SafeGuard: Design and Develop A Cross-Platform Emergency Response and Assistance System using GPS and Artificial Intelligence

Green University

Team Information

Md. Abdur Rahim Sarkar	221902011
Sarwar Hosen	221902292
Sumaiya Akther Chowdhury	221902379
Fahmida Akter Nupur	221902385

Abstract

SafeGuard is a smart emergency response system that helps people during critical situations like natural disasters.Using GPS and artificial intelligence, it provides live location tracking, sends timely alerts, and offers step-by-step solutions during emergencies.The system also includes an SOS distress feature and helps connect rescue teamsand volunteers for better coordination. SafeGuard aims to make emergency response faster and more effective, ensuring safety for everyone.

Introduction

SafeGuard is a real-time emergency response system designed to assist during natural disasters like floods, storms, and earthquakes. The Android app uses GPS tracking and Google Maps to provide accurate live location sharing, enabling faster rescue efforts. Users can send distress signals, communicate with emergency responders via a live chat feature, and receive timely updates or instructions. SafeGuard aims to reduce delays, enhance resource coordination, and save lives by bridging the gap between users and rescue teams during critical emergencies.

Problem Domain

- This year, the southern part of our country met with a sudden flood and this disrupted lives, damaged property, and delayed aid due to inefficient systems. [1].
- The current System has a Lack of coordination, which leads to redundant or inefficient resource allocation, so AI can suggest some efficient direction.[2]
- Lack of live GPS tracking hinders identifying affected areas, delaying timely assistance.[3]

Motivation

- A real-time response system can help minimize casualties through early warnings and live location updates.
- Most current solutions focus on one aspect of emergencies (e.g., SOS signals or alerts) rather than combining live tracking, AI-driven solutions, and volunteer coordination.
- Effective communication between rescue teams, volunteers, and victims ensures a coordinated and timely response.
- Users need fast SOS signals and SafeGuard offers a user-friendly solution to alert authorities and emergency contacts.

Objectives

- To learn develop a robust and user-friendly Android app for emergency response
- To learn integrate GPS functionality and Google Maps API to accurately locations.
- To learn about integrating AI in disaster management
- To learn communication technologies like SOS requests.
- To learn implement a database to store and manage user data, emergency alerts, and response information.

Literature Review

Table 1: Table of Literature Review for Our Proposed System

Author	Contribution	Limitations
Kafi et al. [3]	Established GPS technology applications across various stages of disaster management	Dependence on stable communication networks, which can fail during disasters.
Ahmad et al. [4]	It provides real-world examples, showcasing AI's applications across various domains.	Challenges include data privacy concerns, system integration complexities, and ethical issues.
S. Abdalzaher et al. [5]	Surveys the integration of IoT and cloud infrastructure in early earthquake detection, focusing on improving disaster response	Challenges include the need for efficient and reliable communication networks, sustainability issues.

SCAN ME

Methodology

Fig 1 showing request handling from web and mobile apps, ntegration with AI, cloud services, and real-time updates for efficient management and communication.

