Secure Cross-Layer Deployment for Real-Time Disaster Reporting and Visualisation Using Mobile Applications

Rashid Mustafa , Jun Han , Nurul I. Sarkar * , Krassie Petrova

Posted Date: 3 October 2025

doi: 10.20944/preprints202510.0225.v1

Keywords: disaster management; cross-layer architecture; mobile applications; user-event reporting; hazard visualization; security implementation

 doi:10.20944/preprints202510.0225.v1

Article

Secure Cross-Layer Deployment for Real-Time Disaster Reporting and Visualisation Using Mobile Applications

Rashid Mustafa [1], Jun Han [1], Nurul I. Sarkar [2,*], and Krassie Petrova [2]

[1] School of Information Technology, Whitecliffe College of Arts and Science, Auckland 1010, New Zealand
[2] Department of Computer and Information Sciences, Auckland University of Technology, Auckland 1010, New Zealand
* Correspondence: nurul.sarkar@aut.ac.nz

Abstract

As the number of natural and man-made catastrophes has increased in recent years, there has been an increasing need for disaster response that is quicker and more efficient. Information from traditional sources, such as radio, television, and websites, is sometimes incomplete or delayed. While mobile applications provide a means of enhancing real-time crisis communication, a secure mobile app-based solution has not been fully explored yet. In this paper, we propose a secure and scalable cross-layer disaster management system architecture. To validate the system performance, we developed a user-centred, scalable mobile application known as disaster emergency events application (DEAPP) for real-time disaster reporting and visualization including, disaster notifications and observing the affected areas on an interactive map. The solution connects a web-based backend, cloud database, and native Android mobile app via a cross-layer architecture. Role-based access control, HTTPS connection, and verified event publication all contribute to security. Moreover, Redis caching is employed to expedite data access in emergency situations. Verifying publicly filed reports to prevent false alarms, safeguarding real-time data transfer without slowing down the system, and creating an intuitive user interface for individuals in high-stress circumstances are some of the issues that the project attempts to solve. Results obtained show that a mobile system that is secure, scalable, and easy to use can enhance catastrophe awareness and facilitate quicker emergency responses. For developers, researchers, and emergency organisations looking to leverage mobile technology for disaster preparedness, the findings provide helpful insights.

Keywords: disaster management; cross-layer architecture; mobile applications; user-event reporting; hazard visualization; security implementation

1. Introduction

The frequency, scale, and complexity of disruptive events that endanger lives, infrastructure, and economic activity have intensified, spanning floods, wildfires, earthquakes, industrial accidents, and public-health emergencies. At the same time, mobile technologies and data-driven platforms have become central to preparedness, warning, and coordination, yet many deployed solutions still rely on one-way alerting models with limited interactivity, delayed situational updates, or static user settings. Systematic reviews and domain applications point to both the promise and the persistent gaps: crowdsourced hazard apps emphasise notifications and engagement but vary in verification and usability [1]; evacuation support tools demonstrate the value of real-time routing but depend on model accuracy and timely delivery [2]; IoT/edge/cloud pipelines achieve low-latency sensing and scalable dissemination [3]; and trustworthy-AI studies stress explainability, bias control, and data-fusion challenges in multi-hazard settings [4]. Broader work on digital transformation in disaster management similarly highlights fragmented implementations and the need for integrated, interoperable architectures [5].

 doi:10.20944/preprints202510.0225.v1

Security and resilience remain cross-cutting concerns. Cross-layer defences at physical/MAC/network levels can improve delivery ratios and energy efficiency under adversarial conditions [6], while SDN/NFV approaches help isolate threats and preserve quality of service across slices in next-generation networks [7]. From a user-centric standpoint, purpose-built platforms for medically vulnerable populations show how structured data flows and administrative workflows can strengthen preparedness [8]; emergency medical coordination systems demonstrate secure data exchange and location-aware operations [9]; and web-based 3D GIS underscores the value of interactive visualisation and integrated decision support [10]. Community-facing mobile/web solutions illustrate how shared location, alert orchestration, and resource tracking can elevate situational awareness [11]. Complementary paradigms—crowdsensed early-warning via smartphone sensors [12], handset-derived advanced mobile location to PSAPs [13], and bystander-activation for CPR with AED navigation [14]—further motivate designs that couple reliability with human-centred workflows. A second stream of digital-transformation literature converges on the same conclusion: piecemeal tools should evolve toward coherent, scalable, and secure ecosystems [5].

Emerging architectures for post-disaster connectivity and data integrity—e.g., blockchain-enabled UAV coordination and tamper-resistant exchanges [15,16], offline/multilingual voice networks backed by decentralised ledgers [17], and cross-layer resilience assessment across application/infrastructure [18]—reinforce two design imperatives for public-facing systems: (i) robust end-to-end security with auditable provenance, and (ii) low-friction interfaces that minimise cognitive load in high-stress contexts. In parallel, humanitarian open-source risk-monitoring platforms [19] and global innovation mappings [20] document wide adoption of AR/VR, IoT, UAVs, and mobile apps in preparedness and response. Mobile real-time data-aggregation pipelines show that near-instant reporting and scalable ingestion are feasible in practice [21], while corporate-level assessments of mobile emergency apps expose recurrent weaknesses in encryption, API hardening, and data handling—underscoring the necessity of rigorous security baselines [22]. Visualization research for emergency training also emphasises clarity, cognitive economy, and integration constraints that are directly relevant to on-device hazard mapping and alert comprehension [23].

Taken together, the literature indicates a clear opportunity: to pair citizen-driven, two-way reporting with verified publication and intuitive visualisation, inside a secure cross-layer architecture that sustains performance under load. This paper pursues that opportunity by designing and evaluating a mobile system that supports dynamic incident submission, administrative verification, and real-time map-based visualisation—while enforcing security and usability constraints drawn from the foregoing evidence.

1.1. Research Challenges

There are a number of important research problems in creating a secure cross-layer mobile system for real-time disaster reporting and visualisation. The design, deployment, and operating stages of the mobile application prototype covered in this paper all present these difficulties. The three research challenges are highlighted below.

1. **What secure cross-layer architecture can support real-time disaster reporting and danger zone visualisation without compromising system performance?** One of the most difficult tasks is maintaining system responsiveness while safeguarding communication and data flow between the mobile application, backend server, and database. A MySQL database and a Spring Boot backend are combined with Android in the proposed design in this paper. Only verified administrators can validate and publish events thanks to role-based access, and RESTful APIs via HTTPS ensure safe connections. By improving performance under load, Redis caching achieves a balance between speed and security.
2. **What mechanisms can validate public disaster reports while minimising the risk of false data during emergencies?** The possibility of malicious or erroneous data arises when incidents are left up to public users to report. The developed solution makes use of an admin-controlled

 doi:10.20944/preprints202510.0225.v1

verification system, meaning that all user-submitted events are flagged as "pending" and need to be manually reviewed before being shared with other users. As stated in this paper, this method increases data trust but also introduces future issues, such as scaling verification in high-volume scenarios. These issues could be resolved by using blockchain-backed data provenance or AI-based validation.

3. **What user-interface and system-design principles ensure accessibility, ease of use, and effective information delivery under high-stress conditions?** In times of crisis, usability is critical. There should be little cognitive strain and effort required for users to report incidents and receive warnings. Automatic location pinning, simplified input forms that simply need the event name and time, and an integrated news section with real-time Civil Defence updates are some of the ways the mobile application tackles this issue. With simplicity, usability, and durability as top priorities, these design decisions are primarily focused on effectiveness in high-stress, low-connectivity environments. These difficulties collectively highlight how crucial it is to develop disaster management systems that are not just technically sound and scalable, but also firmly focused on user demands. To guarantee that the public is actively involved in disaster preparedness and response, these problems must be resolved in order to provide trustworthy mobile-based emergency solutions.

1.2. Research Questions

A practical mobile system for real-time disaster reporting and visualisation must (i) safeguard confidentiality, integrity, and availability across the client–server–database path; (ii) validate publicly submitted reports to limit misinformation without sacrificing timeliness; and (iii) reduce cognitive effort so non-expert users can act quickly under stress [1–23]. In this study, we address the following three research questions:

- **RQ1. What secure cross-layer architecture can support real-time disaster reporting and danger-zone visualisation without compromising system performance?**
  The significance of this research question lies in the fact that disaster environments often involve simultaneous high-volume data transmission and exposure to malicious attacks [6,7]. If the system architecture does not strike a balance between security and performance, delays, data loss, or even complete service failure may occur—directly undermining emergency response and decision-making. Investigating a secure yet efficient cross-layer architecture is therefore critical to ensuring that disaster reporting and danger-zone visualisation remain reliable in real-world scenarios.
- **RQ2. What mechanisms can validate public disaster reports while minimising the risk of false or misleading data during emergencies?**
  The importance of this research question stems from the fact that the public is often the first to submit disaster information, but such reports may contain errors, duplication, or intentional misinformation [1,12,13]. Without effective validation mechanisms, inaccurate data could distort emergency assessments, misallocate resources, and even worsen the impact of the disaster. By exploring approaches such as multi-source cross-verification, role-based confirmation, or automated detection methods, this study seeks to ensure that information remains both timely and credible. Addressing this challenge enhances the trustworthiness and practical value of the disaster recovery system.
- **RQ3. What user-interface and system-design principles ensure accessibility, ease of use, and effective information delivery under high-stress conditions?** This research question is significant because users in disaster situations are often under extreme stress, and complex or unintuitive system design can hinder their ability to report or interpret information quickly [10,11,23]. If usability is not prioritised, the system risks being ineffective at the very moment it is most needed. By examining user-interface and system-design principles that reduce cognitive load and enhance clarity, this study ensures that even non-expert users can operate the system effectively during

 doi:10.20944/preprints202510.0225.v1

emergencies. Ultimately, this focus on accessibility and usability determines the real-world value and impact of the proposed disaster recovery solution.

1.3. Study Contribution

We developed and evaluated the *Disasters Emergency Events Application* (DEAPP), a mobile system that: (i) accepts dynamic, GPS-assisted public incident reports, (ii) applies an administrative verification workflow prior to publication, and (iii) renders affected zones on an interactive map in real time. The implementation adopts a layered design with a native Android client, a Spring Boot administrative backend, and a MySQL datastore, communicating via HTTPS; performance under contention is sustained with in-memory caching for frequently accessed artefacts (e.g., event summaries and map layers). The design choices are informed by the reviewed evidence on crowdsourcing and engagement [1,11], low-latency sensing and dissemination [3,21], secure and resilient networking [6,7,22], and effective visualisation for rapid comprehension [10,23]. The resulting contribution is a secure, performance-aware, and usability-focused framework for citizen-driven disaster reporting and visualisation that can strengthen community resilience and accelerate reliable information flows during crises.

