Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

Real-Time Disaster Assistance System

Thamizhchelvan M.[1], Sakthibala S.[2], Sathyabama A.R.[3*]

[1,2]Student, [3]Assistant Professor, Department of Information Technology, Velammal Engineering College, Chennai, India.

E-mail: [1]mthamizh2005@gmail.com, [2]ssakthibala38@gmail.com, [3*]sathya.it1@gmail.com

Abstract

Natural disasters such as floods, earthquakes, landslides, cyclones, and forest fires cause considerable destruction of infrastructural and communication facilities globally. Such disruptions hinder timely rescue missions and endanger human lives. To solve such challenges, the Real-Time Disaster Assistance System (RDAS) is proposed. The platform is a full-stack web application developed using MongoDB, Express.js, React, and Node.js (MERN) technologies and supplemented by the following libraries and frameworks: Socket.IO, Leaflet.js, Tailwind CSS, and Vite-PWA for PWA (Progressive Web Application) development. The platform provides an array of functionalities, which include: a one-click Save Our Souls (SOS) alert feature with automatic capturing of global positioning system (GPS) coordinates, real-time volunteer notifications, interactive maps of disaster-affected areas, curated disaster news feed, locations of relief centers, food and water management, and offline SOS queuing functionality. Three interfaces of the RDAS are provided based on the user roles – victims, volunteers, and administrators. Performance testing of the RDAS revealed its ability to send alerts within 500 milliseconds, ensure a 100% success rate for offline queues, and process over 600 alerts per minute simultaneously.

Keywords: Disaster Management, MERN Stack, Progressive Web Application (PWA), Real-Time Communication; SOS Alert System.

1. Introduction

Natural calamities are always a risk to human lives and property globally. The use of technology to minimize the effect of natural catastrophes has been widely studied by

*** Corresponding Author**

DOI: https://doi.org/10.36548/jucct.2026.2.004
Received: 27.02.2026, received in revised form: 01.04.2026, accepted: 16.04.2026, published: 28.04.2026

researchers. Although significant progress has been made towards improving geospatial technology and collaborative technologies, there is still the challenge of achieving rapid and effective communication between the affected parties and the relevant authorities. In case of a natural calamity, the most common communication means are usually the first to be overwhelmed with many people trying to reach out for help. This includes mobile network communication, landline communication, and the internet. In the process, the victim may not be able to share his position with the emergency team. It also becomes difficult to have a real-time picture of areas where assistance is needed most. Most emergency management tools that exist today cannot offer real-time situational awareness, process numerous requests simultaneously, coordinate volunteer emergency responders and governmental agencies, etc. Automated chatbots, assistants powered by artificial intelligence, crowdsourced maps using satellites have made a little contribution individually but no tool was ever able to address the whole issue. This study presents the design of the Real-Time Disaster Assistance System (RDAS) a web-based application.

The design of RDAS is guided by the decision to create a Progressive Web Application (PWA) as opposed to developing a native mobile application or an iOS emergency app specifically designed for iPhones. The decision to adopt PWA is based on its distinct advantages in the context of disaster management. Firstly, PWAs do not require users to download applications from app stores; hence, users from different platforms such as Android phones, iPhones, computers, and cheap feature phones will have equal opportunities to connect to RDAS using their web browsers. For instance, native applications are usually coded differently for different platforms, and each application has to go through unique approval procedures before being uploaded onto different app stores. Secondly, PWAs allow the development of offline applications using service workers and IndexedDB technologies. Therefore, RDAS will be able to cache its application shell and queue its SOS alert functionalities without requiring internet connectivity since disaster-stricken regions often lack connectivity due to the damage to internet infrastructure.

Finally, PWAs allow for installation straight from the web browser to the user's home screen without any app store verification, which decreases the interval between the user’s first contact with the system to when they can trigger an alert. The fourth factor is that PWAs are much easier to support and cost less than native apps: one single codebase runs on all platforms, meaning there’s no need for separate iOS and Android developers. Any changes

made on the server are reflected instantly for all users regardless of the app store updates. Finally, in disaster relief contexts where budget allocations for IT are low, having a cheaper way to develop and maintain the system through the use of a PWA rather than native application is simply practical. Therefore, the decision to develop the web app using the Progressive Web App architecture was made based on the advantages mentioned above.