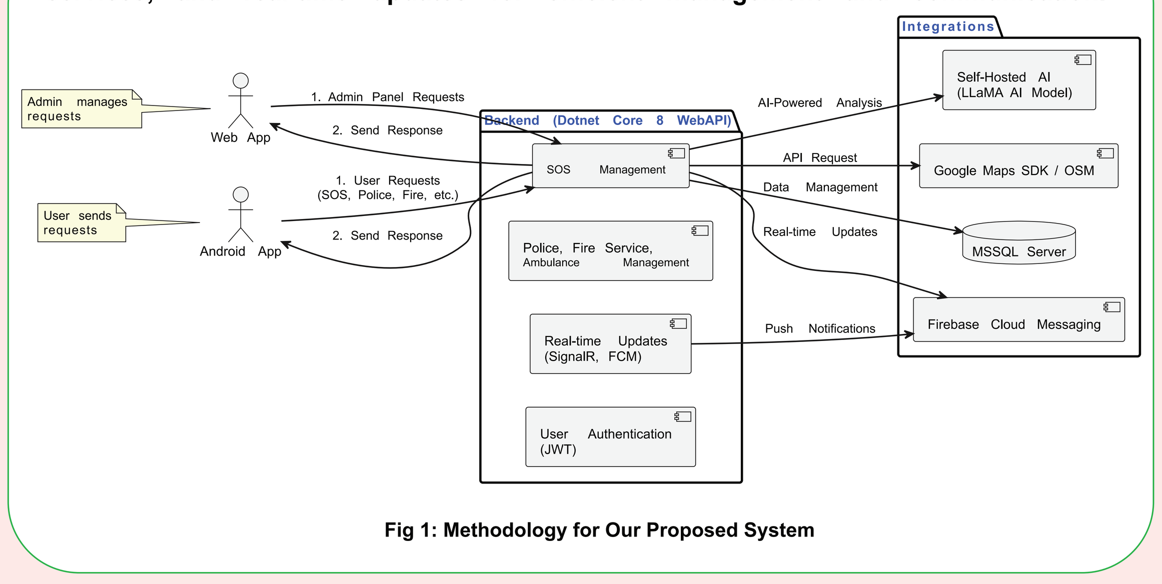

Fig 1: Methodology for Our Proposed System

Software Requirement Specification

Table 2: Table of Software Requirement Specification for Our Proposed System

Functional Requirement	Non-Functional Requirement
SOS Signal: Send SOS signals with one click and share location.	**Performance:** Minimize delays in processing SOS signals and location updates.
Real-Time Disaster Alerts: Send alerts for disasters via push notifications.	**Reliability:** Consistent performance under low-network or stressful conditions.
System Security: Encrypt sensitive data for safety.	**Security:** Protect user data through encryption and authorized access.
AI-Driven Emergency Recommendations: Generate AI-based advice for safety.	**Availability:** Ensure 24/7 availability with minimal downtime and backup.
Live Location Tracking: Track and display real-time GPS location on a map.	**Response Time:** Process SOS signals and messages with minimal delay.
Sensor-Based Monitoring: Integrate IoT sensors for real-time environmental data.	**Usability:** Simple, user-friendly interface for emergency situations.

SDLC Model

Table 3: Table of Software Development Lyfe Cycle Models Comparison

Priority	Criteria	Waterfall Model	Iterative Model	Spiral Model	V-Model	Agile Model	Prototype Model
4	Efficiency	Yes	Yes	**Yes**	Yes	Yes	No
4	User Testing Ability	No	Yes	**Yes**	Yes	Yes	Yes
1	Time-consuming	Yes	Yes	**Yes**	Yes	No	No
2	Technological Knowledge	No	Yes	**Yes**	Yes	No	Yes
2	Well-Known Requirement	Yes	No	**Yes**	Yes	Yes	No
6	Iterative Development	No	Yes	**Yes**	No	Yes	Yes
2	Risk Management	No	No	**Yes**	No	No	No
6	Customer Feedback Integration	No	Yes	**Yes**	No	Yes	Yes
4	Early Testing	No	Yes	**Yes**	Yes	Yes	No
6	Flexibility for Changes	No	Yes	**Yes**	No	Yes	Yes
1	Extensive Documentation	Yes	No	**Yes**	Yes	No	No
5	Supports Complex Requirements	No	Yes	**Yes**	Yes	Yes	No
3	Parallel Testing and Development	No	Yes	**Yes**	Yes	No	No
6	Low Cost of Changes	No	Yes	**No**	No	Yes	Yes
4	Continuous Improvement	No	Yes	**Yes**	No	Yes	No
Total		8	51	**56**	26	50	38

Data Flow Diagram

Fig 2 & 3 shows interactions between users, admins, third-party systems, and detailed processes like registration, management, notifications, & database integration.

