© Volume 2, Issue 10, October 2026 | JATIR
e-ISSN: 2839-5786, p-ISSN: 3139-8464

Smart Emergency Intelligence System (Seis)

[1]Vijay V, [2]Dr. K. Adisesha
[1]IV Sem Student, Department of Computer Science and Applications S.E.A. College of Science, Commerce and Arts, Bangalore
[2]Associate Professor, Department of Computer Science and Applications S.E.A. College of Science, Commerce and Arts, Bangalore

***Abstract*—Emergencies can happen at any time, and the first few minutes are often very important. During situations such as road accidents, medical emergencies, fires, or other dangerous incidents, people may not have enough time to explain their situation or provide their exact location. At the same time, responders may not receive complete information quickly enough to take action.**

To address this problem, Guardian – Smart Emergency Intelligence System is proposed as a centralized platform that connects citizens, emergency responders, and administrators. The system allows a citizen to select the type of emergency and send an SOS request along with important information such as location and contact details. The request is then processed by the backend and shared with suitable nearby responders through real-time notifications.

Responders can view the emergency details, accept the request, navigate to the location, and update the progress of the response. Administrators can monitor active emergencies, responders, and the overall status of incidents from a centralized dashboard.

The main purpose of Guardian is to make emergency communication more organized, faster, and easier to manage. The proposed system also provides a base for future technologies such as artificial intelligence, automatic accident detection, wearable-device integration, predictive analytics, and offline emergency communication.

***Index Terms*—Emergency Response, SOS, GPS, Real-Time Notification, Location Tracking, Artificial Intelligence, Emergency Management, Mobile Application.**

I. INTRODUCTION

Emergencies are situations where every minute can matter. When a person faces an accident, medical problem, fire, crime, or another critical situation, getting help quickly can be difficult. In many cases, the person may be frightened, injured, or unable to clearly explain what happened and where help is required.

Traditional emergency communication mainly depends on phone calls and manual information sharing. Although these methods are useful, they may not always provide all the required information in an organized manner. For example, the caller may not know the exact location, the responder may need additional information, or the emergency may not be properly coordinated between different people.

Recent research has shown that technologies such as GPS, mobile applications, IoT, cloud computing, and artificial intelligence can be useful in emergency management. Studies published in recent years have explored location-based emergency applications, real-time alert systems, intelligent emergency prioritization, and dynamic responder dispatching.

Based on these ideas, Guardian is proposed as a platform that brings the major parts of an emergency-response process together.

The system mainly involves three users:

- Citizen – reports an emergency.
- Responder – receives and handles the emergency.
- Administrator – monitors and coordinates emergency activities.

The goal of Guardian is not simply to provide an SOS button. Instead, it aims to create a complete flow from reporting an emergency to assigning a responder and closing the incident.

II. PROBLEM STATEMENT

During an emergency, people may face several difficulties while trying to get help.
Some common problems are:

- The exact location of the emergency may not be communicated correctly.
- Important information may have to be collected manually.
- People may not know which emergency service they should contact.
- Responders may not receive complete information about the incident.
- Multiple emergencies may happen at the same time.
- It can be difficult to coordinate citizens, responders, and administrators.
- Tracking the progress of an emergency manually can be difficult.
- Emergency records may be stored across different systems.

Therefore, there is a need for a centralized system that can collect emergency information, identify the location of the incident, notify suitable responders, and allow the emergency to be monitored until it is resolved.

III. LITERATURE REVIEW

Several recent studies have explored how modern technologies can improve emergency response. Gabella et al. (2024) developed and evaluated a mobile application that uses caller location information to support emergency medical services. Their research shows how location information from mobile devices can help improve the handling of emergency calls.
Mohsin and Muyeed (2024) proposed an IoT-based Smart Emergency Response System that can monitor different emergency conditions using sensors and communicate alerts to appropriate services. Their work demonstrates how connected devices can support emergency detection and response.

