Software Requirements Specification
for

Aapda Sahayak

Prepared by Abhinav, Gaurish, Snehasish

5th February 2024

Table of Contents

1. Introduction

1.1 Purpose

To build a portal, where all rescue agencies can register their information, including their location, contact details, and areas of expertise. This information could be entered manually by agency administrators, or automated using GPS or other location-tracking technologies. It would include a map that shows the locations of all registered rescue agencies, along with filters that allow users to narrow down the results based on specific criteria, such as the type of disaster, the resources available, or the time since the last reported activity. In addition to displaying the locations of rescue agencies, the portal would also include features for communication and collaboration.

1.2 Document Conventions

In this document wherever we refer to the term “Portal” it is to be noted that it refers to our project “Aapda Sahayak”

1.3 Intended Audience

Rescue Agencies: Organizations providing disaster response and relief services.
Emergency Responders: Individuals actively involved in emergency response efforts.
Government Authorities: Agencies managing disaster response at different levels.
General Public: Accessible features for transparency and awareness during disasters.

1.4 Product Scope

Scope: The portal will provide a comprehensive solution for rescue agencies to register, manage, and share critical information during disasters. It will include location tracking, communication, and collaboration features, ultimately improving the collective response to emergencies.

Benefits:
1. Enhanced Coordination: Real-time information sharing fosters a more efficient and coordinated response among rescue agencies.

2. Resource Optimization: Collaboration prevents duplication, optimizing the utilization of available resources like medical equipment and transportation.

3. Quick Response: Swift location and communication with nearby agencies significantly reduce emergency response time.

4. Improved Situational Awareness: The map-based interface enhances decision-making by providing a visual representation of the disaster area and registered agency locations.

Objectives:
1. Develop a centralized database for rescue agencies to register and manage their information.

2. Implementing location tracking technologies to automate the entry of agency information.

3. Designing an intuitive user interface with a map display showing the locations of all registered rescue agencies.

4. Providing filtering options for users to narrow down results based on specific criteria such as disaster type, available resources, and recent activity.

2. Overall Description

2.1 Product Perspective

The Aapda Sahayak portal is a standalone solution designed to enhance coordination among rescue agencies during disasters. It does not replace existing systems but interfaces with them to create a seamless disaster management ecosystem. Major subsystems include user management, agency registration, mapping, filtering, and communication. External interfaces include geospatial services, communication channels, and existing systems. The portal ensures efficient collaboration and information exchange for effective disaster response.

2.2 Product Functions

User registration, centralized database with GPS-based location tracking, map integration, communication, and collaboration.

2.3 User Classes and Characteristics

Scalability:
Capable of handling increased user activity during emergencies.

Real-time Tracking:
Provides up-to-date location information for timely responses.

User-Friendly Interface:
Intuitive design for easy navigation and efficient use by various stakeholders.

Security Protocols:
Implements robust measures to safeguard sensitive information.

Interoperability:
Easily integrates with mapping services, communication tools, and security solutions.

Customization:
Allows agencies to input and update information based on their specific expertise and resources.

Communication Tools:
Facilitates seamless communication and collaboration among agencies.

2.4 Operating Environment

Hardware Platform: Standard web hosting infrastructure.

Operating System: Compatible with Windows Server and Linux distributions.

Web Server: Node, Express.

Database Management System: MongoDB.

Browser Compatibility: Compatible with major browsers: Chrome, Firefox, and Edge.
External Services: Interfaces with geospatial services and communication channels.

Network Requirements: Requires a stable internet connection.

Software Dependencies: Relies on React and APIs for geospatial and communication services.

2.5 Design and Implementation Constraints

1. Security and Privacy Considerations:

The portal must comply with relevant data protection regulations, necessitating robust security measures to ensure authorized access to the database and protection of sensitive information.

2. Regulatory Policies:

Compliance with regulatory policies related to disaster response and data management will guide the development process.

2.6 User Documentation

README file: to be created after the completion of the portal.

2.7 Assumptions and Dependencies

Assumptions:

1) All the emergency agencies will be registered on the portal.
2) The transit of agencies as shown on the map, is strictly based upon manual changes. (no usage of any hardware device to track the vehicle)

Dependencies:

1) External APIs for maps and communication interface to be used.
2) MongoDB database server.
3) Information on different agencies available on the internet.

3. External Interface Requirements

3.1 User Interfaces

A simple portal kind of interface with a dashboard or map to show different agencies or to locate distress calls. A sidebar/Header with options to communicate, request for assistance, offer to assist, etc.

3.2 Software Interfaces

1. Database Interface:

The portal interacts with a central database to store and retrieve information about registered rescue agencies. The chosen Database Management System (DBMS) is MongoDB

Data Items Inbound:

❖ Agency registration details, including location, contact information, and areas of expertise.
❖ Automated data from GPS or other location-tracking technologies.

Data Items Outbound:

❖ Filtered agency information for display on the user interface.
❖ Collaborative alerts and resource-sharing information.

2. Mapping and Visualization Interface:

The portal utilizes mapping and visualization components to display the locations of registered rescue agencies. The mapping service selected Google Maps API

Data Items Inbound:

❖ Geographical coordinates of registered agencies.
❖ Filtering criteria for specific disaster types, resources, or activity time.