The specific objectives of the RDAS are as follows:

- To provide a one-click SOS alert system with automatic GPS capture
- To enable real-time communication between stakeholders
- To support interactive disaster mapping and relief coordination
- To ensure offline alert queuing and synchronization
- To deliver a scalable and device-independent solution.

2. Literature Survey

A comprehensive review was conducted across disaster management systems, emergency communication methods, geospatial platforms, Progressive Web Application frameworks, and real-time communication technologies. Table 1 summarizes the key works examined, highlighting their technology choices, contributions and relevance to the RDAS design.

Table 1. Systematic Review of Related Works in Disaster Management

Reference	Primary Focus Area	Methodology / Technology	Key Contribution	Application Domain
Iovino et al. (2020)	Emergency Alert Management	Integrated emergency alert platform design	Holistic framework for emergency alert coordination	Extreme natural events
Yang et al. (2021)	Geographic Information Systems	Crowdsourcing, VGI, data integration	Collaborative geographic information service platform	Disaster emergency management

Basak et al. (2020)	Coordinated Disaster Management	Social media crowdsourcing	Framework for community-driven disaster resilience	Multi-hazard disaster scenarios
U & Mahalakshmi (2023)	Disaster Relief Management	Multi-source data integration	Integrated disaster relief support system	General disaster response
Fouzan et al. (2024)	Disaster Management Systems	Mobile app (React Native), Node.js, AI	Unified mobile platform with decision support	Emergency and disaster handling
Rushitha et al. (2025)	AI-Based Disaster Assistance	AI chatbot, Python-based system	Intelligent assistant for rapid disaster response	Emergency alert systems
Gaire et al. (2020)	IoT-Based Disaster Management	IoT, cloud computing, big data	Scalable situation-aware disaster management system	Large-scale disaster environments
Guntha et al. (2020)	Crowdsourced Relief Systems	Crowdsourcing platforms	Practical insights from real flood relief deployment	Flood disaster management
Idris & Ishak (2020)	Relief Coordination	Geo-crowdsourcing, mobile geolocation	Two-way communication framework for relief distribution	Flood relief coordination
Damaševičius et al. (2023)	Emergency Response Systems	IoT, AI, cloud, blockchain	Internet of Emergency Services (IoES) framework	Smart emergency systems
Ssin et al. (2022)	Geographic Emergency Management	Digital twin, XR, ICT	Conversational digital twin for real-time communication	Urban disaster scenarios
Zhang et al. (2022)	Disaster Risk Control	GIS, databases, intelligent systems	Regional disaster risk and rescue platform	Multi-disaster environments

Varun Kumar et al. (2025)	Web-Based Emergency Systems	PHP, MySQL, geolocation	Web-based emergency response platform	Crisis response systems
Le et al. (2023)	Crisis Contextualization	Collaborative and social technologies	Framework for contextual crisis understanding	General crisis management
Hutagalung & Indrajat (2023)	Public Participation Models	Crowdsourcing approaches	Public participation-based disaster relief model	Disaster response planning

However, there are numerous deficiencies associated with current disaster management approaches and technologies. First of all, a majority of existing systems are focused solely on individual tasks such as alerts dissemination, geospatial information gathering and distribution, and resource coordination rather than integrating all aspects of disaster management into a single platform. Second, many disaster response systems rely on the presence of a reliable Internet connection in order to function properly. This requirement is impractical since Internet infrastructure may be damaged along with other facilities in areas affected by disasters. Third, lack of adequate communication facilities reduces their efficacy as well.

In order to address aforementioned shortcomings, the Real-Time Disaster Assistance System is proposed. It features an integrated approach that incorporates various components including real-time communication facilities, geospatial data visualization tools, and resource management system. The PWA concept makes RDAS accessible on any device while allowing offline operation. Thus, the system is able to function under conditions where stable Internet connections are absent. Besides, the inclusion of real-time communication facilities enables quick alerts delivery and improves coordination between users.

3. Proposed Methodology

The RDAS is conceptualized as an integrated and real-time system that would help solve the critical problems of disaster communication and coordination. In the design of this platform, real-time alert propagation, geospatial intelligence, and role-based interactions will be considered as key factors. This system follows the Progressive Web App (PWA) paradigm to be accessible across a wide variety of devices without having the need for application

installation. The system is equipped with offline capabilities enabled by service workers and local storage facilities to provide continuity of operations when infrastructure degradation becomes a concern.