Sako et al. (2024) proposed a situation-aware emergency response system that focuses on monitoring, classifying, notifying, and visualizing emergency situations. This shows the importance of having a structured system instead of relying only on individual communication methods.

Research published in 2025 has also investigated the use of machine learning for emergency prioritization. Such systems can consider information such as urgency, distance, impact, and available resources while processing emergency requests.

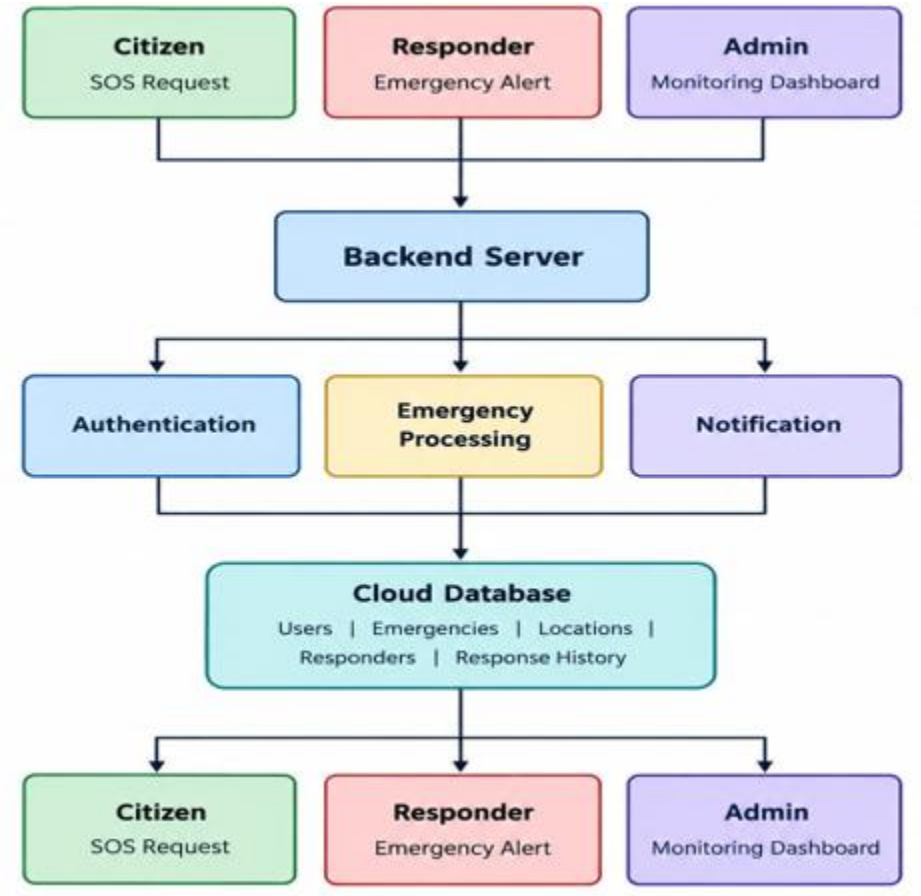
Zhang et al. (2025) studied a real-time IoT-based public safety system using sensors, edge computing, cloud services, and notification mechanisms. Their research highlights the importance of real-time communication in public safety applications.
Other recent research has also examined intelligent emergency dispatching by considering factors such as traffic and response conditions.

Research Gap
The existing studies demonstrate that individual technologies such as GPS, IoT, AI, cloud computing, and real-time communication can be useful for emergency management. However, many systems focus on a particular part of the emergency process.
Guardian attempts to combine these concepts into a single platform where:
Citizen → Emergency Request → Location → Responder Notification → Response → Monitoring → Resolution
This provides a complete and easy-to-understand emergency workflow.

IV. PROPOSED SYSTEM ARCHITECTURE

Guardian follows a multi-layer architecture in which the citizen application, responder application, administrator dashboard, backend services, and database communicate with each other.
High-Level Architecture

GUARDIAN SYSTEM



The architecture is divided into different parts.