The main contributions of this paper are summarized as follows:

- **End-to-end secure cross-layer architecture for real-time disaster intelligence:** We develop and validate a layered design (Android ↔ Spring Boot ↔ MySQL) with role-based access, HTTPS APIs, and in-memory caching to sustain responsiveness under surge while preserving confidentiality, integrity, and availability.
- **Verified two-way reporting pipeline that limits misinformation:** To this end, we develop a mobile application called Disaster Emergency Events Application (DEAPP) for real-time disaster reporting and visualisation. The civil defence news integration and official updates are also included in the DEAPP. We introduce a governance workflow where public submissions default to *pending* and are released only after administrative verification; this process supports rapid broadcast once approved and is designed for extensibility (e.g., automated or provenance-aided checks).
- **Usability-first interaction model for high-stress contexts:** We develop mobile user interface to minimise cognitive load via automatic location pinning, a short event form (mandatory name and timestamp), and an integrated civil-defence news feed, enabling non-experts to act quickly during emergencies.
- **Generalizable blueprint for secure, performance-aware, citizen-driven visualisation.** To this end, we develop secure packet-based communication between mobile client and HTTPS Server. By integrating verified crowdsourced reports with responsive map layers and hardened networking practices, the work contributes a reusable framework for mobile disaster systems that strengthens community resilience and accelerates trustworthy information flows.

2. Related Work

Evaluations of 77 apps from the Google Play Store highlight the value of combining user-driven features, AI, IoT, and crowdsourcing. Applications for managing disasters on mobile devices, especially those pertaining to flood preparedness, are essential for communication, alerting, and coordinating operations. Their ability to improve community resilience over the long term hinges on gamified engagement, inter-agency cooperation, timely updates, and interactive communication skills [1]. Mobile applications like EscapeWildFire demonstrate strong potential for real-time evacuation support during wildfires, validated through case studies in Cyprus and a historic Texas fire. Their effectiveness, however, hinges on precise wildfire modelling, rapid notification, and continuous system improvements, with open-source availability encouraging global adoption by fire authorities [2]. To enable secure emergency response, Zhang et al. introduce and evaluate an Internet of Things (IoT)-based system that integrates cloud platforms, edge nodes, and heterogeneous sensors., low-latency notifications in a

 doi:10.20944/preprints202510.0225.v1

variety of situations, such as medical emergencies, gas leaks, fires, and accidents [3]. Its exceptional accuracy, dependability, and scalability are highlighted by comparison with cutting-edge solutions, highlighting its potential for implementation in smart cities, industries, and infrastructure with upcoming AI-driven improvements. Trustworthy AI is increasingly recognised as an important tool in natural disaster management. Methods such as explainable AI (XAI), machine learning, deep learning, data fusion, and multi-criteria decision-making can support predictive models, early warning systems, and resource allocation. While highlighting its potential and difficulties—such as explainability, bias, ethics, and data limitations—a systematic evaluation of 108 studies also provides important insights and answers to help shape future disaster resilience plans [4]. A theoretical review methodology is used by Fischer et al. [5] to describe the state of digital transformation (DT) in disaster management (DM), making a distinction between organizational initiatives that are facilitated by IT and more general digital initiatives. It describes future research directions in DM, highlights differences from industrial DT, and unifies disparate studies by offering an integrative framework.

This study integrates the physical, MAC, and network layers in a cross-layer security system designed to detect and isolate wormhole and blackhole attacks in wireless ad hoc networks. The system uses an Enhanced Support Vector Machine (E-SVM) with NS3 simulation and shows improvements in energy efficiency, packet delivery ratio, and QoS, while remaining protocol-independent and reducing false positives [6]. Using network function virtualisation (NFV) and software-defined networking (SDN), Allaw et al. [7] offer a cross-layer security (CLS) framework for 5G/6G network slices that guarantees adaptive threat detection, isolation, and resilience. By improving scalability, QoS, and slice integrity through their hybrid SDN/NFV strategy, CLS is positioned as a crucial facilitator for next-generation secure mobile networks. The K-DiPS system, which consists of the web platform K-DiPS Online and the smartphone app K-DiPS Solo, is presented by Nakai et al.. It allows medically vulnerable people (MVPs) to enter personal and health information, which is then sent to local governments via the cloud for disaster preparedness and response coordination. Actionable catastrophe training, resource allocation, simulation, and mapping of susceptible persons are all made possible by the system's smooth MVP-to-government data flows [8]. Perera et al. [9] presented a secure mobile and web-based emergency platform that improves coordination between ambulances and hospitals (EMS) in Sri Lanka by allowing simultaneous AES-256-encrypted health data transmission, real-time GPS tracking of ambulances, and OCR-enabled ID data acquisition. The technology greatly improves the effectiveness and preparedness of emergency response by simplifying interagency communication and protecting patient data. Based on Vue.js and Cesium Digital Earth, Yang et al. [10] create a full web-based interactive 3D GIS for emergency response that includes live spatial analysis, interactive route planning, augmented reality visualisation, landslide susceptibility modelling, and integrated message dispatch. In catastrophe situations, their system is a prime example of how WebGL-powered 3D geospatial tools may improve multi-source data integration, decision support systems (DSS), and situational awareness (SA). By providing real-time location sharing, alert coordination, and resource tracking to promote proactive disaster management, Vera et al. [11] present a mobile (and web) application that improves disaster preparedness and response. Through enhanced emergency response effectiveness, better situational awareness (SA), and smooth data interchange, the platform benefits communities and emergency agencies alike. A crowd-sourced smartphone-based Earthquake Early Warning System (EEWS) that uses accelerometer data to identify seismic events and promptly notify users who may be at risk is provided by the Finazzi et al. [12]. This technology provides essential pre-shaking warnings, improving emergency preparedness through scalable, economical deployment, particularly in areas without conventional seismometers as in Table 1. To improve the accuracy and timeliness of Emergency Services (EMS) responses, the Advanced Mobile Location (AML) system allows smartphones to automatically send handset-derived GNSS/Wi-Fi location data via SMS or HTTPS to Public Safety Answering Points (PSAPs). By enhancing geolocation accuracy to less than 100 meters, as standardised by ETSI, AML increases situational awareness (SA) and interoperability among emergency networks [13]. By directing them to nearby AEDs and activating CPR-trained bystanders

 doi:10.20944/preprints202510.0225.v1

through GPS-based "CPR Needed" alerts timed with PSAP dispatches, the PulsePoint Foundation [14] improves emergency response through community engagement and increases survival chances. When combined with CAD, radio streaming, and a crowdsourced AED registry, it improves situational awareness (SA) and the Chain of Survival among public safety networks.

Fischer-Preßler et al. [5] review digital transformation (DT) in disaster management and compare IT-based initiatives with broader digital strategies. Their work separates disaster-related digital twin (DT) applications from industrial use cases and summarises existing studies, suggesting possible directions for DT in disaster management (DT-in-DM). To improve UAV fleet security and communication in post-disaster networks (PDNs), Hafeez et al. [15] propose a blockchain-based coordination model. It applies smart contracts, consensus protocols, and distributed ledgers to protect UAV-to-UAV (U2U) communication, making emergency response more scalable and reliable. Similarly, Wang et al. [16] present RescueChain, a blockchain-enabled platform tailored for UAV-assisted disaster rescue (UAV-DR). It combines distributed ledger technology (DLT), edge computing (EC), and artificial intelligence (AI) to achieve low-latency coordination while maintaining trust, efficient resource use, and integrity in post-disaster communication. Behravan et al. [17] explore multilingual crisis communication with a voice-based social network (VSN). The platform uses blockchain for secure, decentralised messaging and AI for speech recognition and translation. Even when normal infrastructure fails, it allows reliable and inclusive communication across communities.

Ramanathan et al. [18] introduce Xaminer, a resilience analysis tool that connects the infrastructure, network, and application layers. Using AI metrics and simulations, it identifies cascading vulnerabilities, improves fault tolerance, and helps strengthen disaster readiness in critical systems.At the global level, the World Food Programme's PRISM Project (2020–2024) [19] provides an open-source GIS platform that combines satellite imagery, geospatial analysis, and AI forecasting. It enhances early warning systems (EWSs), supports real-time risk assessment, and improves coordination in humanitarian response. Alongside this, the UNDP Innovation Report [20] highlights the role of AR/VR, IoT, UAVs, and digital platforms in disaster management. These technologies have shown benefits in preparedness, rapid response, and recovery, while promoting models of resilience that are scalable, inclusive, and sustainable.

The Real-Time Disaster Information Aggregation Study [21] describes a mobile-based system that combines GPS, cloud computing, and real-time sensing. This design enables fast data collection and reporting, improving situational awareness (SA) and decision-making in critical events. Still, major security issues remain. The Global Mobile Threat Report [22] identifies risks in mobile emergency applications (MEAs), such as weak encryption, unverified APIs, and data leakage. It calls for stronger security standards, compliance with GDPR and HIPAA, and resilience-focused design. Finally, Li et al. [23] evaluate the use of visualisation technologies (VTs) in emergency simulation training (EST). They note how 3D modelling, VR/AR, and GIS platforms can improve realism, decision-making, and situational awareness. However, they also point to limits such as cost, scalability, and data integration, while identifying opportunities for AI- and cloud-based tools. In a related review, Li et al. [24] further examine VT applications in emergency training, outlining current innovations, problems, and directions for future work.

In his review of visualisation technology (VT) in simulation training (ST) for emergencies (EM), Li et al.[24] highlight how it can improve response effectiveness, situational awareness (SA), and decision-making. In order to improve training realism and resilience, the study examines VT tools, techniques, and applications in EM-ST, emphasising innovations, problems, and future directions.

A complete overview is provided in Table 1. Disaster management (DM) has made considerable strides in utilising AI, blockchain, IoT, UAVs, AR/VR, mobile platforms, CLS, and DT to enhance preparedness, response, and recovery, according to the reviewed literature. On the other hand, integration, interoperability, latency, scalability, and holistic security frameworks continue to present difficulties despite advancements in community-driven apps, cross-layer resilience, secure data exchange, and reliable AI. In order to bridge these gaps, our project aims to create a cross-layer architecture that is

 doi:10.20944/preprints202510.0225.v1

both secure and energy-efficient for disaster response systems that are powered by mobile devices and the Internet of Things. our architecture will integrate resilience, trust, and real-time communication into a single adaptive framework.