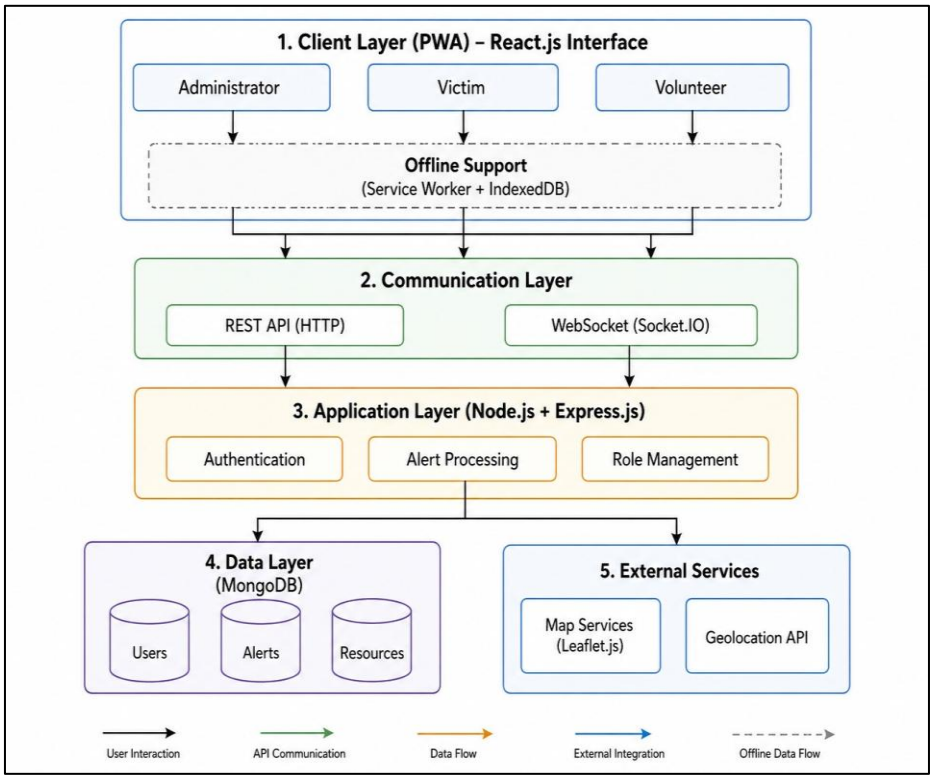

Figure 1. RDAS System Architecture

The architecture of the RDAS (Figure 1) involves several layers that allow seamless data flow and ensure separation of concerns. The implementation of this platform is carried out through the use of the MERN stack (MongoDB, Express.js, React.js, and Node.js). The frontend layer of RDAS uses React.js for building a responsive and intuitive interface, customized to the needs of various roles (victim, volunteer, administrator, etc.). The backend layer is responsible for managing the system API, authentication, and other logic aspects of RDAS, which is accomplished through Node.js and Express.js.

An important feature of the proposed system is the real-time communication protocol, which is based on WebSocket technology. The main purpose of this element of the system is providing immediate transmission of messages about emergencies sent by users. Moreover, the system implements geolocation services, which automatically capture the geographical position of a particular person and then, using GIS technologies, helps identify any available volunteers and aid stations close to the location of an emergency. This approach contributes

greatly to increasing situational awareness. In order to overcome problems with network connectivity, it was decided to implement the offline alert handling functionality within the proposed solution. Namely, when network connectivity is lost, all the alerts sent by a particular person will be stored in the client-side storage, which will ensure their automatic synchronization after connection recovery. Additionally, there is a role-based approach to user interaction, according to which a person can either be classified as a victim, volunteer, or administrator.

In addition to that, the design includes map-based visualizations for showing disaster related data like SOS messages, rescue centers, etc. The graphical presentation helps the users in knowing the locations of incidents and available facilities. Moreover, the system has security features to ensure that only authorized personnel can use the system and perform data manipulation tasks.

