Software Requirements Specification

for

NGO-Driven Community Issue Reporting Platform with AI

Version 1.0 approved

Prepared by Mohnishkumar.S(24BIT0211)

Sriraam.S.V(24BIT0232)

Harshad.S(24BIT0224)

Prepared on 20/01/26

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Mohnishkumar.S	20/01/26	Initial SRS creation	*1.0*

1. Introduction

1.1 Purpose

This document provides a detailed description of the Software Requirements Specification (SRS) for the Web-Based Community Issue Reporting & Action Platform. It serves as a reference for system developers, testers, faculty evaluators, and stakeholders.

1.2 Document Conventions

This document follows the IEEE Software Requirements Specification (SRS) standards.Functional requirements are uniquely identified using the notation REQ-x.Non-functional requirements are identified using the notation REQ-NFR -x.

1.3 Intended Audience and Reading Suggestions

This document is intended for software developers involved in the design and implementation of the system. It is also useful for testers, NGO coordinators, volunteers, and system administrators responsible for system operation and maintenance. Faculty members and academic evaluators may use this document for review, validation, and assessment purposes.

1.4 Product Scope

The platform enables citizens to report civic issues through a centralized web application, while Non-Governmental Organizations (NGOs) are responsible for verifying reported issues and coordinating resolution activities. Volunteers and partner organizations execute assigned tasks, supported by AI-assisted issue classification and prioritization. The system promotes efficient collaboration, transparency, and timely resolution of community issues.

1.5 References

This Software Requirements Specification refers to the IEEE 830 Software Requirements Specification guidelines as the primary standard for structuring and documenting requirements. Additional references include standard Software Engineering course materials and institutional guidelines provided by the Department of Information Technology. These references collectively serve as the foundation for defining system requirements, use cases, and overall project scope.

2.Overall Description

2.1 Product Perspective

The Web-Based NGO-Driven Community Issue Reporting & Action Platform is a new, self-contained software product developed to address community-level civic issues through a centralized digital system. It is not a replacement or extension of any existing system but is designed as an independent web application. The platform integrates multiple components, including user interaction modules, AI-assisted issue analysis, mapping services, and notification services. External interfaces include third-party mapping APIs and communication services, which support location visualization and real-time notifications.

2.2 Product Functions

At a high level, the system provides the following major functions:

- *User registration and authentication*
- *Reporting of civic issues with images and location details*
- *AI-based issue classification, severity assessment, and prioritization*
- *Verification of reported issues by NGO coordinators*
- *Assignment of issues to volunteers or partner organizations*
- *Tracking of issue resolution status and progress updates*
- *Notification of stakeholders regarding status changes*
- *Generation of transparency and impact reports*

2.3 User Classes and Characteristics

The system supports multiple user classes with varying levels of access and technical expertise. Citizens are primary users who report issues and track their resolution, typically requiring a simple and user-friendly interface. NGO Coordinators possess administrative privileges to verify, prioritize, and assign issues and are expected to have moderate technical proficiency. Volunteers interact with the system to view assignments and update task progress and generally require mobile-friendly access. Partner Organizations assist in resolving complex issues and use the system for coordination and reporting. Donors access the system primarily to view impact and transparency reports. System Administrators manage user roles, security, and system configuration and are expected to have advanced technical expertise. Among these, citizens and NGO coordinators are the most critical user classes for successful system operation.

2.4 Operating Environment

The system will operate in a web-based environment and be accessible through modern web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. It will be hosted on cloud-based servers and accessed using internet-enabled devices, including desktops, laptops, and smart phones. The system may integrate with external services such as mapping APIs, AI image processing services, and email or SMS notification gateways.

2.5 Design and Implementation Constraints

The system must comply with data privacy and security regulations applicable to user-generated content. Development is constrained to web technologies and must ensure browser compatibility and responsive design. Integration with third-party APIs such as mapping and notification services may impose usage limits or dependency constraints. Secure communication protocols must be followed, and role-based access control should be implemented. The system design should follow standard coding conventions to ensure maintainability.

2.6 User Documentation

User documentation for the system will include a basic user manual, online help content, and step-by-step usage instructions for different user roles. The documentation will be provided in digital format and may include screenshots and simple tutorials to assist non-technical users.