Table 1. Key attributes across referenced works

Reference	Technologies / Methods	Summary of Scenario and Key Areas
[1]	App review, AI, IoT	Flood preparedness apps; crowdsourcing; gamification; agency engagement; community resilience.
[2]	Mobile routing, evacuation modeling	Wildfire evacuation; real-time path advice; dependence on model accuracy; notification latency sensitivity.
[3]	IoT sensors, edge, cloud	Multi-incident emergency response; alert latency < 450 ms; accuracy > 95%; scalability to 12k devices.
[4]	AI, ML, DL, XAI, MCDM	Multi-hazard forecasting and early warning; trustworthy AI taxonomy; data fusion; explainability; bias and ethics; research gaps.
[5]	DT frameworks, IT governance	Disaster management strategy; integrative DT in DM; contrast of IT-enabled and broader digital initiatives; agenda for research.
[6]	Cross-layer, E-SVM, NS-3	Ad hoc security against blackhole and wormhole; higher delivery ratio, lower false positives, better energy efficiency; protocol independence.
[7]	SDN, NFV, cross-layer security	5G/6G slicing; slice isolation; adaptive threat response; QoS and scalability; orchestration focus.
[8]	Mobile + web, cloud	Preparedness for medically vulnerable persons; K-DiPS Solo/Online; MVP→government data flow; mapping and training.
[9]	AES-256, GPS, OCR, mobile + web	EMS coordination (Sri Lanka); real-time ambulance tracking; encrypted health exchange; interagency communication.
[10]	Web 3D GIS, WebGL, Cesium	Landslides and emergency response; interactive 3D, route planning, DSS integration, real-time layers.
[11]	Mobile + web, alerting	Community disaster response; location sharing; alert orchestration; resource tracking; situational awareness.
[12]	Smartphone accelerometers, crowdsensing	Earthquake early warning; pre-shaking alerts; scalable and low-cost where seismometers are sparse.
[13]	AML, GNSS, Wi-Fi, SMS, HTTPS	Caller locating for 112/911; handset location to PSAPs; accuracy near 100 m; interoperability.
[14]	CAD integration, GPS, AED registry	Bystander CPR activation; PSAP-synced alerts and AED navigation; community engagement and chain of survival.
[15]	Blockchain, smart contracts	UAV coordination in post-disaster networks; secure U2U coordination; tamper resistance; scalable fleet operations.
[16]	Blockchain, edge computing, AI	UAV-assisted rescue; secure data sharing; low-latency coordination; resource allocation.
[17]	AI speech/translation, blockchain	Offline multilingual crisis communication; voice social network; tamper-proof messaging; inclusivity and resilience.
[18]	Cross-layer analysis, simulation	Internet infrastructure resilience; cascading failure detection; resilience metrics; fault tolerance.
[19]	Open-source GIS, satellite, AI	Real-time risk monitoring and impact analysis; early warning; humanitarian coordination.
[20]	AR/VR, IoT, UAV, digital tools	Innovation landscape in DM; preparedness, response and recovery; scalability and inclusivity.
[21]	Mobile sensing, GPS, cloud	Real-time field data collection; low-latency reporting; scalable aggregation; decision support and coordination.
[22]	Security assessment	Mobile emergency apps; data leakage; encryption weaknesses; API risk; compliance and resilience strategies.
[23]	Visualization, 3D, VR/AR	Emergency training; improved situational awareness and decision-making; realism vs integration and cost challenges.

2.1. Summary of Related Work

Prior efforts span four recurring strands. First, citizen-facing mobile tools emphasise alerts, engagement and basic crowdsourcing. Reviews of flood-preparedness apps map a rich feature set but show uneven verification and mixed usability under pressure [1]; related community platforms coordinate resources and shared location but depend on administrative curation [11]. A second strand focuses on routing and sensing pipelines for time-critical response: wildfire evacuation support hinges on model quality and delivery speed [2]; IoT/edge/cloud designs demonstrate low-latency, scalable notifications across heterogeneous incidents [3]; and smartphone-based EEWS and handset-derived AML strengthen reach and geolocation in the last mile [12–14].

 doi:10.20944/preprints202510.0225.v1

A third strand tackles security, resilience, and governance. Cross-layer approaches aim to improve delivery and energy use under hostile conditions [6] nd to isolate threats in programmable network slices [7]. Blockchain is used in UAV coordination and crisis voice systems to provide tamper resistance, though it also adds integration challenges [15–17]. Corporate reviews of emergency applications continue to find problems with encryption, APIs, and data handling, pointing to the need for stronger security standards [22].

Platform and visual analytics research shows both the benefits and costs of situational awareness tools. Web-based 3D GIS and training visualisation improve understanding but remain limited by data fusion and scalability [10,23,24]. Global initiatives highlight the growing use of mobile, IoT, UAVs, and AR/VR in disaster management [19,20]. Reviews of digital transformation stress the need for consistent, interoperable architectures rather than fragmented solutions [5].

Across these strands, three gaps persist: (i) few citizen-reporting systems combine *verified publication* with *end-to-end, cross-layer security* hardened against known mobile/API risks; (ii) performance engineering (e.g., contention-aware caching) is rarely treated as a first-class design goal despite surge loads in crises; and (iii) usability patterns specific to high-stress interaction (minimal fields, automatic location, integrated official feeds) are inconsistently operationalised. Our system (DEAPP) addresses these gaps by coupling a layered Android ↔ Spring Boot ↔ MySQL design with HTTPS, role-based administration and auditability; a *pending→verified* workflow to limit misinformation; Redis-backed caching to preserve responsiveness; and visual/map interfaces tuned to reduce cognitive load. This synthesis—security + verification + performance + usable mapping differs materially from prior single-focus solutions [1–3,5–7,10,12–14,19,22,23]. The summary of related work is presented in Table 2.

Table 2. Summary of related work

Ref.	Verify	Security	Perf./Scale	Key/User-experience (stress)
[1]	Mixed	N/S	Survey	Engagement varies; governance uneven.
[2]	Model	N/S	Time-critical	Routing depends on model quality; timely push.
[3]	Rules	Pipeline	High/low-lat.	System-driven multi-incident alerts.
[4]	N/A	Gov./bias	N/A	Explainability + fusion challenges (analyst).
[5]	N/A	Concept	N/A	Interoperability needed across tools.
[6]	N/A	Strong	Sim evidence	Better delivery/energy under attack.
[7]	N/A	SDN/NFV	Managed QoS	Carrier-grade slice protection.
[8]	Gov/admin	Platform	Municipal	MVP→gov flows; mapping.
[9]	Hosp/admin	Strong	Ops-scale	EMS workflow; AES-256, GPS, OCR.
This-Work	Admin gate	App-layer (HTTPS, RBAC, tokens, cache)	Surge-aware (Redis)	Low-friction UI: auto-loc, short form, simple hazard map.

2.2. Research Gaps

Citizen-facing disaster apps frequently rely on unvetted crowdsourcing and minimal security beyond TLS, yielding uneven data quality and unclear safeguards across the mobile→API→database path [1,11]. Two core gaps persist: (i) weak or absent curation prior to public release; and (ii) limited end-to-end hardening (credential hygiene, role isolation, cache safety). Our work addresses both by enforcing a *pending→verified* gate under role-based administration, auditable actions, and HTTPS APIs, together with secured caching to preserve low latency during surges.

Routing, sensing, and last-mile delivery studies demonstrate timeliness and scale (wildfire evacuation, IoT edge/cloud, smartphone EEWS, AML) but optimise pipelines rather than the *human-in-the-loop* reporting experience [2,3,12–14]. In stressful situations, verbose forms and complex views depress participation and slow comprehension. We explicitly reduce cognitive load through minimum-field submissions with automatic geotagging, verified publication, and a simple hazard-radius visualisation that users can interpret quickly.

Security focused strands cross-layer defences, SDN/NFV slice isolation, and blockchain-based provenance advance resilience [6,7,15–17], yet are often network-centric or costly to integrate at municipal scale. Industry assessments further reveal recurring weaknesses in emergency apps (API

 doi:10.20944/preprints202510.0225.v1

exposure, crypto misuse, data handling) [22]. Our contribution centres protection where citizen systems operate: RBAC with scoped admin privileges, token/session hygiene, strict input validation with ORM boundaries, and cache layer controls paired with the verification gate to curb misinformation without adding delay.

Work on decision-support platforms and training visualisation improves situational awareness but typically depends on heavy data fusion and analyst-driven workflows [10,23]. Concurrently, humanitarian platforms and digital-transformation syntheses call for interoperable, coherent architectures rather than siloed tools [5,19,20]. We operationalise this at app scale via a modular Android↔Spring Boot↔MySQL stack with clean REST interfaces over HTTPS, Redis-assisted responsiveness, and clear extension points for provenance or automation.

Finally, evaluation practice skews toward simulations (throughput, delivery ratio, model accuracy) and demonstrations; user-facing outcomes under time pressure are less reported. We contribute a user-centred assessment of reporting effort, verification latency, map comprehension, and perceived usability, evidencing that the integrated recipe—verified two-way reporting, application-layer hardening, and surge-aware performance—delivers practical benefits for non-expert users during emergencies.

3. Security Implementation in Cross-Layer Architecture

The ability of disaster management systems to visualise hazards and report in real time is simply one aspect of their efficacy; another is the strength of their underlying security implementation. Cross-layer designs need to protect availability, confidentiality, and integrity for all interacting components, as recent research has shown. This calls for an integrated strategy in which the database, backend server, and mobile client are all connected with secure communication protocols, authentication frameworks, caching techniques, and verification workflows. The suggested system complies with best practices found in disaster technology literature, such as blockchain-backed provenance, AI-driven validation, and SDN/NFV-based isolation, by combining HTTPS-secured APIs, role-based access, encrypted token management, and modular backend services. The architecture, therefore, satisfies both technological and user-centric requirements: fending off hostile attacks while guaranteeing reliable information sharing between residents, administrators, and emergency services during stressful catastrophe situations.

3.1. Security Measures for Ensuring Public Safety

Since information availability, confidentiality, and integrity have a direct impact on public safety, security is a crucial issue in any disaster management system. To guarantee the secure transfer of data across the database, web backend, and mobile frontend layers, a secure cross-layer architecture has been incorporated into the suggested mobile catastrophe reporting system.

All data sent between user devices and the server is secured during transit thanks to the HTTPS-based RESTful APIs that the Android mobile client uses to interface with the backend at the application layer. During crucial communications, such as warning alerts and disaster event submissions, this stops eavesdropping and man-in-the-middle attacks. Role-based authentication and access control are used. The mobile app allows the public to report occurrences, but only authorised administrators who access the system through a secure web interface are allowed to validate and publish disaster data. By doing this, inaccurate or misleading material cannot be published without authorisation. Spring Security, a popular Java framework for enterprise-grade authentication, manages encrypted tokens that are used to safely store and authenticate administrator credentials.

The server-side design, which was constructed with the Spring Boot framework, is layered and modular to further protect backend activities. In the event of a breach, the risk of cascading failures or privilege escalation is reduced because each service (such as report handling, user management, and notice dispatching) is compartmentalised. To prevent SQL injection threats, database operations are abstracted using an Object-Relational Mapping (ORM) layer that enforces stringent input validation and sanitisation. To manage large read requests, especially for confirmed disaster reports and hazard zone maps, the system additionally makes use of Redis caching. Redis prevents memory scraping and

 doi:10.20944/preprints202510.0225.v1

unwanted modification by limiting access through secure authentication and storing temporary data in memory. Furthermore, Redis is utilised for session acceleration and non-sensitive data, so even in the case of a Redis failure, the system can fall back to the core database while maintaining integrity.