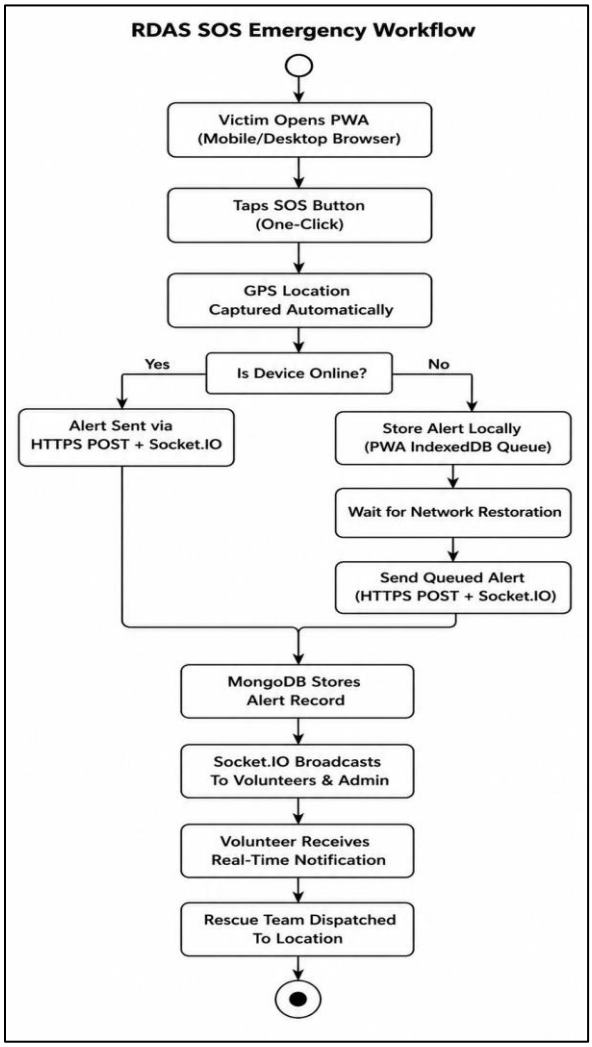

Figure 2. SOS Emergency Alert Workflow

The operations process of RDAS, as shown in Figure 2, is a systematic one and follows a clear-cut end-to-end approach. Firstly, users connect with the system either by accessing its website from their browser or through the PWA. Subsequently, the location of the users is obtained via geolocation services based on the device being used. Upon initiating an SOS alert or resource request in the platform, the system checks the state of the available network connections. If the connection is up, the alert is immediately sent to the server and relayed to users in real-time. Conversely, if there is no internet connection, the alert is temporarily saved until the network becomes available when it gets synced with the server without duplication. On receiving an alert, the server processes it and saves it to the database. Additionally, the server relays the information to the appropriate volunteers and/or administrators. In turn, the volunteers have the capacity to reply to the alert.

4. Results and Discussion

Performance and efficiency of the recommended Real-Time Disaster Assistance System (RDAS) have been analyzed on the basis of both the demonstration of its performance and quantification of the same. Role-based authentication module is shown by the means of Figures 3 and 4. It offers secure and role-based entry into the system to the system users. Login allows the user to enter into the system in a controlled manner, while the user can use the registration window to choose his or her role – citizen or volunteer.