2.7 Assumptions and Dependencies

It is assumed that users have access to internet-enabled devices with basic web browsing capabilities and, where applicable, GPS functionality. The system depends on the availability and reliability of third-party services such as mapping APIs, AI image analysis modules, and notification gateways. Any changes in these external services, development tools, or operating environments may affect system functionality. It is also assumed that NGOs and volunteers actively participate in updating issue status to ensure accurate tracking.

3. External Interface Requirements

3.1 User Interfaces

The system shall provide a web-based user interface designed for ease of use and accessibility across different user roles. The interface will follow standard web design guidelines with a consistent layout, navigation menu, and common controls such as buttons for submission, confirmation, and cancellation. Key screens include user registration and login pages, issue reporting forms, map-based issue visualization, role-specific dashboards, and status tracking views. Error messages and system notifications will be displayed in a clear and user-friendly manner. Detailed user interface designs and screen layouts will be documented separately in a user interface specification.

3.2 Hardware Interfaces

The system shall provide a web-based user interface designed for ease of use and accessibility across different user roles. The interface will follow standard web design guidelines with a consistent layout, navigation menu, and common controls such as buttons for submission, confirmation, and cancellation. Key screens include user registration and login pages, issue reporting forms, map-based issue visualization, role-specific dashboards, and status tracking views. Error messages and system notifications will be displayed in a clear and user-friendly manner. Detailed user interface designs and screen layouts will be documented separately in a user interface specification.

3.3 Software Interfaces

The system interfaces with several external software components to provide its functionality. These include a mapping service API for location visualization, an AI-based image analysis service for issue classification and severity assessment, and a notification service for sending email or SMS alerts. The system also interacts with a backend database for storing user information, issue details, and status updates. Data exchanged between the system and external software components includes images, location coordinates, classification results, and notification messages. All software interactions will be handled through secure and well-defined application programming interfaces (APIs).

3.4 Communications Interfaces

The system requires network communication to support web access and external service integration. Communication between clients and servers will use the HTTP or HTTPS protocol, with HTTPS employed for secure data transmission. The platform will support email and SMS communication for notifications related to issue status updates and task assignments. Standard web communication protocols and encryption mechanisms will be used to ensure data integrity, confidentiality, and secure synchronization between system components.

4. System Features

4.1 User Registration and Authentication

4.1.1 Description and Priority

This feature allows users to register, log in, and log out of the platform, enabling role-based access to system functionalities. Priority: High

4.1.2 Stimulus/Response Sequences

- *Stimulus: Visitor navigates to the registration page and submits registration details.*
- *Response: System validates input, creates user account, and sends confirmation email.*
- *Stimulus: Registered user enters login credentials.*
- *Response: System authenticates credentials and directs user to the role-specific dashboard.*
- *Stimulus: User clicks logout.*
- *Response: System terminates the session and redirects to the homepage.*

4.1.3 Functional Requirements

- *REQ-1: System shall allow users to register by providing required information (name, email, password, role).*
- *REQ-2: System shall validate email format and password strength during registration.*
- *REQ-3: System shall send a confirmation email upon successful registration.*
- *REQ-4: System shall allow users to log in using registered credentials.*
- *REQ-5: System shall restrict access to role-specific features based on user type.*
- *REQ-6: System shall allow users to log out securely and terminate their session.*
- *REQ-7: System shall handle invalid login attempts with appropriate error messages.*

4.2 Issue Reporting and Submission

4.2.1 Description and Priority

Enables citizens to report civic issues by providing descriptions, uploading images, and sharing location details. Priority: High

4.2.2 Stimulus/Response Sequences

- *Stimulus: Citizen selects "Report Issue" on the dashboard.*
- *Response: System displays the issue reporting form.*
- *Stimulus: Citizen fills in details, uploads images, and submits the form.*
- *Response: System validates input, saves issue details, and sends acknowledgment notification.*

4.2.3 Functional Requirements

- *REQ-8: System shall provide a form for reporting civic issues including fields for description, location, and image upload.*
- *REQ-9: System shall validate all required fields before submission.*
- *REQ-10: System shall store issue reports in the database with timestamp and reporter details.*
- *REQ-11: System shall send a confirmation notification to the reporting citizen.*
- *REQ-12: System shall allow citizens to edit or withdraw submitted issues before verification*

4.3 AI-Based Issue Analysis

4.3.1 Description and Priority

Uses AI to classify the type of civic issue, assess severity, detect duplicates, and suggest priority levels for action. Priority: High