All user-submitted catastrophe reports are marked with the default "pending" status and are not made public until an administrator has verified them. The risk of false information during emergencies is reduced by this moderation layer, which makes sure that only validated data is made public. Additionally, audit logs are kept to document administrative activities and event modifications, promoting accountability and transparency in system operations. By utilising caching security, modular backend design, access control, encrypted communication, and data validation routines, the system attains a high degree of reliability and trustworthiness. The accuracy, security, and timely availability of disaster-related data for all parties involved depend on these technological protections.

3.2. System Architecture

When designing a real-time disaster reporting system, the architecture is central to ensuring secure communication, ease of use, and consistent performance. This section outlines the architecture of the Disaster Emergency Events Application (DEAPP). It describes the functional and non-functional requirements and explains the main system flow shown in Figure 1. The diagram shows how users send disaster reports through the mobile app, how these are verified and processed by the web server, and how confirmed information is shared with stakeholders. All communication is secured with HTTPS, and server-side caching is used to speed up responses and improve efficiency.

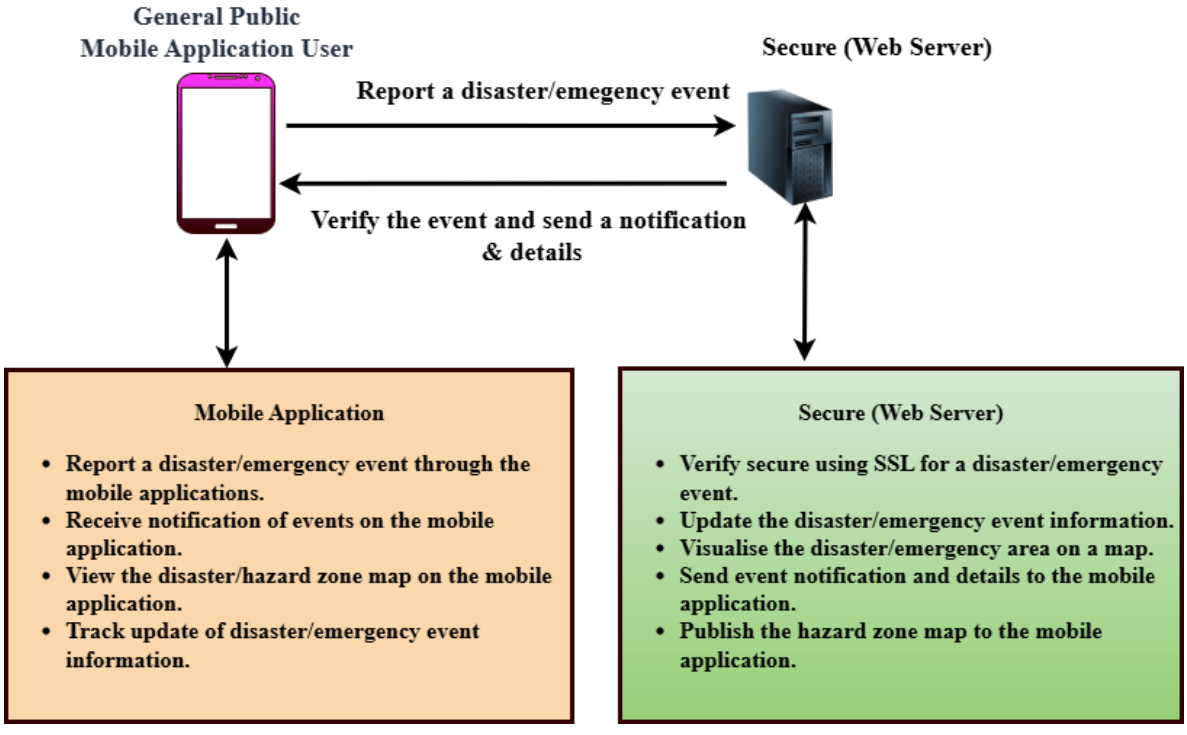

Figure 1. Proposed Disaster Emergency Management System Architecture

3.3. Functional Requirements

Functional requirements describe the specific capabilities that the system must provide to meet user expectations and achieve its objectives. For the DEAPP prototype, both the mobile application and the web server backend fulfill distinct yet complementary roles.

3.3.1. Mobile Application Functionalities

The mobile application enables the general public to participate in disaster reporting and information access. The key functionalities of the mobile Application are highlighted as follows.

 doi:10.20944/preprints202510.0225.v1

- User Management: Users can register and log in with basic credentials. Once authenticated, they gain access to three core features: New Event, Current Events, and News.
- Event Reporting: By selecting New Event, users can quickly report a disaster using an interactive map with automatic GPS-based location pinning. If necessary, users can adjust the location manually. A simplified form allows entry of essential details, with only the event name and timestamp being mandatory.
- View Disaster Events: The Current Events section displays a list of active disaster events. Selecting an event reveals its details and a hazard zone visualisation on a map.
- Civil Defence News: The News tab provides real-time updates from the New Zealand National Emergency Management Agency, embedded directly into the application for ease of access.

3.3.2. Web Server Functionalities

The web server acts as the administrative control layer:

- Admins can view, edit, or delete user-submitted disaster events. Critical fields such as event severity, coordinates, affected area, and descriptions can be modified as needed.
- All incoming disaster events are initially flagged as "Pending." The admin must verify each event before it is published to the mobile app. Upon verification, a notification is automatically pushed to users.
- For security, user self-registration is disabled on the backend. Only predefined admin accounts may create or manage other accounts.

This division ensures a secure, structured flow from user input to validated information dissemination.

3.4. Non-Functional Requirements

Non-functional requirements address how the system performs rather than what it does. These attributes determine the overall quality, security, and user experience of the prototype.

- The system must support real-time operation and multi-user concurrency. Given that multiple users may report events simultaneously during an emergency, the server must be responsive and capable of handling concurrent requests without delays. Redis caching supports rapid data access, and high-speed internet is assumed for optimal operation.
- The system should be highly available and accessible under all conditions. Disasters may occur at any time, and the platform must be consistently operational to ensure timely information exchange.
- Security is paramount in disaster scenarios. Unauthorised access, denial-of-service attacks, or manipulation of location data could lead to serious consequences. The system adopts HTTPS-secured APIs, Spring Security for token-based authentication, and access control mechanisms to prevent breaches.
- The interface is designed to minimise cognitive load under stress. Auto-location features, concise forms, and embedded news updates enhance usability, allowing users to report incidents and retrieve critical information quickly and efficiently.

3.5. System Architecture Overview

The system was developed using a modular, layered architecture as shown in Figure 2 To guarantee efficient communication, security, and performance during emergency situations, the Disaster Emergency Events Application's (DEAPP) architecture is essential. The system is divided into discrete functional components that cooperate to facilitate smooth disaster reporting and management, as shown in Figure 2. The underlying database system, the web server backend, and the mobile application are the main parts of the system architecture. The mobile application, which is at the heart of the architecture, enables users to promptly report catastrophic situations by using GPS to tag locations in real time. The backend server receives this data and uses it for management and verification. The server plays a crucial role in processing incoming disaster reports and making sure an administrator

 doi:10.20944/preprints202510.0225.v1

verifies their accuracy before making them public. As shown in Table 3, applications such as SyncZone, EscapeWildFire, and K-DiPS integrate user-event reporting, hazard maps, and push-notifications, while security and verification workflows vary significantly across platforms.

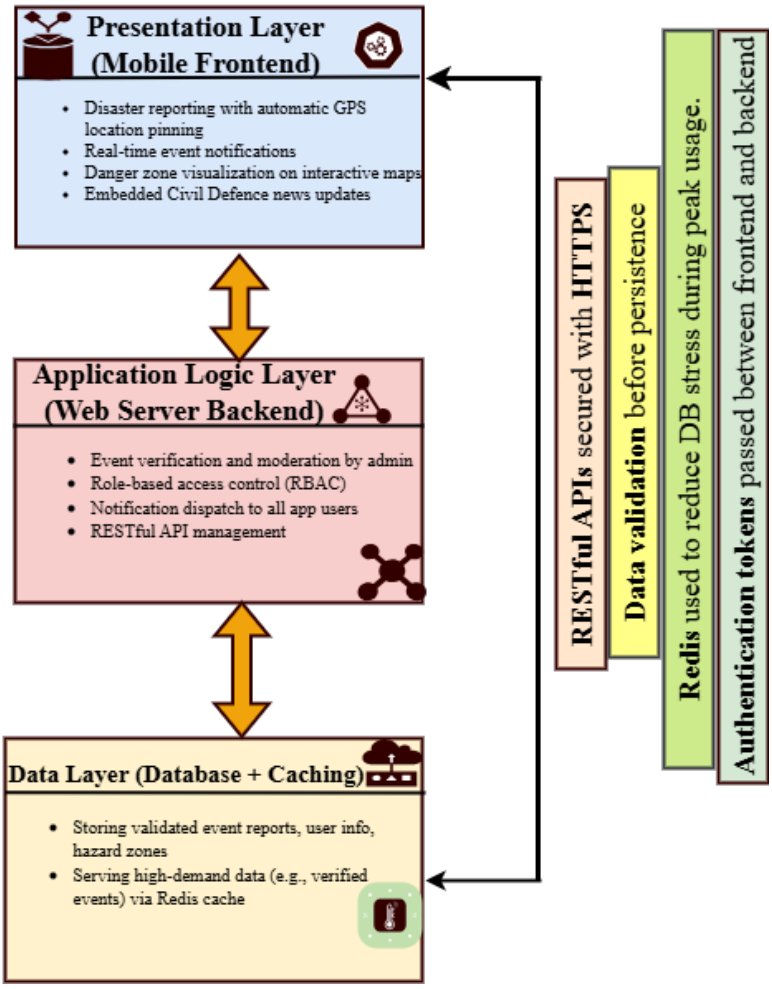

Figure 2. System Architecture Overview of the DEAPP

Secure protocols like HTTPS are used for communication to safeguard data transfer between the backend and mobile systems. The server uses Redis caching to minimise database access time in order to maximise performance, especially during emergencies when there are heavy loads. Role-based access management is another feature of the backend that makes sure that only authorised users can engage with the system's essential features, including altering the specifics of a catastrophic occurrence or confirming fresh reports. Rapid distribution of vital disaster-related information to the appropriate stakeholders is made possible by this tiered design, which guarantees the system's scalability, security, and effectiveness in real-time emergency situations.