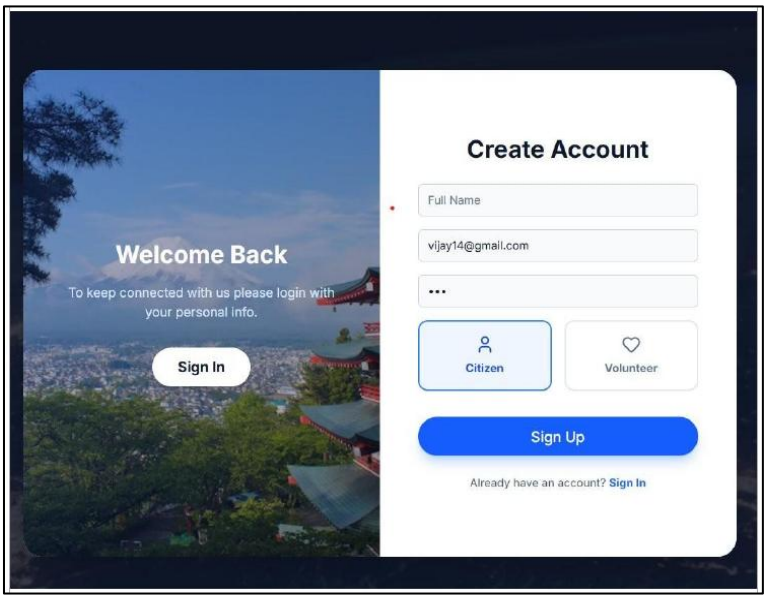

Figure 3. Login Interface

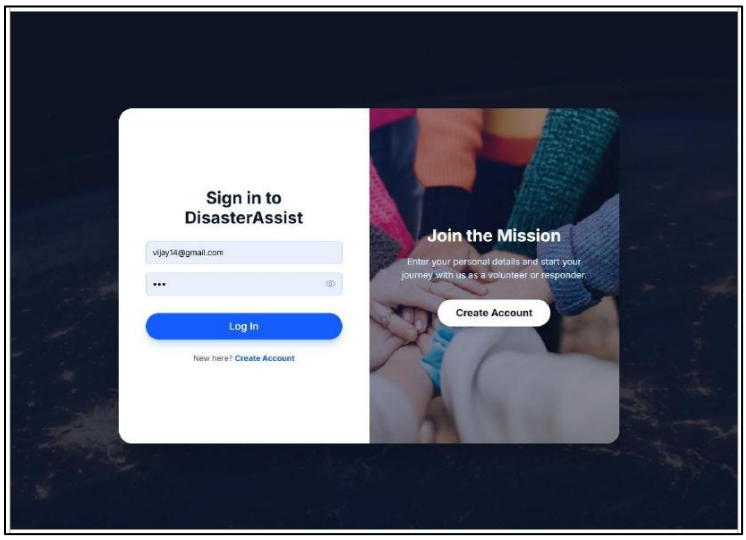

Figure 4. Create Account Interface

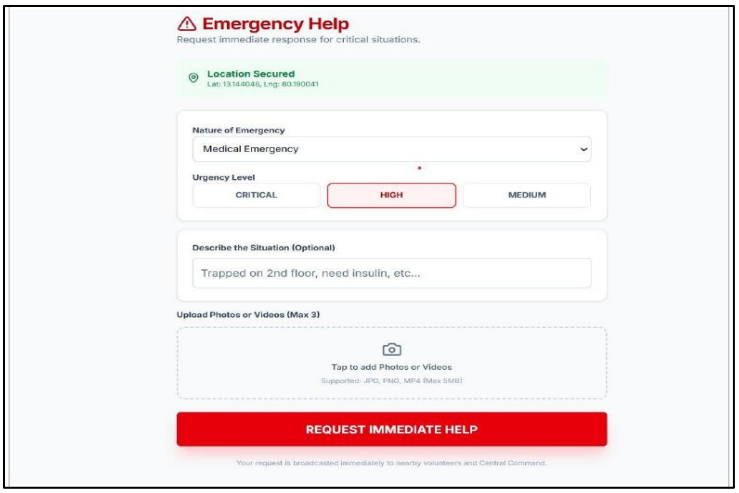

Figure 5. Emergency Help Module

The Emergency Help module is depicted in Figure 5, which shows how it works as the backbone of RDAS. The system allows users to trigger SOS alerts using one-click capability while automatically obtaining the GPS coordinates and specifying their type of emergency and urgency. Real-time communication capabilities have been incorporated to make sure that alerts are relayed to volunteers and administrators immediately.

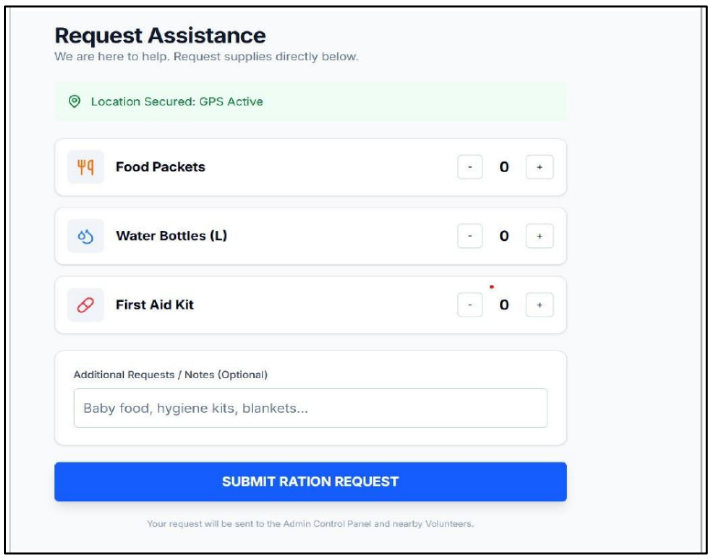

Figure 6. Request Assistance Module

The Request Assistance module is shown in figure 6. This allows victims to request necessary supplies like food, water, and medical aid. The Request Assistance module represents the functionality of the system in distributing resources during disasters. Figure 7 shows the Find Shelter module, where geospatial mapping is used to provide details of relief centers located nearby. With the help of the combination of a sortable list and the map, it becomes easier for users to choose shelters that can be accessed within a short time.