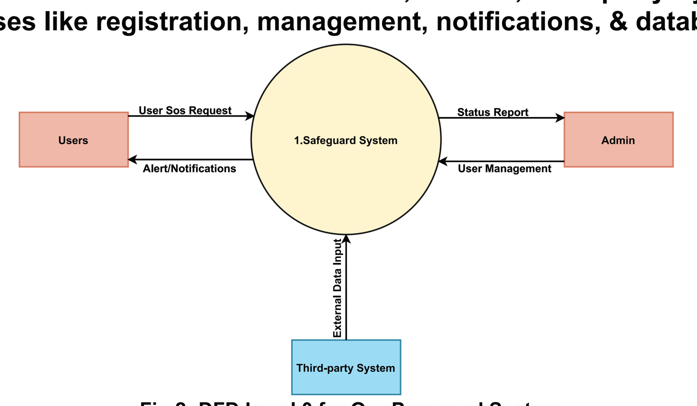

Fig 2: DFD Level 0 for Our Proposed System

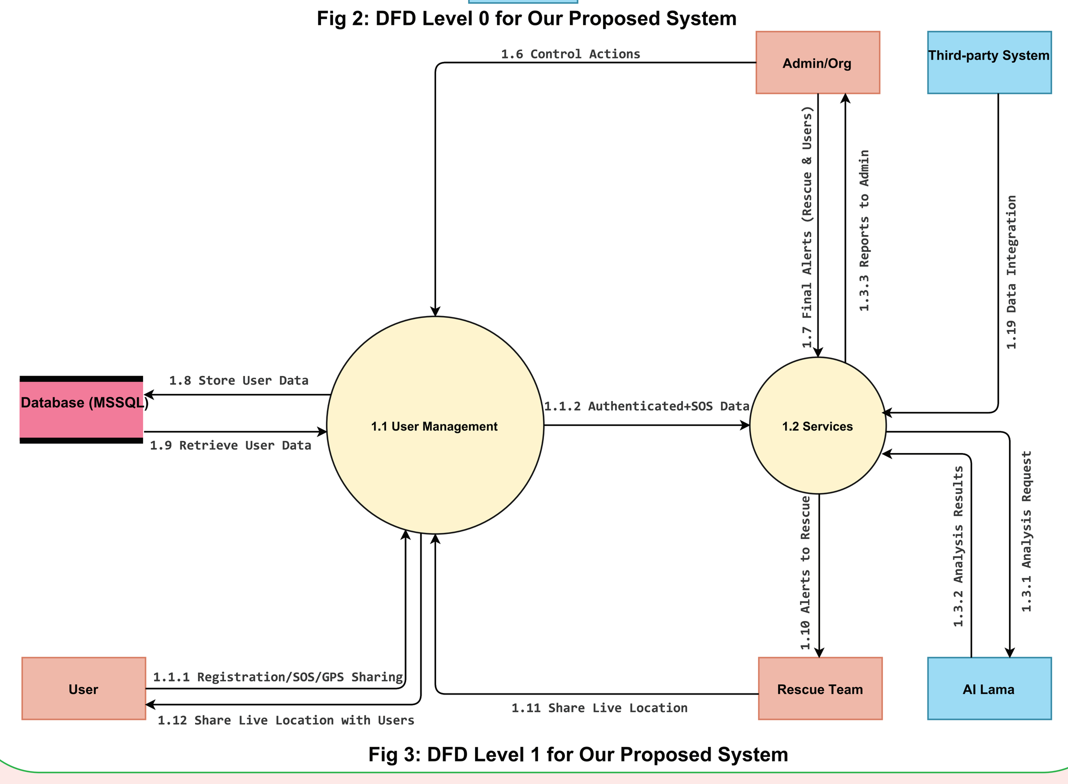

Fig 3: DFD Level 1 for Our Proposed System

UML Use Case Diagram

Fig 4 show system functions and interactions for our proposed system

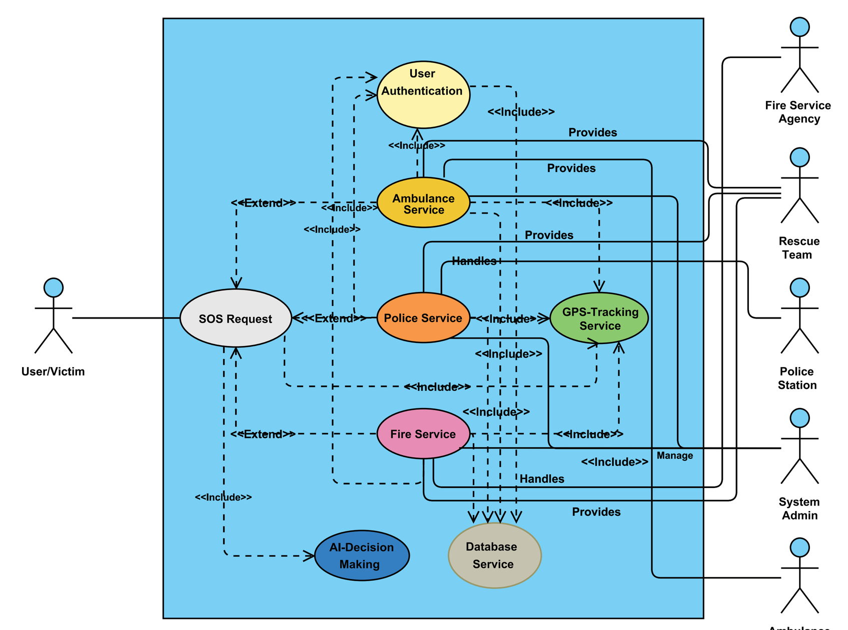

Fig 4: UML Use Case Diagram for Our Proposed System

UML Sequence Diagram

Fig 5 Showing SOS request handling, status updates, and GPS location sharing between the Actor, Web/Android App, and Server.