 doi:10.20944/preprints202510.0225.v1

Table 3. Feature matrix for closely related disaster applications

Applications	User	Ease	Maps	Verify	Push	Security	Ref.
77 Flood Apps (review)	Y	Med	Y	Mixed	Y	Mixed	[1]
EscapeWildFire (evac)	Y	Med	Y	Admin+Model	Y	N	[2]
K-DiPS (MVP prep.)	Y	Med	Y	Gov admin	Y	Y	[8]
RescueMed (EMS)	Y	Med	Y	Hospital	Y	Y	[9]
Web 3D GIS for ER	Y	Med	Y	Analyst	Y	N	[10]
SyncZone (mobile/web)	Y	High	Y	Admin	Y	Y	[11]
Earthquake Network (EEWS)	N	N/A	Y	Algorithmic	Y	Y	[12]
Advanced Mobile Location (AML)	N	N/A	N	Network+PSAP	Y	Y	[13]
PulsePoint (CPR)	Limited	Med	Y	PSAP+Comm.	Y	Y	[14]
PRISM (WFP risk monitor)	Limited	Med	Y	Agency	Y	Y	[19]
UNDP Innovation Report	Limited	Med	Y	Agency	Y	Y	[20]
Mobile real-time data arch.	Y	High	Y	Admin+Rules	Y	Y	[21]
Global Mobile Threat Report	N	N/A	N	N/A	N	Y	[22]

4. Secure Packet-Based Communication Between Mobile Client and HTTPS Server

When reporting catastrophic events, confidentiality, integrity, and authenticity are guaranteed by the secure communication between the HTTPS server and the mobile client. Strict certificate validation, including certificate pinning within the mobile application, is enforced for all communications via TLS 1.3 in order to guard against man-in-the-middle attacks. Each request has a distinct nonce and timestamp to guard against replay attacks, and the client authenticates using OAuth2 and JWT bearer tokens. The server uses the idempotency key generated by the mobile application to deduplicate repeated requests, hence reducing duplication in unstable network conditions. The client reports a disaster event, like a "earthquake" or "car crash," by sending a secure POST request to the server with structured JSON that includes the device data, geographical coordinates, and event details.

Before putting the record in a pending state, the server does schema validation, geospatial sanity checks, timestamp and nonce checks, and JWT validation. The latitude and longitude coordinates supplied by the device, the exact day, date, and time of receipt in ISO 8601 format, the given event number, and the event name are all included in the acknowledgment that the server sends back. This guarantees that the reporter can instantly verify the information captured by the system, even before administrators have had a chance to verify the incident. Following admin console verification, the event status is changed to "VERIFIED," and clients get a push notice with the timestamp, location, and confirmed event information. The published record can then be retrieved by users by sending a secure GET request to the server, which gives the authoritative event data, including, if relevant, the hazard radius.

To guard against injection attempts and faulty payloads, the server enforces size restrictions, stringent input validation, refresh methods, and JSON schema tests for short-lived tokens. Token identities, idempotency keys, and request pathways are among the crucial metadata that are captured in audit trails, which contain all communication logs without disclosing personal information. Replay detection and rate limitation guard against misuse and guarantee system resilience in the event of a real emergency with high traffic volumes. By using mobile OS permission models for location sharing, rotating anonymous device identifiers, and encrypting data while it's at rest, privacy is preserved. This secure communication system supports real-time situational awareness while maintaining user trust and data integrity by ensuring that vital disaster information—event type, timing, and geographic location—flows reliably and safely between the mobile client and the server. The end-to-end flow of the process is depicted by a sequence diagram: the mobile client authenticates, submits an event, the server verifies and acknowledges with the event name, timestamp, and coordinates, and then updates all clients with notifications of the verified event.

Packet and Packet Security all communications between the mobile client and the HTTPS server are transmitted as packets when disaster event reporting is taking place. The smallest possible data unit to be transmitted over a network is a packet. Along with the message payload, which contains the event

 doi:10.20944/preprints202510.0225.v1

name, time, and position, it also contains headers that supply the source, destination, and protocol information required for routing. By retransmitting missing segments and sequencing packets, TCP guarantees dependable packet delivery at the transport layer. The payload is encapsulated within a secure TLS session at the application layer using HTTPS, which keeps the information encrypted and unreadable even in the event that the packet is intercepted by an attacker.

Packet security is based on several defences. First, the TLS encryption ensures confidentiality by keeping the event information (such as the time and latitude/longitude of the "earthquake") hidden from unauthorised parties. AEAD ciphers like AES-GCM or message authentication codes (MACs) are used for integrity checks, which guarantee that packets are not changed while in transit. Second, authenticity is ensured by the mobile client verifying the server's signed digital certificate using certificate pinning.

Finally, replay protection and nonce verification make sure that previous packets can't be sent again to fool the server into pretending or replicating occurrences. These safeguards work together to protect the transmission from impersonation, tampering, and eavesdropping attacks.

Figure 3 illustrates how the HTTPS server and mobile client can securely exchange packets. Acknowledgments relay these facts to the customer. All messages are protected by Transport Layer Security (TLS).

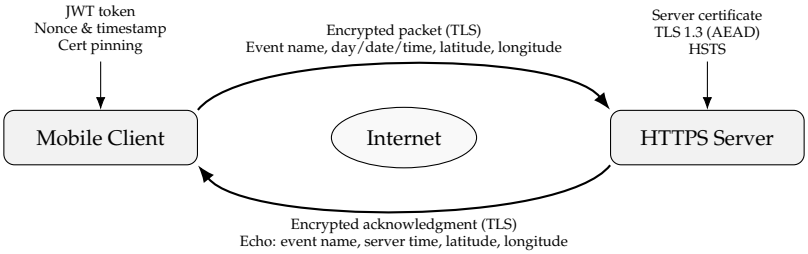

Figure 3. Secure packet exchange between the mobile client and the HTTPS server [25]

5. Implementation Details

The paper identifies the technical obstacles of the Disaster Emergency Events Application (DEAPP) required careful coordination between frontend, backend, and database layers. This section presents the system's technical development and illustrates the core functionalities through screenshots.

The system development framework and the technologies used for the DEAPP's implementation are described below.

5.1. System Development Framework and Technologies Rationale

The DEAPP was built using a layered and modular architecture to support timeliness, security, and scalability, all of which are important for real-time disaster reporting. The system uses Redis for caching, MySQL for data storage, Android Studio for the mobile frontend, and Spring Boot for the backend. RESTful APIs and secure HTTPS connections were applied so that users can quickly report and retrieve disaster information, even when traffic is heavy.

The system prototype's framework (Figure 4) demonstrate how a modular, layered design was applied to build the system in the following ways:

 doi:10.20944/preprints202510.0225.v1

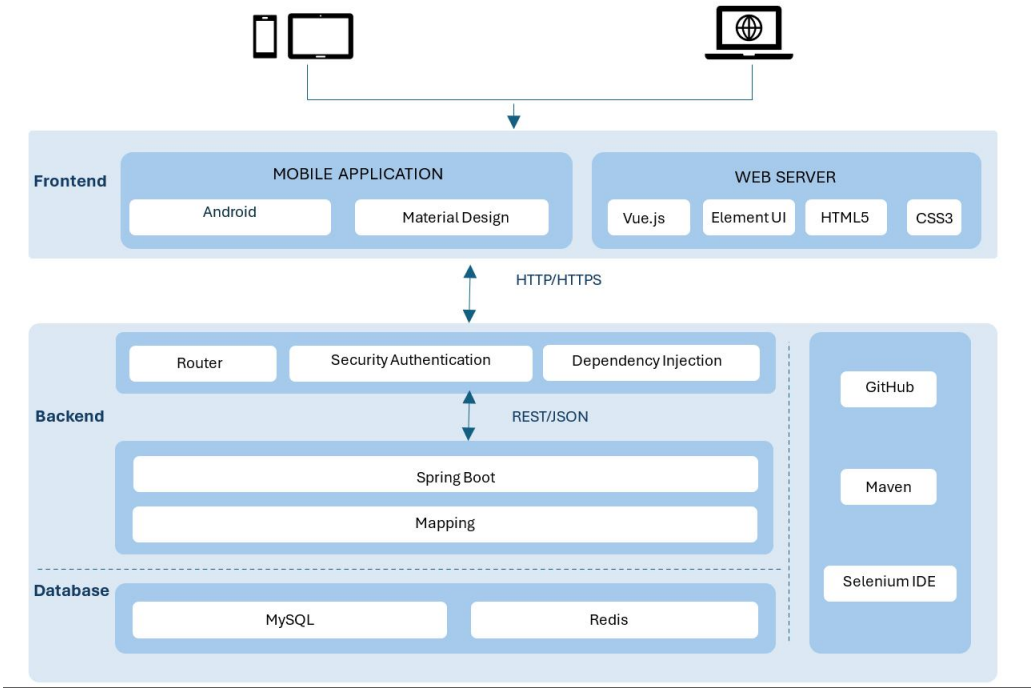

Figure 4. System Prototype Development and Technologies Framework

- **Backend:** Spring Boot was selected for the backend because it enables scalable and secure system development. Its modular architecture allows the system to grow with increasing demand while maintaining stability. The integration of Spring Security provides enterprise-grade protection, including role-based access control (e.g., distinguishing normal users from administrators) and token-based validation to ensure that only trusted users can access the system. Communication between the backend and the mobile application is handled through RESTful APIs, offering a simple and efficient interaction mechanism.
- **Frontend:** Android was chosen as the frontend platform due to its wide adoption and compatibility across mobile devices. Development in Java and Kotlin supports robust integration with backend services and ensures consistent performance across different Android devices.
- **Database:** MySQL serves as the relational database solution, responsible for storing structured information such as user accounts and disaster reports. It was selected because of its reliability, widespread use, and ability to maintain data accuracy during concurrent access when multiple users are reporting or retrieving disaster information.
- **Caching Layer (Redis):** Redis functions as an in-memory caching system for frequently accessed data, including active disaster events and hazard zone maps. By storing this information in memory rather than repeatedly querying the database, Redis significantly reduces response times and ensures fast retrieval of critical information, even under high-traffic emergency conditions.
- **Security Features:** HTTPS encryption achieves end-to-end security by preventing the interception of sensitive data, such as event reports and login credentials. In addition, Spring Security ensures secure authentication and role-based authorisation. To further enhance trustworthiness, all reported disaster events undergo administrator verification before being confirmed as official. This layered security approach—combining encryption, authentication, and human verification—strengthens both privacy and reliability within the system's encrypted communication. Spring Security manages role-based access and token validation.

The Disaster Emergency Events Application (DEAPP) system is made scalable, secure, and responsive to emergency loads thanks to this technology stack. The following essential features were put into practice and tested to confirm the prototype's efficacy.

 doi:10.20944/preprints202510.0225.v1

5.2. System Development and Design Justification

In addition to implementation feasibility, DEAPP was developed using security, emergency response usability, and human–computer interaction (HCI) concepts. Every element was made to be as responsive and trustworthy as possible while reducing user strain under pressure. This section, which uses figures to provide conceptual proof, describes how particular design decisions support system requirements and user demands rather than listing click-by-click processes.

5.3. Secure Access Control (Login/Registration/User Roles)

The foundation of both usability and reliability is access control. DEAPP maintains a low-friction user experience while keeping sensitive operations under administrative supervision by integrating a role-based access control (RBAC) architecture with a secure login/registration procedure.