4.3.2 Stimulus/Response Sequences

- *Stimulus: Citizen submits an issue with image attachment.*
- *Response: AI module analyzes image, classifies issue type, determines severity, detects duplicates, and stores suggested priority.*
- *Stimulus: NGO coordinator reviews issue.*
- *Response: System displays AI-generated classification and severity to assist in verification.*

4.3.3 Functional Requirements

- *REQ-13: System shall send uploaded issue images to the AI module for analysis.*
- *REQ-14: System shall classify issue types automatically (e.g., pothole, waste, damaged streetlight).*
- *REQ-15: System shall assess severity and urgency of reported issues.*
- *REQ-16: System shall detect and flag duplicate or potentially fake reports.*
- *REQ-17: System shall provide AI-suggested priority levels to the NGO coordinator.*

4.4 Issue Verification and Assignment

4.4.1 Description and Priority

Allows NGO coordinators to review, verify, prioritize, and assign reported issues to volunteers or partner organizations. Priority: High

4.4.2 Stimulus/Response Sequences

- *Stimulus: NGO coordinator views newly reported issues.*
- *Response: System displays AI analysis results and issue details.*
- *Stimulus: Coordinator verifies issue authenticity and sets priority.*
- *Response: System updates issue status and enables assignment to volunteers or partners.*

4.4.3 Functional Requirements

- *REQ-18**:** System shall allow coordinators to approve or reject reported issues.*
- *REQ-19**:** System shall allow coordinators to set priority levels.*
- *REQ-20**:** System shall enable assignment of verified issues to volunteers or partner organizations based on criteria such as location, availability, and skill.*
- *REQ-21**:** System shall notify assigned volunteers or partners of new tasks.*

4.5 Task Execution and Tracking

4.5.1 Description and Priority

Volunteers and partners can view, accept, update, and complete assigned tasks. Citizens and coordinators can track resolution progress. Priority: High

4.5.2 Stimulus/Response Sequences

- *Stimulus**:** Volunteer views assigned tasks.*
- *Response**:** System displays task details with deadline and location.*
- *Stimulus**:** Volunteer updates task progress.*
- *Response**:** System records updates and notifies coordinator and citizen.*

4.5.3 Functional Requirements

- *REQ-22**:** System shall allow volunteers to accept or decline assigned tasks.*
- *REQ-23**:** System shall allow volunteers to update task progress and upload completion proof.*
- *REQ-24**:** System shall notify coordinators and citizens when task status changes.*
- *REQ-25**:** System shall allow citizens to view the current status and timeline of reported issues.*

4.6 Notification and Communication

4.6.1 Description and Priority

Manages all system-generated notifications to users about issue updates,assignments, and completion. Priority: Medium

4.6.2 Stimulus/Response Sequences

- *Stimulus**:** Issue status changes (submitted, verified, assigned, completed).*
- *Response**:** System sends email, SMS, or push notification to relevant users.*

4.6.3 Functional Requirements

- *REQ-26**:** System shall send notifications for status changes to citizens, volunteers, and coordinators.*
- *REQ-27**:** System shall notify users of task assignments and completions.*
- *REQ-28**:** System shall provide configurable notification preferences for users.*

5.Other Nonfunctional Requirements

5.1 Performance Requirements

- *REQ-NFR-1: The system shall support up to 1000 simultaneous users without performance degradation.*
- *REQ-NFR-2: Issue submission response time shall not exceed 2 seconds under normal network conditions.*
- *REQ-NFR-3: AI-based image analysis shall complete classification within 5 seconds per image.*
- *REQ-NFR-4: Map loading and visualization shall render within 3 seconds for up to 500 active issue markers.*
- *REQ-NFR-5: Database queries for issue tracking and reporting shall return results within 2 seconds for datasets of up to 10,000 records.*

5.2 Safety Requirements

- *REQ-NFR-6: The system shall implement secure storage, regular backups, and transactional updates to prevent data loss or corruption.*
- *REQ-NFR-7: Unauthorized access to user data, issue reports, or task assignments shall be prevented.*
- *REQ-NFR-8: Input validation shall prevent harmful or malicious data submissions.*
- *REQ-NFR-9: The system shall use HTTPS for secure communication of all sensitive data.*
- *REQ-NFR-10: The system shall comply with applicable data protection and privacy regulations.*