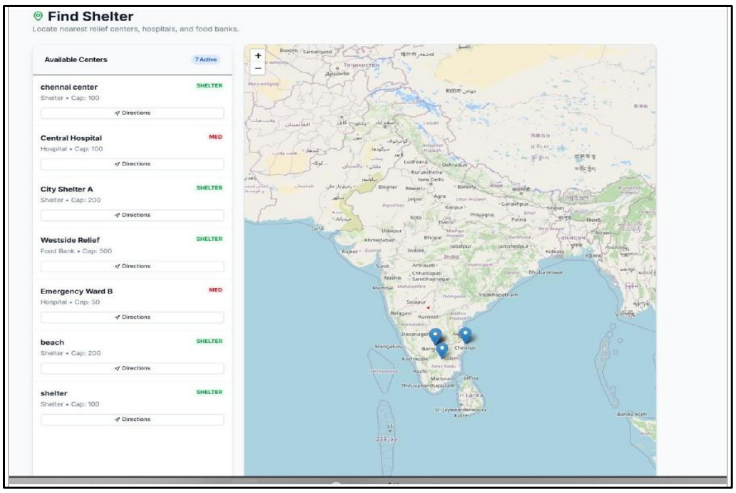

Figure 7. Find Shelter Module

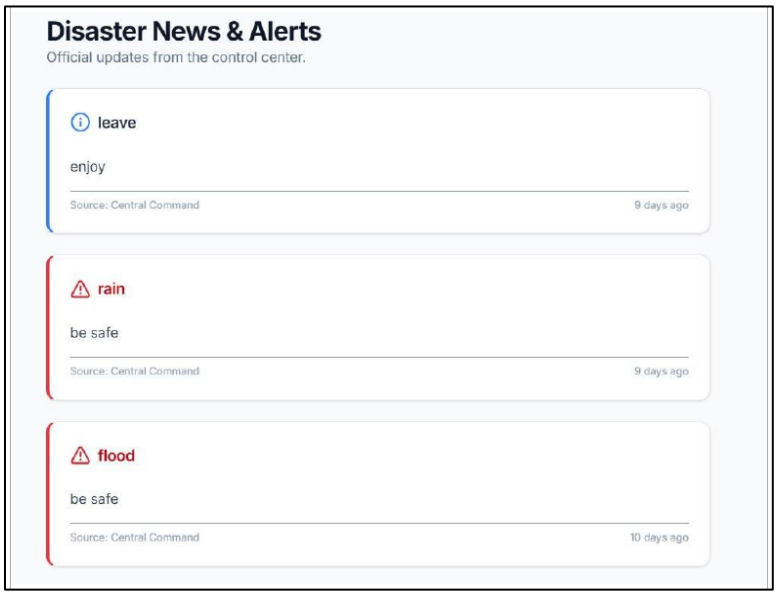

Figure 8. Disaster News and Alerts Module

Disaster News and Alerts is presented in Figure 8, and its aim is to provide users with timely notifications from authorities. By transmitting data directly through the system and avoiding third parties, the solution prevents the occurrence of miscommunication and guarantees reliability of messages. Public Relief Ledger is illustrated by Figure 9, where a blockchain-based transparency tool is proposed for monitoring relief operations. By keeping a permanent record of all transactions, the system increases credibility and makes activities transparent and traceable.

