



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

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 teams and 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
Kamil Muhammad Kafi and Mohamed Barakat A. Gibril [3]	Established GPS technology applications across various stages of disaster management.	Dependence on stable communication networks, which can fail during disasters.
Zeeshan Ahmad [4]	It provides real-world examples, showcasing AI's applications across various domains.	Challenges include data privacy concerns, system integration complexities, and ethical issues.
Mohamed S. Abdalzaher, Moez Krichen, Derya Yiltas-Kaplan, Imed Ben Dhaou, Wilfried Yves Hamilton Adoni [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

Block Diagram

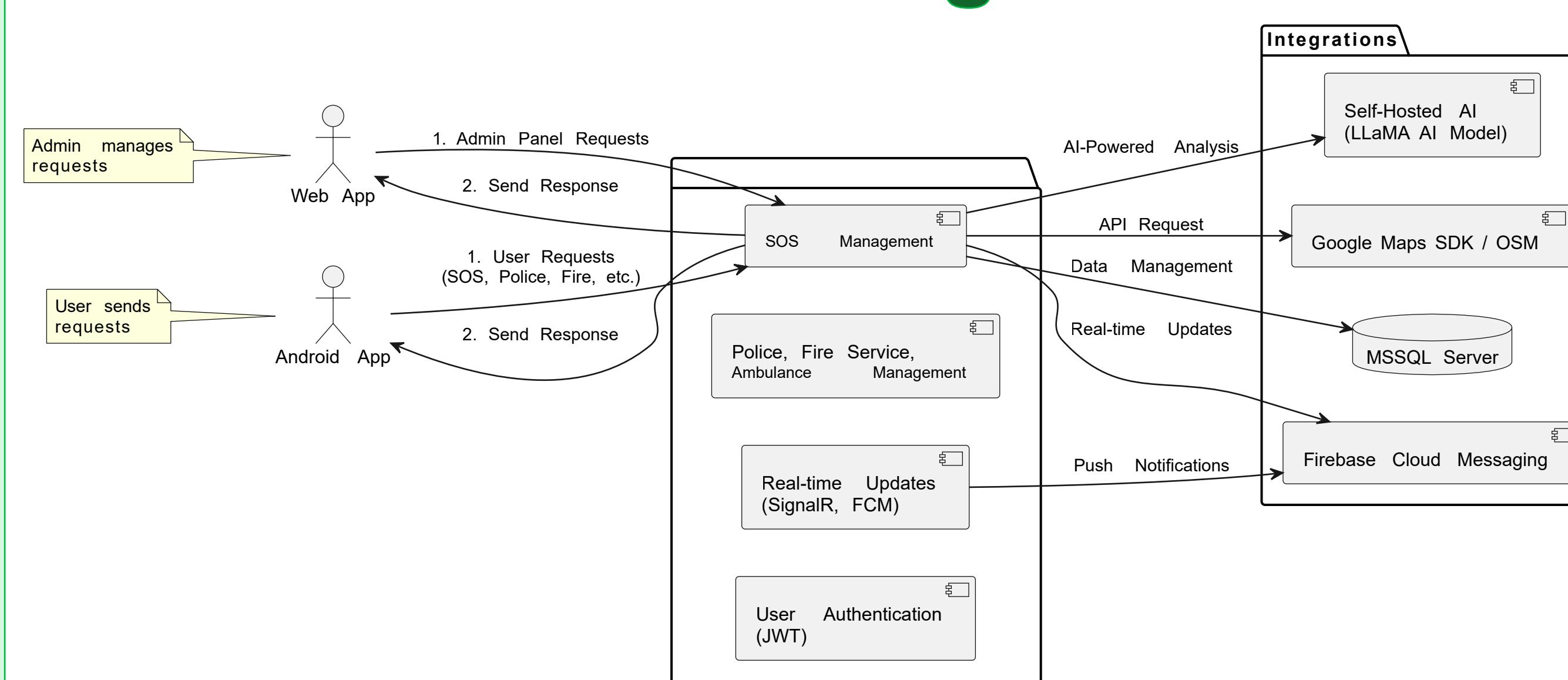


Fig 1: Block Diagram for Our Proposed System

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 Life 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