5.3 Security Requirements

- *REQ-NFR-11: The system shall require strong authentication for all registered users, including email verification and password policies.*
- *REQ-NFR-12: Role-based access control shall restrict system functionality according to user roles.*
- *REQ-NFR-13: Sensitive user data shall be encrypted at rest and in transit.*
- *REQ-NFR-14: The system shall implement protections against SQL injection, XSS, and CSRF attacks.*
- *REQ-NFR-15: Security audits shall be performed regularly to maintain compliance with best practices.*

5.4 Software Quality Attributes

- *REQ-NFR-16: The system shall have a target uptime of 99.5% per month.*
- *REQ-NFR-17: The code shall be modular and maintainable with complete developer documentation.*
- *REQ-NFR-18: The system shall be portable across modern web browsers and responsive for desktop and mobile devices.*
- *REQ-NFR-19: The system shall provide a clear, intuitive, and accessible user interface.*
- *REQ-NFR-20: The system shall implement reliable error handling and rollback mechanisms for failed operations.*
- *REQ-NFR-21: The system shall allow extensibility for future AI modules or third-party integrations.*

5.5 Business Rules

- *REQ-NFR-22: Only registered citizens shall be allowed to report civic issues.*
- *REQ-NFR-23: Only NGO coordinators shall be able to verify, approve, or assign tasks.*
- *REQ-NFR-24: Volunteers may accept and update assigned tasks but cannot modify verified issue details.*
- *REQ-NFR-25: Partner organizations may update task completion status but require coordinator approval for major changes.*
- *REQ-NFR-26: Donors may view transparency and impact reports but cannot modify system data.*
- *REQ-NFR-27: AI-suggested classifications and priorities shall assist coordinators, but final verification decisions remain with human users.*

6. Other Requirements

This section specifies additional requirements that are not covered in previous sections of the SRS but are essential for the successful implementation and operation of the system.

6.1 Database Requirements

- *REQ-OR-1: The system shall use a centralized database to store user profiles, issue reports, task assignments, and status updates.*
- *REQ-OR-2: Each issue record shall include issue ID, description, image references, location coordinates, reporter details, status, and timestamps.*
- *REQ-OR-3: The database shall support backup and recovery mechanisms to prevent data loss.*
- *REQ-OR-4: The system shall ensure data consistency and integrity during concurrent access and updates.*

6.2 Legal and Regulatory Requirements

- *REQ-OR-5: The system shall comply with applicable data protection and privacy laws governing user-generated content.*
- *REQ-OR-6: User consent shall be obtained before collecting or processing personal and location data.*
- *REQ-OR-7: Uploaded images and content shall be subject to moderation to prevent misuse or violation of legal policies.*

6.3 Reusability and Extensibility Requirements

- *REQ-OR-8: The system shall be designed using modular components to allow reuse in future civic or NGO-related projects.*
- *REQ-OR-9: The system shall allow future integration of additional AI modules, reporting tools, or external service providers without major redesign.*

6.4 Data Retention Requirements

- *REQ-OR-10: The system shall retain issue and task data for a minimum period defined by NGO policy.*
- *REQ-OR-11: Users shall be allowed to request deletion of their personal data, subject to regulatory constraints.*

Appendix A: Glossary

AI
Artificial Intelligence used for image analysis, classification, and prioritization of reported issues.

Citizen
A registered user who reports civic issues and tracks their resolution.

NGO
Non-Governmental Organization responsible for verifying issues and coordinating resolution activities.

Issue
A reported civic problem such as potholes, waste accumulation, or damaged infrastructure.

Volunteer
A registered individual who performs assigned on-ground tasks.

Partner Organization
An external service provider assisting the NGO in resolving issues.

Severity
A measure of the seriousness or urgency of a reported issue.

Priority
The order in which verified issues are addressed based on severity and impact.

Appendix B: Analysis Models

The following analysis models are used to represent the system behavior and structure:

- *Use Case Diagram showing interactions between actors and the system*
- *Data Flow Diagram representing the flow of issue data through the system*
- *Entity Relationship Diagram illustrating relationships between users, issues, tasks, and reports*

- *State Transition Diagram depicting issue status changes from reporting to resolution*

These models assist in understanding system requirements and guiding design and implementation.

Appendix C: To Be Determined List

1. *TBD-1: Final selection of AI image analysis algorithm and model architecture.*
2. *TBD-2: Exact performance thresholds for large-scale deployments.*
3. *TBD-3: Notification service provider (Email/SMS/Push).*
4. *TBD-4: Data retention duration based on NGO operational policies.*
5. *TBD-5: Legal compliance certification requirements, if any.*