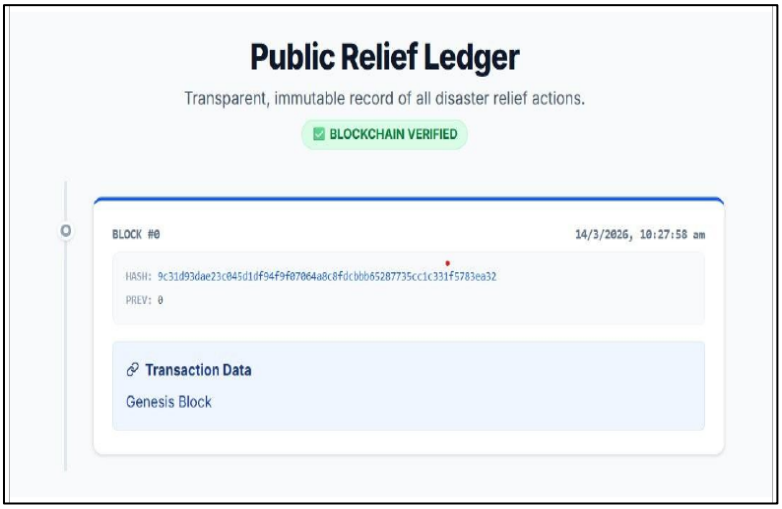

Figure 9. Public Relief Ledger

Table 2 shows the evaluation of the performance metrics in RDAS and proves high efficiency and capacity of the designed system. Latency in alert transmission does not exceed

120 ms on local networks and 480 ms on mobile networks. The Socket.IO framework enables the real-time communication between nodes and reduces broadcast latency to 80-250 ms.

Table 2. RDAS System Performance Evaluation Results

Metric	Result	Conditions / Notes
Alert Delivery — LAN	< 120 ms	500 test alerts; Node.js + MongoDB + Socket.IO
Alert Delivery — 4G Mobile	< 480 ms	200 test alerts on live 4G network
Socket.IO Broadcast Latency	80–250 ms	Server receipt to React state update on client
Geolocation Accuracy (GPS)	3–5 m	HTML5 Geolocation API with device GPS hardware
Geolocation (Wi-Fi/Network)	20–100 m	Urban Wi-Fi triangulation fallback
MongoDB Write Latency	< 20 ms	Indexed collection; Atlas shared cluster
Concurrent Alert Throughput	600+ alerts/min	Apache JMeter load test; Node.js cluster
Offline Queue Success Rate	100%	50 queued alerts across 10 offline sessions
Offline-to-Online Sync Time	< 3 s per alert	Measured on 4G reconnect
PWA Cold Launch Time	< 1.2 s	Service worker cache; Vite production build
Leaflet Map — 50 Markers	< 1.8 s	Mobile Chrome browser

Alert Transmission Success	99.4%	Including degraded-network scenarios
System Uptime (30 days)	99.7%	Node.js + MongoDB Atlas; Docker deployment
Usability Task Completion	94.3%	n=12 participants; timed task completion

The system proves to be highly scalable, as it can support over 600 simultaneous alerts per minute without any performance impact. All database transactions occur with minimal delays (<20 ms), guaranteeing smooth data processing even during peak loads. The use of Progressive Web Application technology allows for a cold start time of less than 1.2 seconds. One of the most significant advantages of the RDAS system is its ability to operate offline, allowing for a 100% offline storage and synchronization rate of alerts created when there is a lack of Internet access. On average, the synchronization process takes less than three seconds per message, confirming the effectiveness of the offline queueing strategy. Moreover, the accuracy of geolocation in GPS mode varies from 3-5 meters, thus allowing accurate detection of the affected people.

Moreover, the system presents impressive reliability features, including a 99.4% success rate in alert notifications and a 99.7% uptime rate within a 30-day evaluation period. Usability tests reveal that 94.3% of all tasks can be successfully accomplished by users, proving that the application is reliable and convenient under stressful conditions. When comparing the new solution to existing ones, like SMS or email messaging, the benefits become clear. The reduced lag time, real-time two-way communication, the ability to visualize geospatial information, and the offline operating mode represent significant improvements in coordination of victims, volunteers, and administrators. Nevertheless, a few restrictions can be recognized. For instance, there may be a lack of accuracy in the process of locating objects in indoor locations or crowded cities due to poor signal availability, whereas the effectiveness of the solution may depend on user devices. Concluding, it becomes evident that the new software proves effective in solving critical issues related to disasters. The real-time communication, geospatial intelligence, and offline operation present powerful features in enhancing disaster response effectiveness.