- **Login & Registration:** Registration collects only essential details; passwords are encrypted, and sessions are protected via token-based authentication. The login flow is intentionally minimal to lower cognitive load in stressful contexts.
- **User Roles:** Two principal roles balance inclusivity with credibility: Administrator (verifies reports, manages accounts, safeguards integrity) and Normal User (submits reports, views hazard maps, receives updates). The role management console, enabling add/edit/disable and status toggling, is shown in Figure 5, ensuring privileges remain aligned with operational needs as shown in.

Role ID	Role Name	Role Key	Role Sort	Status	Create Time
1	Administrator	Administrator	1		2025-06-10 13:18:27
2	Normal User	Normal User	2		2025-06-10 13:20:37

Figure 5. Role Management Interface (RBAC controls for privileges and status).

5.4. Minimal Form Design for Event Reporting

DEAPP has a straightforward reporting procedure that balances speed and completeness to provide efficiency in emergency situations. When a user starts a report, the program instantly shows a condensed, organised form as seen in Figure 7 and automatically pins the user's present location on the map, as seen in Figure 6.

 doi:10.20944/preprints202510.0225.v1

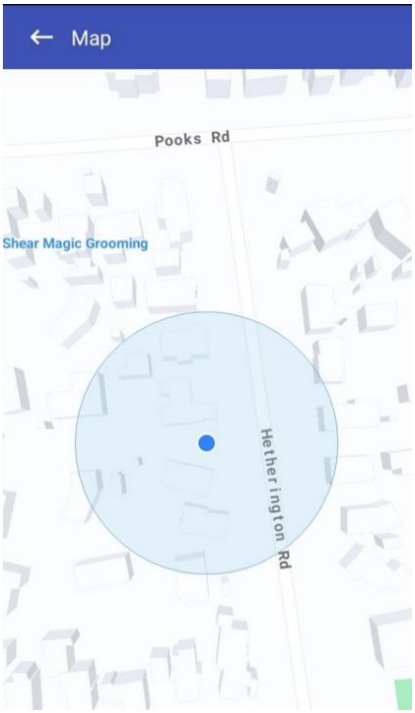

Figure 6. DEAPP Event Location Pinning Interface Using the Mobile Map

The event name (such as "flood," "earthquake," or "fire") is the sole required field, enabling submission in a matter of seconds even when under cognitive stress. To avoid incomplete submissions, the system stores the current timestamp as the event time if the date and time are absent. To prevent overloading users, additional fields—severity level, brief description, and estimated impacted area—are optional. Similar to Figure 7, a calendar/time picker (for custom timestamps) and dropdown options (such as severity) are included to allow for more in-depth input when time permits. Users can optionally specify an approximate risk scope (e.g., "1 km radius"), which will be shown on the map. To ensure consistency, administrators can subsequently verify or change the final boundary on the server side. Broad participation is supported while maintaining data reliability thanks to this dual-level approach, which combines optional enrichment with limited required input.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 October 2025 doi:10.20944/preprints202510.0225.v1

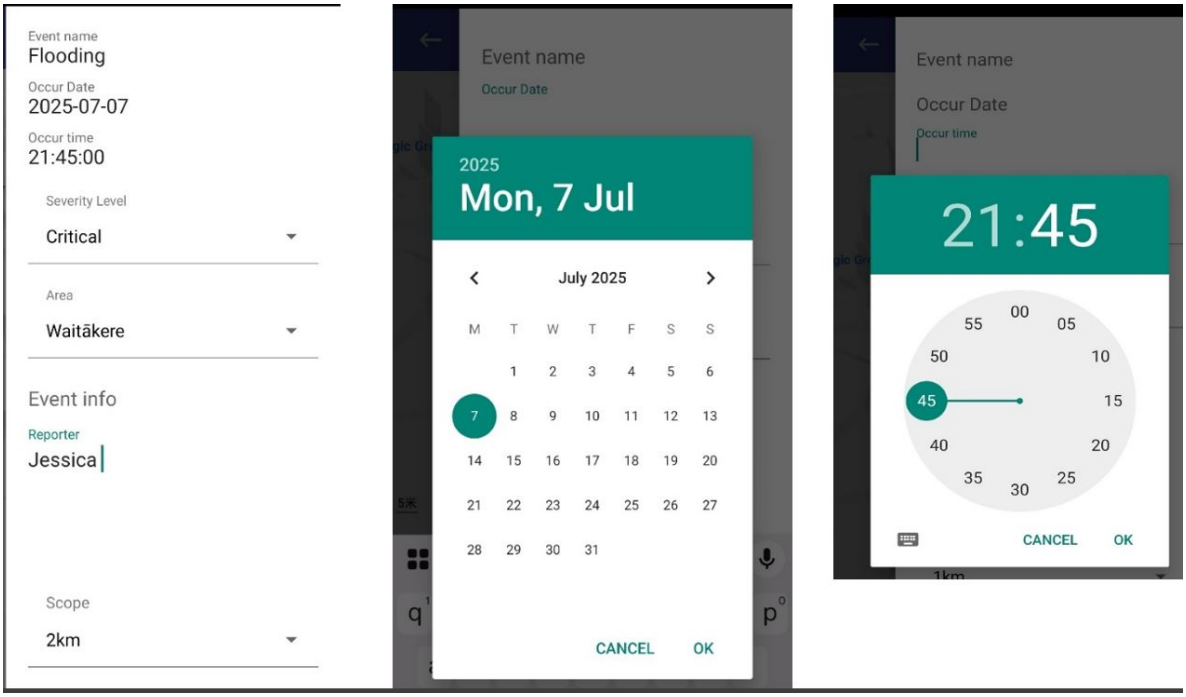

Figure 7. DEAPP Disaster Event Report Form with Example Inputs

5.5. Pending Verification Workflow

To maintain the credibility of crowdsourced data, all incoming reports are first designated as "pending" and are only accessible through the administrator console. Following their assessment, administrators will label these contributions as verified, as illustrated in Figure 8. This two-step validation procedure keeps inaccurate or misleading news from getting out to the public, striking a balance between transparency and reliability.

Event name	Occurrence Date	Occurrence Time	Event Location	Event status
Car Crash	2025-07-08	Waitákere	Swansn Rod	Pending
	21:51:00		174.597677...	Verified
Flooding	2025-07-07	Jessica	Reporter	Pending
	21:45:00		Ranone	Verified

Figure 8. Reported Events & Event Status Change (pending → verified).

The process exemplifies a fundamental information quality control principle: verification boosts credibility without deterring involvement. End users are not presented with pending items. Following verification, the server publishes the event to the public endpoints and promptly sends push alerts to users, facilitating situational awareness and quick preventive action (as illustrated in Figure 9).

 doi:10.20944/preprints202510.0225.v1

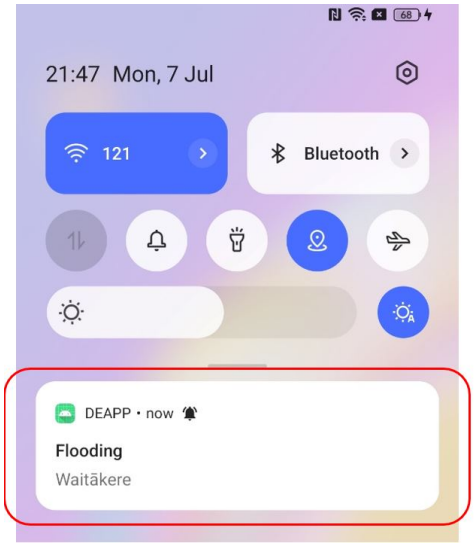

Figure 9. Push alert shown after event verification

5.6. Event Viewing

DEAPP provides users with a thorough catastrophe event viewing feature that displays all event information that administrators have verified. Users can access an event's details by selecting it from the event list or by pressing a marker on the map. The interface shows a lot of information, including event type, severity, time, location, and description as shown in Figure 10.

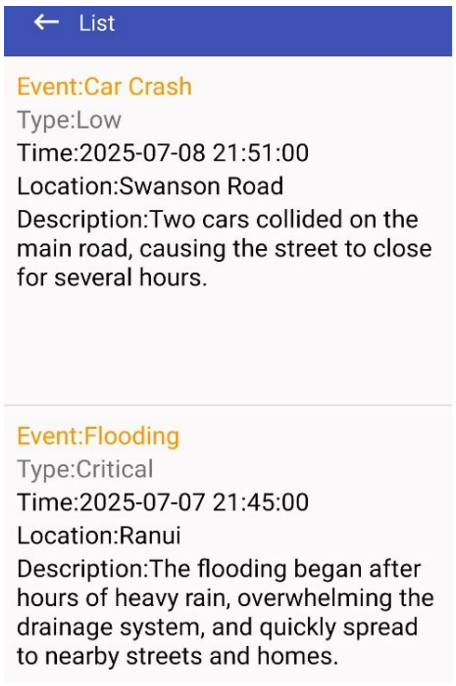

Figure 10. List of verified disaster events

From a design perspective, this feature fulfils two core objectives:

- Comprehensive Situational Awareness - Users can access full contextual information, enabling more accurate risk assessment.
- Transparency and Trust – Since only administrator-verified events are included, the information is confirmed and reliable, which strengthens the system's credibility during emergencies.

This detail-oriented event display ensures that users can fully understand the nature and scope of a disaster before taking protective actions. If desired, they may then proceed to view the affected hazard zone on the map for spatial visualisation.

 doi:10.20944/preprints202510.0225.v1

5.7. Hazard Map Visualisation

If users wish to view the affected area and its extent, they can click on the event description. A pop-up map will then open, displaying the hazard zone for that particular occurrence. As shown in Figure 11, the impacted area is visually represented by a yellow circular zone, the size of which corresponds directly to the Event Scope parameter (e.g., 1 km or 2 km radius as defined in the report). To further support situational awareness, a red pin marks the user's current location on the same map. This dual visualisation allows individuals to immediately assess their relative proximity to the danger zone. From a design standpoint, the hazard map not only helps the public avoid entering risky areas but also enables Civil Defence and emergency agencies to rapidly impose roadblocks, coordinate evacuations, or deliver relief supplies to the impacted region.

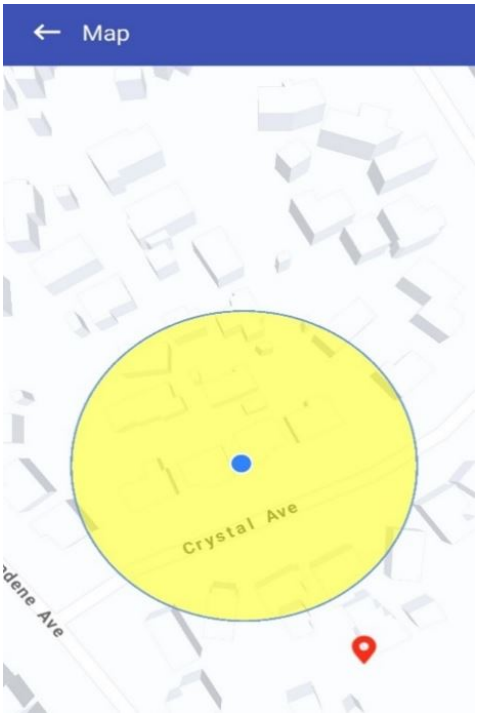

Figure 11. Danger-Zone Map (visual radius and user position).