DFD Diagram

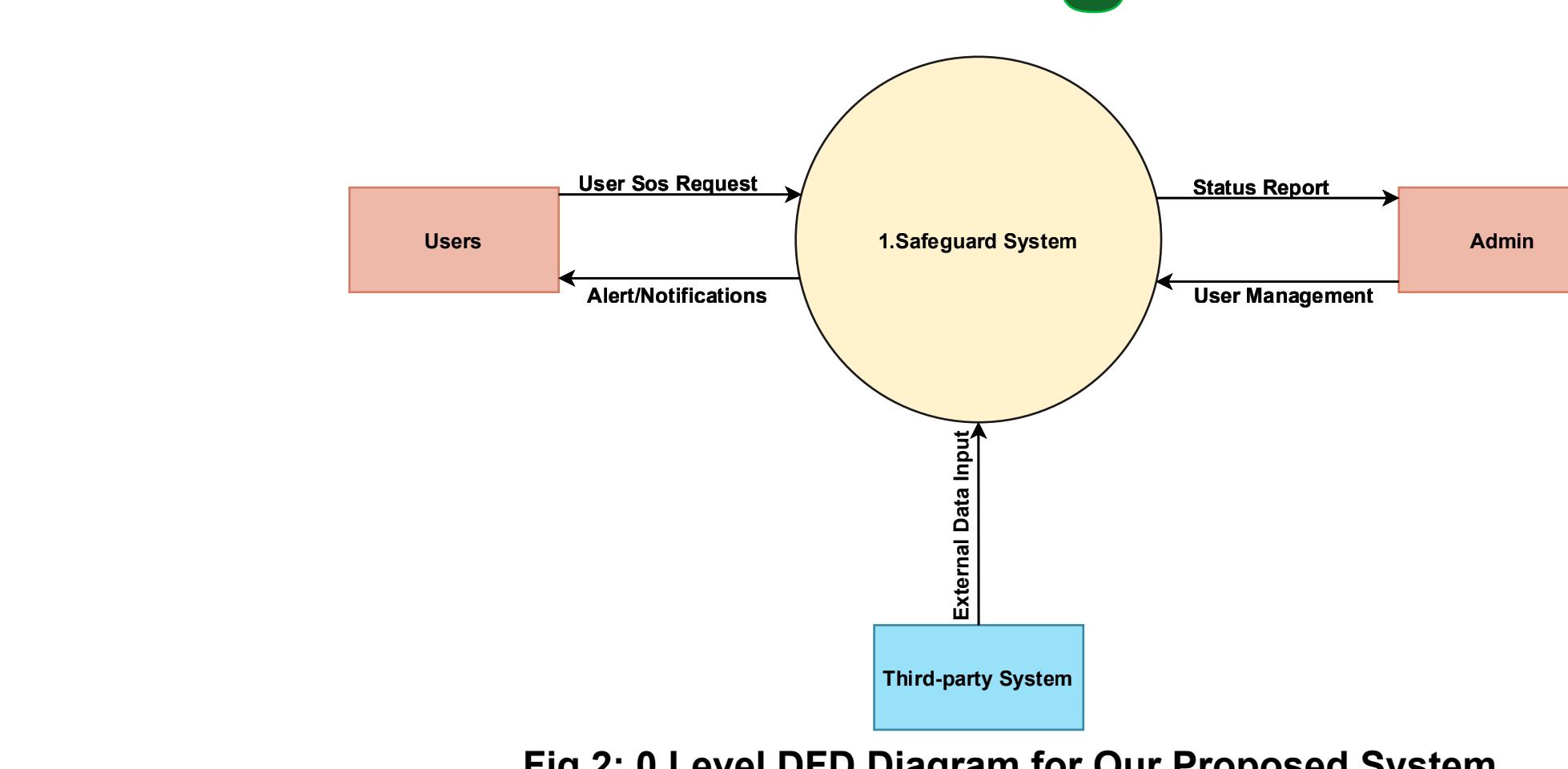


Fig 2: 0 Level DFD Diagram for Our Proposed System