5. Conclusion

This research describes the design and development of Real-Time Disaster Assistance System (RDAS), which is a scalable application that intends to improve communication and coordination when dealing with disasters. RDAS is implemented using PWAs and the MERN stack technology, ensuring that the application is not dependent on specific devices and is also capable of offline use. Evaluations show that RDAS has good performance in terms of fast alert propagation, with less than 480 milliseconds of latency on mobile networks, more than 600 alerts per minute throughput capacity, and reliability with 99.4% success rates for alerts and 99.7% uptime. RDAS's offline queueing feature ensures consistency by providing 100% success rate in storing and syncing alerts during offline periods. Precise geolocation and usability are two other strengths of RDAS that have been confirmed to work in emergency situations. Despite some limitations in geolocation accuracy in indoor environments and the dependence on device's capacity, RDAS can greatly improve response times with real-time bidirectional communications and situational awareness. In the future, RDAS will evolve by improving the accuracy of positioning through AI prioritizations and integrating robust communication infrastructure.

References

[1] Iovino, Ludovico, Mattia d'EMIdIO, and Marco Modica. "Creating an Holistic Emergency Alert Management Platform." Journal of urban technology 27.2 (2020): 3-20.

[2] Yang, J., W. Huang, H. Zhang, and H. Li. "Research and Engineering Practice of Emergency Geographic Information Collaborative Service Mode with Crowdsourced Data." The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 43 (2021): 209-214.

[3] Basak, Jayanta, Parama Bhaumik, Siuli Roy, and Somprakash Bandyopadhyay. "A Crowdsourcing based Information System Framework for Coordinated Disaster Management and Building Community Resilience." In Proceedings of the 21st International Conference on Distributed Computing and Networking, 2020, 1-6.

[4] U, A., & Mahalakshmi, T. (2023). Hand for the World. International Journal of Advanced Research in Science Communication and Technology, 392–396

[5] Fouzan, M., Pavithran, P., A, V., & Fardeen, R. (2024). Disaster Guard for Disaster Management. International Journal for Multidisciplinary Research, 6(1).

[6] Manda Rushitha, Kakani Nikesh, Kudithi Sai Nitheesh, S Neduncheliyan “Rescue Connect: AI-Powered Disaster Relief System”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2025, 3828.

[7] Gaire, Raj, Chigulapalli Sriharsha, Deepak Puthal, Hendra Wijaya, Jongkil Kim, Prateeksha Keshari, Rajiv Ranjan et al. "Internet of Things (IoT) and Cloud Computing Enabled Disaster Management." In Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things, 273-298.

[8] Guntha, Ramesh, Sethuraman N. Rao, and Avinash Shivdas. "Lessons Learned from Deploying Crowdsourced Technology for Disaster Relief During Kerala Floods." Procedia Computer Science 171 (2020): 2410-2419.

[9] Idris, N. H., and M. H. I. Ishak. "A 2-Way Geo-Crowdsourcing Approach in Flood Relief Coordination and Distribution." In IOP Conference Series: Earth and Environmental Science, 2020, vol. 540, no. 1, 012075.

[10] Damaševičius, Robertas, Nebojsa Bacanin, and Sanjay Misra. "From Sensors to Safety: Internet of Emergency Services (IoES) for Emergency Response and Disaster Management." Journal of sensor and actuator networks 12, no. 3 (2023): 41.

[11] Ssin, Seungyoub, Junseong Bang, and Woontack Woo. "CDT-GEM: Conversational Digital Twin for Geographic Emergency Management." In International XR Conference, 2022, 134-138.

[12] ZHANG, Lei, W. U. Binzhuo, and T. E. N. G. Zhoubin. "Construction and Application of Natural Disaster Risk Control and Emergency Rescue Management Platform: A Case Study in Zhejiang Province." The Chinese Journal of Geological Hazard and Control 33, no. 4 (2022): 134-142.

[13] B. Varun Kumar, P. Durgavihashini, C. Kaleeswari, B. Subanu, “Emergency Management System: A Web-Based Platform for Efficient Crisis Response”, Journal for Research in Applied Science and Engineering Technology (IJRASET), 2025, 4830.

[14] Le, Ngoc Luyen, Jinfeng Zhong, Elsa Negre, and Marie-Hélène Abel. "CORec-Cri: How Collaborative and Social Technologies Can Help to Contextualize Crises?." 2023, arXiv preprint arXiv:2310.02143.

[15] Hutagalung, Simon, and Himawan Indrajat. "Developing a Crowdsourcing-Based Disaster Relief Model Based on Public Participation." International Journal of Safety and Security Engineering 13, no. 1 (2023): 115-120.