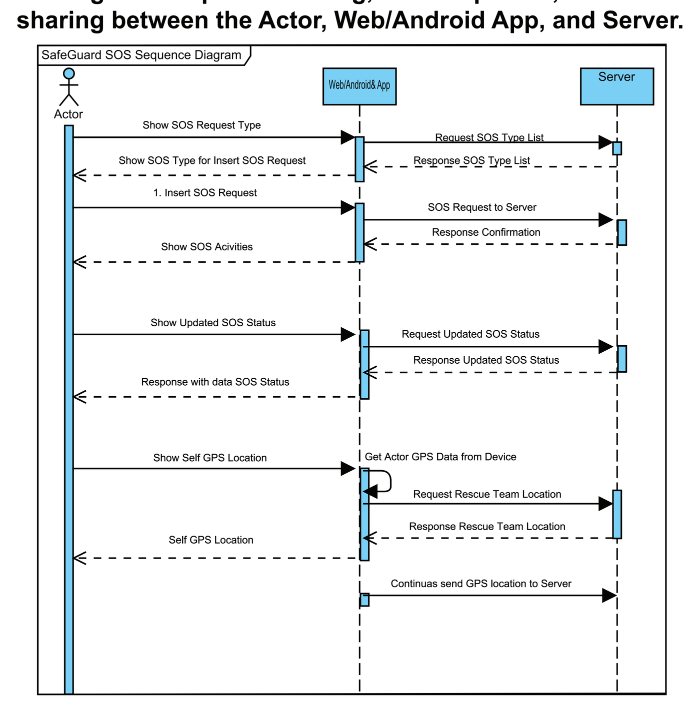

Fig 5: UML Sequence Diagram for Our Proposed System

UML Class Diagram

Fig 6: Class Diagram depicting classes, attributes, methods, and relationships for managing SOS requests, users, volunteers, and emergency responders.

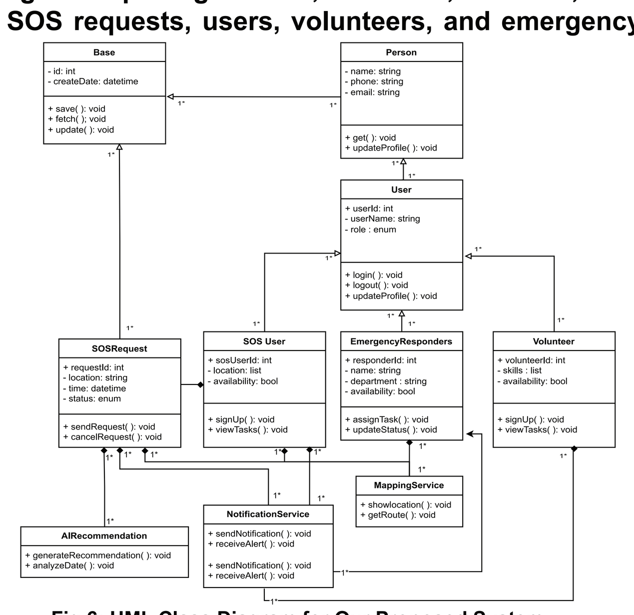

Fig 6: UML Class Diagram for Our Proposed System

Social Impact

- **Saving Lives:** Sends real-time alerts for quick action during disasters like floods and earthquakes.
- **Improved Rescue Operations:** Connects users with rescue teams and volunteers for faster assistance.
- **Disaster Awareness:** Educates the public on safety and disaster preparedness.
- **Reaching Remote Areas:** Offers timely aid to rural and inaccessible regions.
- **Building Community Strength:** Promotes global teamwork among volunteers and responders.
- **Environmental Focus:** Encourages sustainable disaster recovery efforts.

Conclusion

SafeGuard is a game-changing solution designed to revolutionize disaster management.By using real-time GPS, AI, and efficient communication, it helps save lives, support vulnerable groups, and coordinate rescue efforts.SafeGuard empowers communities to act swiftly and effectively. Its global scalability and focus on disaster preparedness make it a vital tool for a safer, more resilient world.

References

[1] The Daily Star. Flash floods wreak havoc in 12 dists, December 2024. Available at: https://www.thedailystar.net/environment/climate-crisis/natural-disaster/news/flash-floods-wreak-havoc-12-dists-3684126 (Accessed: 2024-12-14).

[2] Wenjuan Sun, Paolo Bocchini, and Brian D Davison. Applications of artificial intelligence for disaster management. Natural Hazards, 103(3):2631–2689, 2020.

[3] Kamil Muhammad Kafi and Mohamed Barakat A. Gibril. "gps application in disaster management: A review". Asian Journal of Applied Sciences, 4(1), 2016.

[4] Zeeshan Ahmad. How ai is transforming emergency response systems. TGI, July 2024.

[5] Derya Yiltas-Kaplan Imed Ben Dhaou Wilfried Yves Hamilton Adoni Mohamed S. Abdalzaher, Moez Krichen. Integration of iot and cloud infrastructure for early earthquake detection. Journal Name, 2024. Accessed: 2024-12-17.