Presentation Layer
This layer contains the interfaces used by citizens, responders, and administrators.

Application Layer
This layer handles the main operations of the system, including emergency creation, responder assignment, status updates, and user management.

Real-Time Communication Layer
This layer is responsible for delivering emergency notifications and updating emergency status.

Location Layer
This layer handles the location information received from the citizen's device.

Database Layer
This layer stores users, emergency requests, responders, locations, and response history.

V. SYSTEM COMPONENTS

5.1 Citizen Module

The citizen is the person who reports an emergency.
The citizen can:

- Create an account.
- Log in securely.
- Select an emergency type.
- Send an SOS request.
- Share their current location.
- Provide emergency information.
- View the status of an emergency.
- Track response progress.
- View previous emergency records.

When an SOS request is created, the system can collect information such as:
User ID
Emergency Type
Contact Number
Latitude
Longitude
Date and Time
Description
Emergency Status
This information gives the responder a basic understanding of the situation before reaching the location.

5.2Responder Module

The responder is responsible for handling the emergency.
When a suitable emergency request is received, the responder can:

- Receive an emergency notification.
- View the emergency type.
- View the citizen's details.
- View the emergency location.
- Accept the emergency.
- Navigate to the location.
- Update the response status.
- Close the emergency after resolution.

The response can follow the following process:
Reported → Dispatched → Accepted → Arrived → Resolved → Closed

This provides a clear record of what happened during the emergency.

5.3 Administrator Module

The administrator or command center provides centralized monitoring.

The administrator can:

- View active emergencies.
- Monitor responders.
- Assign responders.
- View emergency details.
- Monitor response status.
- Manage users.
- Manage responders.
- View emergency history.
- Generate reports.

The administrator acts as the central point for monitoring the overall emergency-response process.

5.4 Location Services

Location is one of the important parts of Guardian.

When the citizen sends an SOS request, the application can obtain the device's geographical location using GPS or available location services.

The location information can include:

- Latitude
- Longitude
- Location accuracy
- Timestamp

This information helps the responder understand where assistance is required.

5.5 Real-Time Notification System

A real-time notification system is required so that responders can receive emergency requests without depending on manual communication.

The basic process is:

Citizen sends SOS
↓
Backend receives request
↓
Emergency information is processed
↓
Nearby/available responders are identified
↓
Notification is sent

↓

Responder accepts the request

↓

Status is updated

This approach can reduce unnecessary communication steps and provide responders with structured information.

VI. METHODOLOGY

The development of Guardian follows a step-by-step software development process.

Step 1 – Requirement Analysis

The requirements of citizens, responders, and administrators are first identified.

Step 2 – System Design

The application architecture, database, user interfaces, APIs, and emergency workflow are designed.

Step 3 – User Authentication

A role-based authentication system is implemented so that citizens, responders, and administrators can access their respective features.

Step 4 – Emergency Request

The citizen selects the emergency category and activates the SOS function.

Step 5 – Location Collection

The application obtains the user's current location and attaches it to the emergency request.

Step 6 – Backend Processing

The backend receives the request, validates the information, and stores the emergency record.

Step 7 – Responder Selection

The system identifies suitable responders based on factors such as:

- Location
- Availability
- Emergency type
- Current assignments

In future versions, machine-learning techniques can be used to make this process more intelligent.

Step 8 – Notification

The selected responders receive the emergency notification.

Step 9 – Response Tracking

The responder updates the emergency status while handling the incident.

Step 10 – Closure

Once the emergency has been handled, the responder closes the case. The complete record is then stored for future reference and analysis.

