and hospital records. Mongoose manages database operations, while Socket.IO distributes live location and status updates to authorized users.

The architecture diagram in the source paper presents the main components rather than a deployment configuration. Socket.IO provides the real-time update channel. Access permissions limit each role to the functions and data needed for its responsibilities. The actual availability and quality of hospital data will affect how useful the recommendations are, so those records must be maintained and checked.

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

VI. SYSTEM WORKFLOW

The workflow moves from reporting to dispatch, tracking, hospital confirmation, and status closure. After a citizen submits a report, the system validates the details and checks priority and ambulance availability. The assigned driver receives the report and authorized users can follow the ambulance on a map. GPS and route information are used to estimate arrival time. Hospital options are scored and the selected hospital's recorded readiness is checked. The incident status can be updated when the ambulance reaches the incident and again after reaching the hospital.

The order of steps may change in a real emergency because ambulance and hospital information can change while the request is being handled. The prototype will therefore be checked for missing updates, unavailable ambulances, incomplete hospital information, and communication failures. The simulated corridor demonstrates the workflow only; no real traffic signal is controlled.

VII. FUNCTIONAL MODULES AND DATA MANAGEMENT

SERS is divided into connected modules so that each part can be tested separately. The reporting module stores incident details and GPS location. The dispatch module uses priority, availability, and distance to select an ambulance. The tracking module shares the current location and status after dispatch. The hospital recommendation module scores possible destinations using the factors listed earlier, while the readiness module checks the saved information before confirmation. Traffic coordination is represented by a simulated green corridor. The user-communication module supports status messages, voice guidance, and other interface assistance where those functions are available. The project does not claim that these features replace the ambulance control room or professional medical judgment.

MongoDB holds user records, emergency reports, ambulance details, and hospital data. Mongoose is used to read and update those records. Node.js and Express handle requests and backend processing. Socket.IO sends live ambulance updates to authorized users. Role-based access separates the views for citizens, drivers, hospitals, police, traffic authorities, and administrators. The implementation should also validate submitted values and handle cases in which location or availability data are missing.

VIII. IMPLEMENTATION AND PRELIMINARY TESTING

The proposed prototype uses React and Vite on the frontend and Node.js with Express on the backend. MongoDB and Mongoose provide data storage and access, while Socket.IO supports updates. GPS data provide the incident and ambulance locations. The dispatch, hospital-ranking, and readiness stages apply defined rules. Testing should be performed on each module before checking the complete workflow.

Table I lists basic expected outputs. These are expected behaviours, not measured results. During testing, the team should record dispatch time, ambulance arrival time, hospital selection and confirmation time, ETA error, communication delay, and the number of manual communication steps for each scenario. Enough test cases are needed before comparing performance or drawing conclusions.

Basic Functional Test Expectations:
- Emergency reporting: Incident details and location saved.
- GPS tracking: Latest available location shown.
- Ambulance dispatch: Suitable available ambulance assigned.
- Hospital selection: Options ranked and readiness checked.
- Real-time updates: Location and status shared with authorized users.
- Traffic simulation: Priority corridor shown in the interface.

Results from simulated emergencies must be labelled as simulation-based. Until tests are completed and results recorded, the paper should not claim a particular response-time reduction, ETA accuracy, or clinical benefit. Testing is intended to find technical problems and show how consistently the modules work together.

IX. SECURITY, PRIVACY, AND RESPONSIBLE USE

The system may process personal details, GPS positions, photos, and emergency records. Access should therefore be limited according to user role, and API inputs should be validated. Secure communication should be used to reduce the risk of unauthorized access. Location data should be collected only for the emergency-related purpose and should not be available to every user. Hospital and ambulance records also need regular updates because availability can change quickly.

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

SERS is a coordination aid, not a replacement for medical professionals or official emergency services. The green corridor remains a simulation and must not be presented as an integration with public traffic infrastructure. Further security and usability testing is required before any consideration of real-world use.

X. DISCUSSION

SERS brings emergency reporting, ambulance assignment, tracking, hospital recommendation, readiness checking, and traffic simulation into one workflow. Its main design choice is to consider more than the nearest ambulance or hospital: emergency priority and availability affect dispatch, while travel time and recorded medical facilities affect hospital ranking. Live status sharing is intended to reduce the need for separate calls and manual updates, but that benefit has not yet been measured.

The prototype also has limitations. Location accuracy depends on the device and GPS conditions. ETA estimates depend on the quality of route information. Hospital recommendations are only as useful as the facility and capacity records available to the system. An entry may become outdated after it is saved. For these reasons, the team must test different cases, record failures as well as successful flows, and report measured results separately from expected behaviour.

XI. CONCLUSION AND FUTURE WORK

This paper describes SERS, a web-based prototype for coordinating emergency reports, ambulance dispatch, GPS tracking, hospital recommendations, and hospital readiness checks. It also includes a simulated green corridor to demonstrate where traffic priority could fit into the process. The design provides one workflow for sharing incident and ambulance status information, but its practical effect has not yet been established through measured results.

Future work includes running a wider set of test scenarios, improving ETA estimates with better route and location information, and connecting to updated hospital records when permission and suitable data access are available. Voice guidance and communication features can also be improved. Any connection to real traffic systems would require authorization from the responsible agencies. Real-world testing and stronger security checks would be needed before practical deployment.

REFERENCES

[1] P. Laksono, S. R. Wulan, S. H. Supangkat, and W. D. Sunindyo, “AHP and Dynamic Shortest Path Algorithm to Improve Optimum Ambulance Dispatch in Emergency Medical Response,” in Proc. International Conference on ICT for Smart Society (ICISS), 2017.

[2] B. K. Dar, M. A. Shah, S. U. Islam, C. Maple, M. Musaddiq, and S. Khan, “Delay-Aware Accident Detection and Response System Using Fog Computing,” IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2910862.

[3] M. Poulton, A. Noulas, D. Weston, and G. Roussos, “Modeling Metropolitan-Area Ambulance Mobility Under Blue Light Conditions,” IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2018.2886852.

[4] S. E. Alhammadi, R. M. Almheiri, and A. Wani, “NAJDA: A Smart Ambulance System for Real-Time Emergency Response and Inclusive Healthcare Support,” in Proc. 3rd International Conference on Foundation and Large Language Models (FLLM), 2025, doi: 10.1109/FLLM67465.2025.11390935.

[5] K. P., P. S., and R. S., “A Smart MedCare Scheduling with Real-Time Hospital Acceptance and Ambulance Dispatch,” in Proc. International Conference on AI-Driven Smart Systems and Ubiquitous Computing (ICAUC), 2026, doi: 10.1109/ICAUC68182.2026.11441220.

[6] S. R., S. S. M., and M. Sangeetha, “AI-Powered Ambulance Dispatch and Accident Detection System Using Edge ML,” in Proc. IEEE International Conference on Intelligent Technologies for a Sustainable and Inclusive Future (ICITSIF), 2026, doi: 10.1109/ICITSIF69060.2026.11608391.

[7] C. K., P. K. R., R. Hosur, M. K. P., M. S., and R. Pathi, “AI-Driven Smart Ambulance and Dynamic Traffic Control System for Optimized Emergency Response,” in Proc. Control Instrumentation System Conference (CISCON), 2025, doi: 10.1109/CISCON66933.2025.11337425.

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