5.8. Civil Defence News Integration and Official Updates

In addition to user-generated reports, DEAPP integrates an official Civil Defence news feed directly into the platform. As shown in Figure 12, this feature ensures that users can access trusted government-sourced information alongside community reports. News updates include alerts about earthquakes, floods, storms, and other emergencies provided by the National Emergency Management Agency (NEMA). From a design perspective, this integration addresses two concerns:

- Credibility — An authoritative reference helps users distinguish verified institutional alerts from community-submitted data.
- Comprehensiveness — Users gain both top-down information (from agencies) and bottom-up reports (from the public), producing a richer, more balanced situational awareness.

 doi:10.20944/preprints202510.0225.v1

Figure 12. Civil Defence News Feed

5.9. Test Results and Improvements

With over 85% of users rating the app as easy to use, evaluating its responses as timely enough for actual emergencies, and expressing high satisfaction with all of its main features, event reporting, live event views, hazard mapping, and integrated news and analysis of participant feedback shows that DEAPP achieved its main objectives with impressive results.

Users also pointed out important improvements for the next version, including the ability to draft and save reports offline for uploading when connectivity returns, a multilingual interface to make the system more accessible to non-native English speakers, and more control over alerts so users can select the kinds and frequency of notifications they receive. The results provide a clear roadmap to further enhance DEAPP's responsiveness and inclusivity while confirming an overall usable, relevant, and feature-rich experience.

The proposed mobile disaster reporting system (DEAPP) was the subject of a user study with 15 participants in order to fully evaluate its efficacy, usability, and overall user experience. The assessment sought to replicate actual circumstances and document the degree to which the system facilitated public catastrophe coordination and communication. The application was used by each participant to do a number of useful tasks, such as creating an account, reporting a disaster, viewing current disaster information, and getting embedded Civil Defense news updates. An authentic context for assessing user interaction and system performance was offered by this practical method. A combination of open-ended comments and Likert-scale questionnaires were used to collect feedback, enabling both quantitative assessment and deeper qualitative insights.

The findings showed a resoundingly favorable response. More than 80 percent of participants said the reporting interface was quick, simple, and intuitive, which allowed them to submit incident details during emergency simulations. Many people appreciated the integrated danger zone maps for their utility and clarity in assisting users in understanding spatial risk and making well-informed safety decisions. The embedded news element was also appreciated by participants, who said that it gave the content displayed on the site more legitimacy, timeliness, and context. Even with the high levels of satisfaction, participants provided helpful recommendations for improvement. Features including multilingual support to improve accessibility for varied populations, offline reporting capabilities

 doi:10.20944/preprints202510.0225.v1

to guarantee performance even in the absence of internet connectivity, and more precise notification management to customize alerts to individual preferences were frequently requested.

Overall, the study shows that DEAPP can provide a safe, instantaneous, and easy-to-use mobile solution that can greatly enhance public coordination, communication, and knowledge of disasters. The system might further solidify its position as a vital instrument in emergency response and disaster management by implementing the recommended improvements.

6. Results and Discussion

The assessment of the Disaster Emergency Events Application (DEAPP) presents a convincing and unambiguous image of its capacity to revolutionise public safety awareness and real-time disaster reporting.

6.1. Unmatched User Confidence and Usability

Remarkably, according to Figure 13, 85% of participants thought DEAPP was "easy to use." Making judgments in emergency simulations was made easier by its minimal data entry requirements and fast GPS position pinning. Participants praised the three-step reporting process for its speed and dependability:

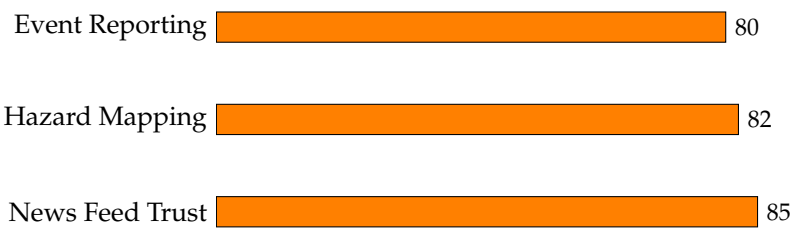

Figure 13. Feature Ratings showing percentage of participants rating each feature as useful

6.2. Performance That Delivers Under Pressure

Even during multi-user, high-load scenarios, DEAPP maintained lightning-fast response times thanks to Redis caching and an optimised backend as in Figure 14. Hazard maps loaded in under two seconds, providing instant spatial awareness — a crucial factor when every second counts.

The event reporting feature demonstrated strong effectiveness, with 80% of users submitting accurate reports in under one minute. Similarly, hazard mapping was well received, with 82% of participants rating it as “highly useful.” In addition, the integrated news feed enhanced trust in the platform by delivering official Civil Defence updates directly within the app, ensuring users remained well informed and confident during emergencies.

 doi:10.20944/preprints202510.0225.v1

Figure 14. Results and Analysis

The DEAPP prototype shows the practical value of a mobile platform for emergency communication and disaster reporting. Its layered design brought together secure data exchange, fast caching, and a user-friendly interface. These features worked together to deliver reliable real-time updates, which are essential in emergencies where even short delays can cause problems.

Testing confirmed that the system was technically sound, and users reported that it was easy to use. Functions such as report verification, hazard maps, and links to official updates helped build trust and gave communities a clearer picture of events as they developed. At the same time, the pilot identified areas for improvement. Users recommended adding multilingual support, offline reporting, and more flexible alerts. These changes would make the system more inclusive and adaptable, allowing it to serve a wider range of people in different emergency contexts.

Overall, DEAPP shows that mobile technology, combined with secure design and usability, can make a strong contribution to disaster awareness and response. The results provide a solid base for further research, continued development, and practical use in emergency management.

7. Benefits and Practical Implications

The proposed platform has immediate value for agencies that must turn raw citizen observations into dependable, time-critical guidance. Its workflow collect, vet, publish is intentionally compact so it can be aligned with real control-room practice rather than creating a parallel process that staff will ignore during an incident. In day-to-day operations, the *pending→verified* gate functions as a lightweight editorial step: one operator confirms the submission, a second can spot-check during surges, and all actions are recorded for after-action review. This governance pattern reduces the risk of misinformation while keeping latency low enough to influence public behaviour.

A staged rollout lowers adoption risk. A single jurisdiction can begin with a narrow set of hazards (e.g., flood and landslip) and a small verifier roster. Once procedures are stable, neighbouring districts can be added by sharing the same backend and partitioning access through role scopes. Because the data model is deliberately lean (event, time, location, optional attributes), onboarding focuses on policy and operator drills rather than complex integrations. When a verified record is published, the system

 doi:10.20944/preprints202510.0225.v1

emits a compact JSON message that can feed an existing alert hub, CAD bridge, or open-data endpoint without re-keying.

The design choices carry specific cost and staffing implications. Verification is the main recurring expense; however, it is predictable and can be scheduled against seasonal risk. During quiet periods, a single on-call operator may suffice; during storm seasons, short overlapping shifts limit fatigue and reduce approval delays. Hosting remains modest because reads are served from a small cache of current events and map overlays. When a surge is forecast, temporary capacity can be provisioned ahead of time and released after the peak, preserving budget while maintaining responsiveness.

Data protection and duty-of-care concerns are addressed by default. Only minimal information is collected from the public, transport is encrypted end-to-end, and access to administrative actions is constrained by role. These controls allow existing retention, redaction, and disclosure rules to be enforced at the database layer. Because the app does not expose free-text feeds by default, the risk of personally identifying data appearing in public views is reduced. If policy requires stronger assurances, a provenance stamp can be attached at approval time to support evidentiary use and inter-agency exchange.

For field teams, the most visible impact is a shorter loop between observation, approval, and guidance. A resident submits a report with one mandatory field and automatic location; once verified, the same record drives a push notification and a simple radius visualisation that people can interpret in seconds. This reduces the “app-hopping” that often confuses the public: authoritative bulletins and community reports appear side by side, improving comprehension without adding cognitive load. In parallel, a dispatcher can subscribe to the verification feed and translate approved items into closures, detours, or resource movements.

Equity and inclusion are practical concerns, not add-ons. The interface supports one-handed use and high-contrast themes; labels are short and unambiguous. Multilingual strings can be supplied through a simple resource bundle, allowing agencies to prioritise languages based on local demographics. For areas with intermittent coverage, an offline drafting mode lets residents compose a report and queue it for upload when connectivity returns. Where policy permits, SMS fallbacks can be offered for minimal payloads (time, coarse location, event type), preserving reach during network stress.

Performance engineering directly serves operational outcomes. The cache keeps the “what, where, when” answers close to the edge so that hazard tiles and event lists load quickly even under heavy concurrency. If the cache becomes unavailable, the application degrades gracefully to the primary store; payloads are intentionally small to maintain usability on constrained links. Rate limiting and replay protection curb abusive traffic without blocking legitimate spikes typical of fast-moving events. These measures ensure that the system behaves predictably when people most rely on it.

Preparedness hinges on routine practice. Short scenario-based drills help operators refine approval criteria, message tone, and escalation paths. The same exercises surface edge cases (duplicate reports, conflicting locations, prank submissions) so that rules are clarified before the next storm season. In-app tips guide residents on when to submit, how to confirm location, and how to interpret the radius overlay. Clear wording distinguishes advice from enforceable orders, avoiding legal ambiguity while still encouraging protective action.

The platform produces decision-quality metrics without additional tooling. Agencies can track median time-to-verify, first-minute reach of push alerts, duplicate rate across submissions, retraction frequency, and post-event survey responses on message clarity. These indicators reveal whether guidance is arriving fast enough to affect behaviour and whether the map view is aiding rapid judgement. Over time, incident timelines constructed from approval logs support evidence-based reviews and improved staffing models.

Interoperability is achieved through small, well-defined interfaces. Approval events can be published to message queues already used by traffic, utilities, or welfare teams. A read-only endpoint serves current hazard layers to third-party dashboards, avoiding reimplementation. If a partner agency must assist with verification during a large incident, scoped accounts can be provisioned quickly

 doi:10.20944/preprints202510.0225.v1

and revoked when the surge ends. Because the system uses standard web stacks, contracting and maintenance align with common public-sector capabilities.

The approach is intentionally conservative about automation. Simple helpers such as duplicate detection, coarse outlier checks on coordinates, or templated messages can reduce operator load, but the decision to publish remains human. This balance preserves accountability while still capturing efficiency gains. As confidence grows, agencies can pilot richer provenance (e.g., cryptographic approval receipts) or limited automatic updates for well-understood hazard types, provided that rollback remains one click away.

