

Project Title

S.H.I.E.L.D – Smart Hazard Identification & Emergency Live Dispatch

1. Introduction

1.1 Purpose

The purpose of this document is to specify the functional and non-functional requirements of S.H.I.E.L.D, an AI-assisted public safety incident reporting and alerting system. This SRS serves as the primary reference for system design, development, testing, and evaluation.

1.2 Scope

S.H.I.E.L.D is a distributed software system that enables citizens to report public safety incidents in real time using geo-tagged inputs and optional media evidence. The system uses AI techniques to classify incidents, estimate severity, detect hotspots, and trigger alerts through a centralized dashboard.

1.3 Definitions, Acronyms, Abbreviations

Term	Meaning
AI	Artificial Intelligence
NLP	Natural Language Processing
GPS	Global Positioning System
Incident	Any reported safety-related event
Severity Score	Numerical value indicating urgency
Hotspot	Area with high incident density

1.4 Intended Audience

- Project evaluator / faculty guide
- Software engineering students
- Developers implementing the system

2. Overall Description

2.1 Product Perspective

S.H.I.E.L.D is a standalone web-based system built using a service-oriented architecture. It integrates frontend interfaces, backend services, AI/ML components, and geospatial databases.

The system does not depend on external emergency infrastructure and simulates dispatch behaviour for demonstration.

2.2 Product Functions (High-Level)

- Citizen authentication and reporting
- AI-based incident classification
- Severity estimation
- Geo-visualization of incidents
- Hotspot detection
- Alert generation and dispatch
- Administrative monitoring dashboard

2.3 User Classes

User	Description
Citizen	Reports incidents and views public alerts
Admin	Monitors incidents, alerts, and analytics

2.4 Operating Environment

- Web browser (desktop/mobile)
- Backend hosted on cloud platform
- Database with geospatial support
- AI models served as backend services

2.5 Design and Implementation Constraints

- Solo development
- 3-month development window

- Pretrained ML models only
- No real integration with police or hospitals

2.6 Assumptions and Dependencies

- Users provide honest reports
- Internet connectivity is available
- Map APIs are accessible
- AI models provide probabilistic outputs

3. System Features & Functional Requirements

3.1 User Authentication & Trust Management

Description:

The system shall authenticate users and maintain a basic trust score to discourage misuse.

Functional Requirements:

- FR1: The system shall allow users to register using email.
- FR2: The system shall authenticate users before allowing incident submission.

3.2 Incident Reporting Module

Description:

Users can submit incident reports with location, description, and optional media.

Functional Requirements:

- FR3: The system shall allow users to select an incident category.
- FR4: The system shall store all incident data in the database.

3.3 NLP-based Incident Classification

Description:

Textual descriptions are analyzed using NLP to identify incident type.

Functional Requirements:

- FR5: The system shall analyze incident text using an NLP model.
- FR6: The system shall assign a predicted category with confidence score.

3.4 Severity Scoring Engine

Description:

The system computes a severity score for each incident.

Functional Requirements:

- FR 7: The system shall compute severity on a scale of 0–100.
- FR8: The system shall consider AI outputs, location context, and report density.
- FR 9: The system shall update severity dynamically if new data arrives.

3.5 Geo-Visualization & Hotspot Detection

Description:

Incidents are visualized on a live map with hotspot detection.

Functional Requirements:

- FR10: The system shall display incidents on a map.
- FR11: The system shall generate heatmaps based on severity.
- FR12: The system shall detect hotspots using clustering algorithms.

3.6 Alert & Dispatch Module

Description:

Alerts are triggered when severity thresholds are exceeded.

Functional Requirements:

- FR13: The system shall generate alerts for high-severity incidents.
- FR14: The system shall notify admins through the dashboard.

3.8 Admin Dashboard & Analytics

Description:

Admins can monitor system activity and trends.

Functional Requirements:

- FR15: The system shall provide real-time incident statistics.
- FR16: The system shall display historical trends.

4. External Interface Requirements

4.1 User Interfaces

- Responsive web interface
- Map-based visualization
- Dashboard panels for analytics

4.2 Software Interfaces

- Map APIs (Leaflet)
- ML model inference libraries
- Database interfaces

5. Non-Functional Requirements

5.1 Performance

- NFR1: Incident submission response time < 3 seconds

5.2 Reliability

- NFR4: No single point of failure in core workflow

5.3 Security

- NFR5: Authentication required for submissions

5.4 Scalability

- NFR7: System shall support increasing number of users
- NFR8: AI services shall be independently scalable

5.5 Usability

- NFR9: Interface shall be intuitive and minimal
- NFR10: Incident reporting shall require ≤ 1 minute

6. Architecture Diagram

