

Software Requirements Specification

Disaster Alert System (DAS)

Software Engineering (CS-304) – Assignment 1
Group 17

Prepared by

Group 17

Contents

1	Introduction	3
1.1	Purpose of the Document	3
1.2	Scope of the System	3
2	Overall System Description	3
3	Functional Requirements	4
3.1	User Interface and Authentication Module	4
3.2	Alert Management Module	4
3.3	Location and Region Processing Module	5
3.4	Notification Delivery Module	5
3.5	Logging and Audit Module	5
4	Non-Functional Requirements	5
4.1	Usability	5
4.2	Reliability	6
4.3	Performance	6
5	Assumptions and Constraints	6
6	Requirements Validation Strategy	6
7	Conclusion	7

1 Introduction

1.1 Purpose of the Document

This document defines the functional and non-functional requirements for the **Disaster Alert System (DAS)**. It serves as a formal reference for developers, testers, and quality assurance personnel to guide system design, implementation, and validation.

The requirements described herein are written to be precise, testable, and implementation-independent, enabling systematic verification through automated testing at later stages of development.

1.2 Scope of the System

The Disaster Alert System is a web-based platform designed to detect, manage, and disseminate disaster alerts related to earthquakes, tsunamis, floods, and cyclones. Authorized personnel can create and manage alerts, while registered users receive location-relevant notifications through supported communication channels.

The system emphasizes:

- Correctness of alert generation
- Timely and reliable notification delivery
- Robust behavior under failure and high-load conditions
- Safe handling of invalid or boundary inputs

2 Overall System Description

The Disaster Alert System is composed of modular components, each responsible for a specific area of functionality:

- User Interface Module
- Authentication and Authorization Module
- Alert Management Module
- Location and Region Processing Module
- Notification Delivery Module
- Logging and Audit Module

The system primarily exposes RESTful APIs and is designed to support both backend and frontend clients. External services such as SMS gateways and geocoding providers may be integrated or simulated depending on deployment context.

3 Functional Requirements

3.1 User Interface and Authentication Module

- The system shall allow users to register using name, email address, phone number, and location details.
- The system shall prevent duplicate user registration using the same email address.
- The system shall allow registered users to authenticate using secure credentials and obtain an authorization token.
- The system shall support role-based access control for distinguishing authorized alert creators from regular users.
- The system shall allow authenticated users to update their profile and location information.

3.2 Alert Management Module

- The system shall allow authorized users to create disaster alerts with the following attributes:
 - Disaster type
 - Severity level
 - Description and title
 - Affected location or region
 - Optional geographic coordinates
- The system shall validate all mandatory alert fields before creation.
- The system shall classify alert severity according to predefined severity levels ranging from LOW to CATASTROPHIC.
- The system shall assign each newly created alert an initial state of *Active*.
- The system shall prevent duplicate alerts for the same region within a configurable temporal and spatial window.
- The system shall restrict alert modification and cancellation to authorized personnel only.
- The system shall support alert acknowledgment and status transitions (e.g., Active, Updated, Cancelled, Expired).

3.3 Location and Region Processing Module

- The system shall associate users and alerts using geographic coordinates derived from location data.
- The system shall validate latitude and longitude values to ensure they fall within valid geographic bounds.
- The system shall identify users affected by an alert based on spatial proximity and region matching.
- The system shall ensure that users outside the affected region do not receive alert notifications.
- The system shall support fallback mechanisms when external geocoding services are unavailable.

3.4 Notification Delivery Module

- The system shall deliver alerts to affected users via messaging notifications.
- The system shall optionally support email-based notifications.
- The system shall support retry mechanisms for notification delivery in case of transient failures.
- The system shall suppress repeated notifications for unchanged alerts.
- The system shall support multi-channel notification delivery when enabled.

3.5 Logging and Audit Module

- The system shall log alert creation, modification, and cancellation events.
- The system shall record timestamps and user identifiers for all critical operations.
- The system shall preserve historical alert records for auditing and review.

4 Non-Functional Requirements

4.1 Usability

- The system shall expose consistent and predictable API responses.
- Error responses shall be descriptive and suitable for consumption by frontend applications.

4.2 Reliability

- The system shall remain operational during partial failures of external services.
- The system shall degrade gracefully when notification or database services are unavailable.
- The system shall preserve alert data during failure scenarios.

4.3 Performance

- The system shall efficiently handle concurrent alert creation and notification delivery.
- The system shall support burst alert scenarios typical of large-scale disaster events.

5 Assumptions and Constraints

- The system is developed for academic and simulation purposes.
- External services such as SMS, email, and geocoding may be mocked.
- Internet connectivity is assumed during normal operation.
- Disaster sensor data may be simulated.

6 Requirements Validation Strategy

Each requirement defined in this document shall be validated through one or more of the following approaches:

- Unit Testing
- Boundary Value Analysis
- Integration Testing
- Risk-Based and Failure-Mode Testing

Automated testing frameworks shall be used wherever feasible to ensure repeatable and objective validation.

7 Conclusion

This Software Requirements Specification provides a clear, forward-looking, and testable definition of the Disaster Alert System. The requirements establish a foundation for systematic development, verification, and evaluation of a mission-critical alerting platform.

All subsequent design, implementation, and testing activities shall adhere to the requirements defined in this document.