VII. SYSTEM IMAGE

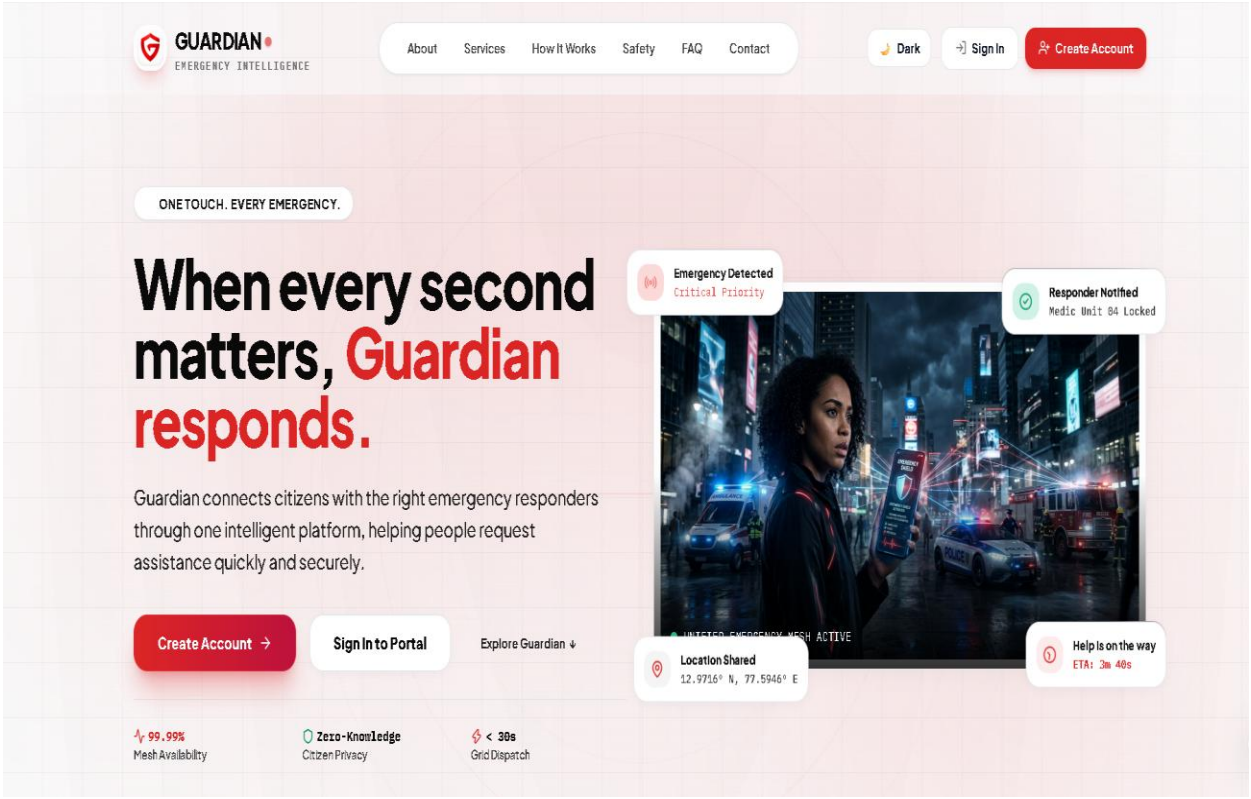


Figure 1: Guardian Emergency Landing page

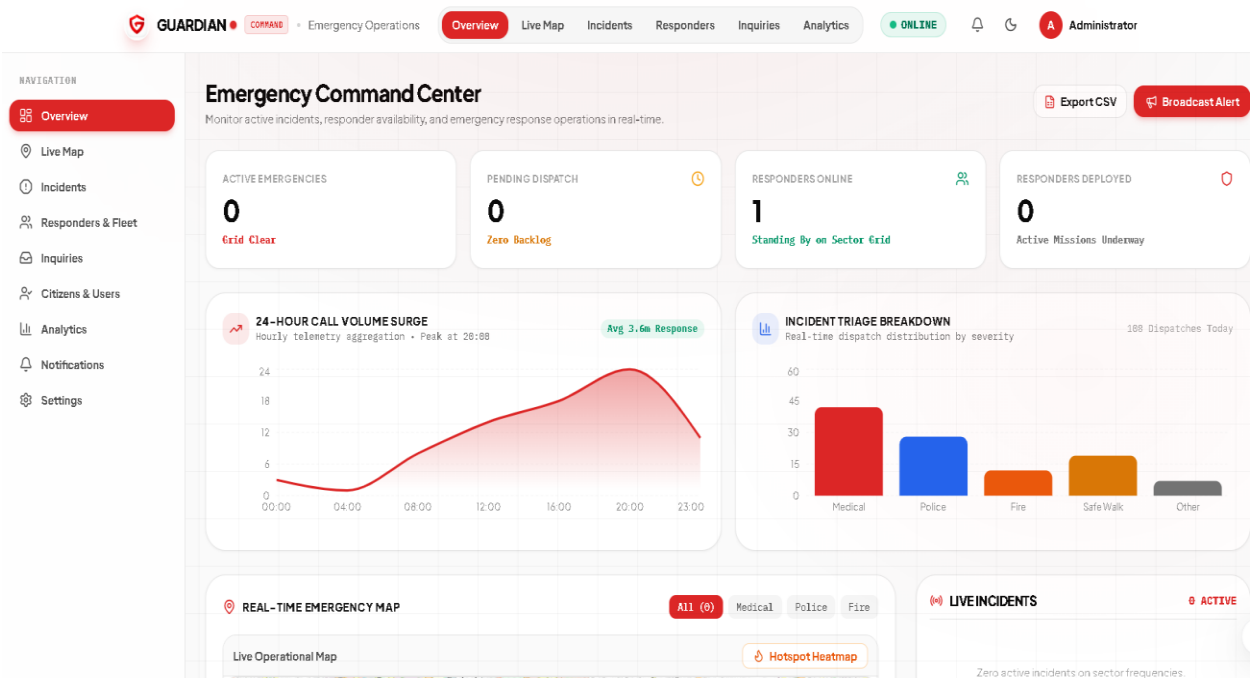


Figure 2 : Guardian Emergency Admin page

VIII. LIMITATIONS

Like any software system, Guardian also has certain limitations.

Internet Connectivity

Real-time communication generally requires an active internet connection. Poor network connectivity can affect notifications and location updates.

GPS Accuracy

GPS may not always provide an exact location, especially inside buildings, underground locations, or areas with poor satellite visibility.

Smartphone Dependency

The system depends on the user's device, battery, GPS, and network availability.

False Emergency Requests

Users may accidentally activate SOS or misuse the system. Such requests can unnecessarily occupy responder resources.

Responder Availability

The system can identify available responders, but it cannot guarantee that a suitable responder will always be available nearby.

External Services

Some functions such as maps, authentication, notifications, and cloud services may depend on third-party platforms.

Security and Privacy

The system handles personal information and location data. Therefore, proper authentication, authorization, encryption, and data protection are necessary.

IX. CHALLENGES

Developing a real-time emergency system involves several technical challenges.

Real-Time Notification

Emergency notifications need to reach responders quickly and reliably.

Multiple Emergencies

The system should be capable of handling several emergency requests at the same time.

Responder Selection
Choosing an appropriate responder can be challenging because the system needs to consider location, availability, emergency type, and current workload.

Location Accuracy
Incorrect or outdated location information can affect the response process.

Data Security
Emergency information can contain sensitive personal and location details, so strong security measures are required.

System Scalability
As the number of users and emergencies increases, the backend should continue to operate reliably.

Network Problems
A weak or unavailable network can interrupt real-time communication. Future versions can therefore include alternative communication methods.

X. EXPECTED OUTCOME

The main expected outcome of Guardian is a structured platform that simplifies the emergency-response process.
The proposed system is expected to:

- Provide a simple way to report emergencies.
- Automatically capture location information.
- Share important emergency details with responders.
- Provide real-time emergency notifications.
- Help responders manage assigned emergencies.
- Allow administrators to monitor active incidents.
- Maintain emergency history.
- Improve coordination between different users.
- Provide a foundation for future AI-based emergency management.

