Software Requirements Specification

Disaster Alert System (DAS)

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

Prepared by

Group-17

January 24, 2026

Contents

1 1. Introduction

1.1 1.1 Purpose of the Document

This document defines the functional and non-functional requirements for the **Disaster Alert System (DAS)**. It serves as an authoritative reference for developers, testers, and quality assurance personnel to ensure the system is implemented and validated according to agreed specifications.

Each requirement in this document is explicitly validated through automated test cases implemented using `pytest`. Test identifiers (e.g., FT-001, BVA-001, IT-001) are referenced to ensure full traceability between requirements and verification artifacts.

1.2 1.2 Scope of the System

The Disaster Alert System is a web-based backend application designed to create, manage, and disseminate disaster alerts to users located in affected geographic regions. Authorized personnel can generate alerts, while registered users receive notifications relevant to their location.

The system emphasizes correctness, reliability, authorization control, and boundary-safe input handling.

2 2. Overall System Description

The Disaster Alert System is to be composed of the following logical modules:

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

Although the system exposes REST APIs, its behavior is validated independently of any frontend through automated backend tests.

3 3. Functional Requirements

3.1 3.1 User Interface Module

- The system shall allow users to register using name, email, phone number, and location information. **Validated by: FT-002, IT-001**

- The system shall allow authenticated users to log in and obtain a valid authorization token. **Validated by: FT-002, IT-002**
- The system shall allow authenticated users to update their profile and location details. **Validated by: FT-004**
- The system shall allow users to retrieve alerts relevant to their location. **Validated by: IT-001, IT-003**

3.2 3.2 Alert Management Module

- The system shall allow authorized users to create disaster alerts with title, description, type, severity, location, and optional coordinates. **Validated by: FT-001, FT-003**
- The system shall validate mandatory alert fields before creation. **Validated by: BVA-001, BVA-002, BVA-006**
- The system shall accept valid boundary inputs such as long titles without failure. **Validated by: BVA-003**
- The system shall restrict alert modification to the alert creator only. **Validated by: AUTH-001**
- The system shall assign newly created alerts an initial state of *Active*. **Validated by: FT-001**

3.3 3.3 Location and Region Processing Module

- The system shall associate alerts and users using geographic coordinates. **Validated by: FT-003**
- The system shall ensure that coordinate values lie within valid geographic bounds. **Validated by: BVA-004**
- The system shall ensure that users outside an affected region do not receive alerts. **Validated by: IT-003**

3.4 3.4 Notification Delivery Module

- The system shall support delivery of alerts via messaging notifications. **Validated by: IT-001**
- The system shall optionally support email notifications. **Validated by: FT-003**

- The system shall retry notification delivery on transient failure. **Validated by: FT-001**
- The system shall suppress repeated notifications for unchanged alerts. **Validated by: FT-001**

3.5 3.5 Logging and Audit Module

- The system shall record alert creation events with timestamps and user identifiers. **Validated by: FT-001, IT-001**
- The system shall preserve historical alert records for retrieval. **Validated by: IT-001**

4 4. Non-Functional Requirements

4.1 4.1 Usability

- The system shall expose consistent and predictable API responses. **Validated by: All FT and IT tests**
- Error messages shall be returned for invalid input conditions. **Validated by: BVA-001, BVA-002, BVA-006**

4.2 4.2 Reliability

- The system shall handle external service failures gracefully using mocks during testing. **Validated by: FT-001**
- The system shall maintain correct behavior under repeated alert creation attempts. **Validated by: FT-001**

4.3 4.3 Performance

- The system shall respond to API requests within acceptable time limits for simulated workloads. **Validated by: IT-001, IT-003**
- The system shall support concurrent users within the scope of academic simulation. **Validated by: IT-001**

5 5. Assumptions and Constraints

- The system is developed for academic evaluation purposes.

- External services such as SMS and Email gateways are mocked.
- Internet connectivity is assumed.
- Real disaster feeds are simulated.

6 6. Requirements Validation Strategy

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

- **Functional Testing** (FT-001 to FT-004)
- **Boundary Value Analysis** (BVA-001 to BVA-006)
- **Authorization Testing** (AUTH-001)
- **Integration Testing** (IT-001 to IT-003)

All tests are automated and executed using the `pytest` framework.

7 7. Conclusion

This Software Requirements Specification provides a clear, testable, and traceable definition of the Disaster Alert System. All requirements are directly mapped to implemented automated tests, ensuring verifiability, consistency, and alignment with system behavior.

Status: REQUIREMENTS VERIFIED AND TESTABLE