Data Items Outbound:

❖ Visual representation of agency locations on the map.
❖ User interface updates based on applied filters.

3. Communication and Collaboration Services:

The portal incorporates communication and collaboration features to facilitate interaction between rescue agencies.

Data Items Inbound:

❖ Alerts and requests for assistance from other agencies.
❖ Shared resource information such as medical equipment or transportation availability.

Data Items Outbound:

- ❖ Alerts and assistance requests sent to other agencies.
- ❖ Updates on shared resource availability

3.3 Communications Interfaces

- Utilizes Socket.io, a WebSocket library, for low-latency, bidirectional communication, enabling real-time tracking of agency locations.
- Implements Socket.io for real-time chat communication, facilitating instant messaging and alert notifications between rescue agencies and emergency responders.
- Leverages RESTful APIs over HTTPS for secure data exchange, ensuring seamless communication between the application and external APIs for technology integration.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

Response Time:

- ❖ Requirement: The portal will respond to user interactions (e.g., filtering, map navigation) within a few seconds under normal operating conditions.
- ❖ Rationale: Prompt response time ensures a smooth user experience, enabling quick access to critical information during emergencies.

Database Query Performance:

- ❖ Requirement: Database queries for retrieving agency information will execute within a few seconds.
- ❖ Rationale: Fast database query performance ensures timely access to updated agency data, facilitating effective decision-making during emergencies.

Map Rendering Speed:

- ❖ Requirement: The map display will render agency locations and updates within a short period.
- ❖ Rationale: Rapid map rendering speed allows users to visualize the current situation and plan response strategies efficiently.

Scalability:

- ❖ Requirement: The portal will support a minimum of 1000 simultaneous users without significant degradation in performance.
- ❖ Rationale: Scalability ensures that the application can handle increased user traffic during large-scale emergencies without compromising performance.

4.2 **Safety Requirements**

Data Security:

- Requirement: Implement robust encryption measures to protect sensitive user and agency information.
- Safeguard: Regularly update encryption protocols and conduct security audits to identify and address vulnerabilities.

User Authentication:

- Requirement: Ensure secure authentication processes to prevent unauthorized access.
- Safeguard: Implement multi-factor authentication to add an extra layer of security.

Location Privacy:

- Requirement: Protect the privacy of agency locations, allowing only authorized entities to access this information.
- Safeguard: Implement access controls and encryption for location data.

External API Integration:

- Requirement: Ensure secure communication with external APIs to prevent data breaches.
- Safeguard: Validate and authenticate external API requests, and regularly update integration protocols.

4.3 **Security Requirements**

User Identity Authentication:

- Requirement: Implement a secure user authentication system to ensure only authorized individuals have access to the application.

- Implementation: Utilize strong password policies, multi-factor authentication (MFA), and OAuth 2.0 for secure authorization.

Access Controls:

- Requirement: Define role-based access controls to restrict access to specific features and data based on user roles.
- Implementation: Implement fine-grained access controls to ensure that users only have access to the information necessary for their roles.

4.4 Software Quality Attributes

Usability:

- ❖ Requirement: The application interface should be intuitive and easy to navigate, achieving a minimum score of 4 out of 5 in user satisfaction surveys.
- ❖ Rationale: High usability ensures that rescue agencies can quickly access and utilize the application's features during emergencies, reducing response time.

Security:

- ❖ Requirement: The application should comply with industry-standard security protocols (e.g., HTTPS, encryption) to protect sensitive data.
- ❖ Rationale: Robust security measures are essential for safeguarding confidential information, such as personal contact details, from unauthorized access or breaches.

Maintainability:

- ❖ Requirement: The Code will follow established coding standards and be well-documented
- ❖ Rationale: A maintainable codebase simplifies future updates and enhancements, ensuring the long-term viability of the application for ongoing rescue operations.

Interoperability:

- ❖ Requirement: The portal will support interoperability with external systems or APIs commonly used by rescue agencies (e.g., emergency alert systems).
- ❖ Rationale: Interoperability enables seamless integration with existing emergency response infrastructure, enhancing collaboration and coordination among rescue agencies.

4.5 Business Rules

Administrator (Admin):

- ❖ Functions:
    - ➢ Manage user registration and access.
    - ➢ Monitor and manage the overall system.
    - ➢ Conduct security audits and implement security measures.
    - ➢ Configure and customize system settings.

- ❖ Circumstances:
    - ➢ Admins have full access to all functions at all times.

Rescue Agency Administrator:

- ❖ Functions:
    - ➢ Register and manage agency information.
    - ➢ Update real-time location and resources.
    - ➢ Communicate and collaborate with other agencies.
    - ➢ Access relevant mapping and location-based tools.
- ❖ Circumstances:
    - ➢ Full access to agency-related functions.

Government Authority:

- ❖ Functions:
    - ➢ Monitor and coordinate rescue agency efforts.
    - ➢ Access comprehensive data on disaster response.
    - ➢ Communicate with and guide agencies.
- ❖ Circumstances:
    - ➢ Access to monitoring and coordination functions.