The actual performance of the system should be evaluated through testing. Important measurements can include notification delivery time, responder assignment time, location accuracy, system availability, and user satisfaction.

XI. FUTURE SCOPE

Guardian can be improved significantly in future versions.
Artificial Intelligence

AI can be used to understand emergency descriptions and automatically classify the type and priority of an emergency.

Intelligent Responder Assignment
The system can consider distance, traffic, responder availability, emergency type, and workload when selecting a responder.

Automatic Accident Detection
Smartphone sensors, vehicle sensors, or wearable devices could be used to identify possible accidents and automatically initiate an emergency request.

Smartwatch Integration
Future versions could connect with smartwatches and other wearable devices for features such as fall detection, health monitoring, and emergency alerts.

Offline Emergency Support
SMS, mesh communication, or other technologies could be explored for situations where internet connectivity is unavailable.

Voice-Based Emergency Reporting
Users could report an emergency using voice commands instead of manually entering information. Natural-language processing could then be used to understand the emergency.

Predictive Analysis
Historical emergency data could be analysed to identify areas where emergencies occur more frequently and understand patterns over time.

Multi-Agency Integration
In a future production environment, Guardian could potentially integrate with authorized police, ambulance, fire, hospital, and disaster-management systems.

XII. CONCLUSION

Guardian – Smart Emergency Intelligence System is proposed as a centralized platform for improving the way emergency information is reported, communicated, and monitored.
The system connects citizens, responders, and administrators through a common platform. A citizen can report an emergency and share their location, while responders can receive the request, access the required information, navigate to the incident location, and update the emergency status. Administrators can monitor the overall situation through a centralized dashboard.

The main idea behind Guardian is to make emergency communication more organized and reduce unnecessary steps between reporting an incident and responding to it.

The current system provides a foundation that can be expanded with technologies such as artificial intelligence, machine learning, wearable devices, automatic accident detection, predictive analytics, and offline communication.

However, the effectiveness of the system should be established through real-world testing rather than assumptions. Future evaluation can focus on measurable factors such as notification speed, location accuracy, responder assignment time, reliability, scalability, and usability.

Overall, Guardian represents a practical approach to developing a technology-supported emergency-response environment and provides a foundation for further research and development in intelligent emergency management.

REFERENCES

[1]. J. L. Gabella, I. A. P. Gualda, I. L. Altoé, et al., "Designing, development and validation of an app to reduce the response time of the emergency medical services," PLOS ONE, 2024.

[2]. A. S. M. Mohsin and M. A. Muyeed, "IoT based smart emergency response system (SERS) for monitoring vehicle, home and health status," Discover Internet of Things, vol. 4, 2024.

[3]. D. J. S. Sako Rivers, C. G. Igiri, E. O. Bennett, and F. B. Deedam, "ESARS: A Situation-Aware Multi-Agent System for Real-Time Emergency Response Management," European Journal of Information Technologies and Computer Science, 2024.

[4]. H. Zhang, R. Zhang, and J. Sun, "Developing real-time IoT-based public safety alert and emergency response systems," Scientific Reports, 2025.

[5]. "Automatic priority analysis of emergency response systems using Internet of Things (IoT) and machine learning (ML)," Transportation Engineering, vol. 19, 2025.

[6]. S. Mohammadi, A. Olivier, and A. Smyth, "Dynamic penalty-based dispatching decision-making for improved emergency response in urban environments: A heuristic approach," Frontiers in Future Transportation, 2025.

[7]. "AI for Disaster Response: Developing Intelligent Systems for Real-Time Emergency Management," IEEE Conference Publication, 2025.

[8]. "AI-Enhanced Emergency Call Center Assistant and Situational Awareness for Optimized Emergency Response: Using Speech Recognition, NLP Mapping, NER," IEEE, 2025.

[9]. "Crowdsourced geolocation: Detailed exploration of mathematical and computational modeling approaches," Cognitive Systems Research, 2024.