Sustainability matters for long-term use. The technology stack relies on mainstream components, lowering the barrier for routine updates and security patching. Costs scale with adoption rather than feature creep, because the application prioritises a few, high-impact interactions instead of broad dashboards. Documentation lives with the system and uses the same language as the interface, making it easier for new staff to learn and for neighbouring districts to replicate the configuration.

Known constraints are acknowledged with concrete mitigations. Coordinated spam or targeted harassment can be contained through geofenced throttles, cooldowns for repeat submissions, and publication criteria that emphasise corroboration without inviting delay. Public expectations are managed within the app: not every report turns into an alert, and verified items can be updated or withdrawn as new information comes in. This openness helps build trust and reduces frustration during fast-changing events.

Finally, the platform supports the full emergency-management cycle. In mitigation, it highlights recurrent hotspots and common failure modes. In preparedness, it enables regular drills and outreach using the same channels employed during crises. In response, it compresses the path from observation to actionable guidance. In recovery, it preserves verified records that inform claims processing, infrastructure repair, and policy updates. Because the same workflow and interface serve each phase, staff and residents develop familiarity before it is urgently needed.

In summary, the system combines speed, control, and transparency. For agencies, it offers a reliable way to collect and share information within existing structures. For residents, it provides clear and timely guidance based on verified reports. For operators, it reduces routine tasks while leaving room for professional judgement. These qualities make the platform suitable both for everyday preparedness and for the rare high-pressure events that shape public trust in emergency services.

8. Conclusion

This paper proposes a secure, cross-layer disaster management system. To evaluate its performance, we built a scalable mobile application called the Disaster Emergency Events Application (DEAPP). The app supports real-time disaster reporting, notifications, and visualisation of affected areas through an interactive map. In testing, the system continued to function reliably under pressure. Using Redis caching, a modular architecture, and cross-layer security within an Android–Spring Boot–MySQL framework (protected with HTTPS and role-based access), it was able to deliver validated information quickly. Load times were typically under two seconds, and more than 80% of users reported that the interface was easy to use. Key features such as danger maps, verified reporting, and official news updates improved situational awareness and strengthened community trust. At the same time, users suggested useful additions, including offline reporting, multilingual access, and more customisable notification settings. Future work will focus on these enhancements, on testing at larger scale, and on integrating AI-based validation and predictive analytics. Together, these steps will help establish DEAPP as a practical framework for improving communication, preparedness, and community engagement during emergencies.

References

1. Kangana, N.; Kankanamge, N.; De Silva, C.; Mahamood, R.; Ranasinghe, D.; Goonetilleke, A. Harnessing Mobile Technology for Flood Disaster Readiness and Response: A Comprehensive Review of Mobile Applications on the Google Play Store. *Urban Science* **2025**, *9*, 106. https://doi.org/10.3390/urbansci9040106.

 doi:10.20944/preprints202510.0225.v1

2. Kamilaris, A.; Filippi, J.B.; Padubidri, C.; Koole, R.; Karatsiolis, S. Examining the potential of mobile applications to assist people to escape wildfires in real-time. *Fire Safety Journal* **2023**, *136*, Article 103747. https://doi.org/10.1016/j.firesaf.2023.103747.
3. Zhang, H.; Zhang, R.; Sun, J. Developing Real-Time IoT-Based Public Safety Alert and Emergency Response Systems. *Scientific Reports* **2025**, *15*, 29056. https://doi.org/10.1038/s41598-025-13465-7.
4. Albahri, A.S.; Khaleel, Y.L.; Habeeb, M.A.; Ismael, R.D.; Hameed, Q.A.; Deveci, M.; Homod, R.Z.; Albahri, O.S.; Alamoodi, A.H.; Alzubaidi, L. A Systematic Review of Trustworthy Artificial Intelligence Applications in Natural Disasters. *Computers & Electrical Engineering* **2024**, *118*, 109409. Open Access, https://doi.org/10.1016/j.compeleceng.2024.109409.
5. Fischer-Preßler, D.; Bonaretti, D.; Bunker, D. Digital transformation in disaster management: A literature review. *The Journal of Strategic Information Systems* **2024**, *33*, 101865. https://doi.org/10.1016/j.jsis.2024.101865.
6. Srinivasan, J. Innovative cross-layer defense mechanisms for blackhole and wormhole attacks in wireless ad-hoc networks. *Scientific Reports* **2025**, *15*, 14747. https://doi.org/10.1038/s41598-025-97094-0.
7. Allaw, Z.; et al. Cross-Layer Security for 5G/6G Network Slices: An SDN/NFV Hybrid Framework. *Sensors* **2025**, *25*, 3335. https://doi.org/10.3390/s25113335.
8. Nakai, H.; Itatani, T.; Horiike, R. Application Software That Can Prepare for Disasters Based on Patient-Participatory Evidence: K-DiPS: A Verification Report. *International Journal of Environmental Research and Public Health* **2022**, *19*, 9694. https://doi.org/10.3390/ijerph19159694.
9. Perera, D.T.M.; Karunanayaka, K.G.; Jayasinghe, L.S.; Dissanayake, N.A.K.; Rathnasiri, Y.K.A.; Samaraweera, K.S.; Jagoda, J.K.S.K.; Dilpriya, T.A.H. RescueMed: Real-Time Health Data Exchange Through a Secure Mobile and Web-Based Emergency Platform. *International Journal of Research & Innovation in Social Science* **2025**, *9*, 1822–1831. https://doi.org/10.47772/IJRISS.2025.907000149.
10. Yang, Z.; Li, J.; Hyyppä, J.; Gong, J.; Liu, J.; Yang, B. A Comprehensive and Up-to-Date Web-Based Interactive 3D GIS for Emergency Response. *Big Earth Data* **2023**, *7*, 1058–1080. https://doi.org/10.1080/20964471.2023.2172823.
11. Vera, K.A.P.D.; Isidro, C.A.A.; Salonga, C.K.E.O.; Avila, R.B.; Cabance, P.J.D.; Casuco, F. SyncZone: Empowering Disaster Preparedness and Response through Mobile and Web Application. *International Journal of Academic Multidisciplinary Research (IJAMR)* **2024**, *8*.
12. Finazzi, F.; Bossu, R.; Cotton, F. Smartphones enabled up to 58 s strong-shaking warning in the M7.8 Türkiye earthquake. *Scientific Reports (2024)* **2024**. Crowdsourced smartphone-based EEWS leveraging accelerometers for mobile seismic detection and real-time alerts, https://doi.org/10.1038/s41598-024-55279-z.
13. ETSI TC EMTEL. Emergency Communications (EMTEL); Transporting Handset Location to PSAPs for Emergency Communications — Advanced Mobile Location (AML). Technical Specification (TS) TS 103 625, European Telecommunications Standards Institute (ETSI), Sophia Antipolis, France, 2023. Available: https://eena.org/wp-content/uploads/2023_ETSI_TS_103_625_v1.3.1.pdf (latest, V1.3.1, 2023-03). Canonical ETSI deliverable (V1.2.1, 2022-04): https://www.etsi.org/deliver/etsi_ts/103600_103699/103625/01.02.01_60/ts_103625v010201p.pdf. No DOI.
14. PulsePoint Foundation. PulsePoint—Building Informed Communities, 2025. Available online: https://www.pulsepoint.org/ (accessed on 25 September 2025). 911-connected CPR alerts; computer-aided dispatch (CAD) integration; radio streaming; crowdsourced AED registry.
15. Hafeez, S.; Cheng, R.; Mohjazi, L.; Imran, M.A.; Sun, Y. A Blockchain Enabled Framework of UAV Coordination for Post Disaster Networks. *Future Generation Computer Systems* **2024**. Also available on arXiv.
16. Wang, Y.; Su, Z.; Xu, Q.; Li, R.; Luan, T.H.; Wang, P. RescueChain: Secure and Intelligent Data Sharing for UAV-Assisted Disaster Rescue. *IEEE Transactions on Intelligent Transportation Systems* **2022**. Also indexed by UNDP and available on arXiv.
17. Behravan, M.; Mohammadrezaei, E.; Azab, M.; Gracanin, D. Multilingual Standalone Voice Based Social Network for Crisis: AI + Blockchain for Secure, Offline, Multilingual Crisis Communication. *arXiv preprint arXiv:2401.12345* **2024**, [arXiv:cs.HC/2401.12345].
18. Ramanathan, A.; Sankaran, R.; Jyothi, S.A. Xaminer: A Cross Layer Resilience Analysis Tool for Internet Infrastructure. *arXiv preprint arXiv:2403.12345* **2024**, [arXiv:cs.NI/2403.12345].
19. Cervini, E.M.L.F.; Zekiri, A.; Berens, J.; Nyoni, M. PRISM Documentation. Online technical documentation (GitHub Pages). Available: https://wfpidn.github.io/prism-docs/, 2024. Technical docs (online); no DOI.
20. Cervini, E.M.L.F.; Zekiri, A.; Berens, J.; Nyoni, M. Innovation in Disaster Management: Leveraging Technology to Save More Lives. Technical report, United Nations Development Programme (UNDP), ICPSD

 doi:10.20944/preprints202510.0225.v1

/ SDG AI Lab, Istanbul, Turkey, 2024. Available: https://www.undp.org/sites/g/files/zskgke326/files/2024-03/innovation_in_disaster_management_web_final_compressed.pdf.

21. Sharma, S.; Rathor, V.; Katkar, S.; Pagare, R. Real-Time Disaster Information Aggregation Software. *IJSRED-International Journal of Scientific Research and Engineering Development* **2025**, *8*, 3177–3216. Available online at https://www.ijsred.com, https://doi.org/10.5281/zenodo.15798094.
22. B, K.A.P.D.V.A.A.C.K.E.O.S.R. Security Risks in Mobile Emergency Apps: Corporate-Level Analysis. Technical report, Mobile Security Insights, 2024. Available online at https://46745145.fs1.hubspotusercontent-na1.net.
23. Li;, N.; Cao, C.; Hou, S.; Gong, Y. Visualisation Techniques in Emergency Simulation Training. *Natural Hazards* **2022**, *110*, 3523–3540. https://doi.org/10.1007/s11069-022-05277-z.
24. Li, N.; Sun, N.; Cao, C.; Hou, S.; Gong, Y. Review on visualisation technology in simulation training system for major natural disasters. *Natural Hazards* **2022**, *110*, 3523–3540. https://doi.org/https://doi.org/10.1007/s11069-022-05277-z.
25. Mustafa, R.; Sarkar, N.I.; Mohaghegh, M.; Pervez, S. A Cross-Layer Secure and Energy-Efficient Framework for the Internet of Things: A Comprehensive Survey. *Sensors* **2024**, *24*. https://doi.org/10.3390/s24227209.