

SOFTWARE REQUIREMENTS SPECIFICATION (SRS DOCUMENT)

DISASTER RELIEF MANAGEMENT SYSTEM

1. Introduction

1.1 Purpose

The DRMS is designed to be a comprehensive and collaborative platform to streamline and enhance the coordination of disaster relief efforts. The system will serve as a centralized information hub, connecting disaster victims and government agencies to facilitate efficient reporting, verification, and allocation of contributions.

1.2 Scope

The DRMS will cover the entire disaster relief lifecycle, from victim reporting to the allocation of contributions by government agencies and relief organizations. It will provide distinct features tailored for victims and government agencies, ensuring a seamless and efficient disaster response.

1.3 Overview

The DRMS will enable disaster victims to log in and document their losses comprehensively, detailing the extent of damages to property, personal belongings, and livelihoods. Using a user-friendly interface, victims can provide real-time updates on their needs and access relevant resources. The DBMS will play a pivotal role in securely storing and managing this information, ensuring data accuracy and integrity.

Government agencies and relief organizations will have access to a centralized dashboard where they can review and verify reported losses. Upon validation, the system will facilitate the seamless allocation of appropriate contributions, whether financial aid, resources, or services. The DBMS will support efficient data retrieval and reporting functionalities to streamline the decision-making process for relief efforts.

2. Overall Description

2.1 Product Perspective

1. Lifecycle Model

The DRMS will follow an incremental model of development, allowing for iterative enhancements and additions to the system. The incremental process model is also known as the Successive version model. First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.

The incremental model is well-suited for projects like the Disaster Relief Management System (DRMS) for several reasons:

a) Evolution of Requirements:

- In disaster relief scenarios, requirements can evolve rapidly as the situation on the ground changes. The incremental model allows for flexibility and accommodates changing requirements in subsequent iterations.

b) Continuous Feedback:

- The incremental model involves regular cycles of development and testing. This ensures that stakeholders, including victims and government agencies, can provide feedback on the system's functionality and make adjustments as needed.

c) Progressive Development:

- Disaster relief systems often have urgent and critical needs. The incremental model allows for the delivery of a functioning system in increments, ensuring that at each iteration, there is a tangible product with enhanced features and capabilities.

d) Risk Management:

- Incremental development helps in early identification and mitigation of risks. As each iteration is a mini-project, potential issues can be addressed before they escalate, reducing overall project risk.

e) Adaptability to Changing Circumstances:

- Disasters are unpredictable, and the requirements for a relief system may change rapidly based on the severity and nature of the disaster. The incremental model allows for the adaptation of the system to changing circumstances.

f) User Involvement:

- Incremental models encourage frequent interaction with end-users. In the case of DRMS, victims and government agencies can actively participate in the development process, ensuring that the system meets their specific needs.

g) Early Deployment of Critical Features:

- The incremental model allows for the early deployment of critical features. In a disaster relief scenario, certain functionalities, such as victim reporting or resource allocation, may be more urgently needed. These can be prioritized and deployed in early iterations.

2. Programming Languages

It will be a web-based application built on a scalable and secure database management system. The programming language selected for development for front end is html, css and javascript. For backend Django is chosen and MySQL is used as database chosen for its versatility and suitability for web applications.

2.2 User Classes and Characteristics

1. Victims:

- Victims shall be able to register and log in to the system to report losses.
- They can provide detailed information about damages to property, personal belongings, and livelihoods.
- Victims can update their needs and access relevant resources through the syst

2. Government Agencies:

- Government agencies shall monitor and coordinate relief efforts through the system.
 - They have the authority to approve and allocate contributions based on verified losses.
 - Agencies can access comprehensive reports for decision-making and future disaster planning.
- Various individuals, organisations etc can send contributions intended for the victims to the government which will be shown in the contributions table and can be appropriately used and distributed by the government.

2.3 Operating Environment

The DRMS will be accessible through standard web browsers such as Chrome, Firefox, and Safari. It will be hosted on a secure server with database connectivity. The system will be designed to handle a potentially large user base and concurrent transactions during disaster scenarios.

3. Functional Requirements

3.1 Victim Reporting

- Victims shall be able to register and log in to the system securely.
- Victims can report losses by providing detailed information on affected areas, property damage, and personal losses.

3.2 Contribution Verification

- Government agencies shall have designated accounts with secure authentication.

- The system shall enable government to review and verify reported losses, ensuring authenticity.

3.3 Contribution Allocation

- Verified losses shall trigger the automatic or manual allocation of appropriate contributions by government agencies and relief organizations.
- Contributions may include financial aid, supplies, services, or specific relief efforts.

3.4 Resource Inventory

- Government shall maintain an updated inventory of available resources, including financial aid, supplies, and services.
- The system shall facilitate the easy addition, modification, and deletion of resources in the inventory.

3.5 Contribution Tracking

- Government agencies shall have access to a contribution tracking system.
- The system shall display the status of each contribution, from verification to allocation, ensuring transparency.

3.7 System Administration

- The system shall have an administrative interface for managing user roles, permissions, and system settings.
- Administrators shall be able to monitor system logs for security and audit purposes.

4. Non-functional Requirements

4.1 Performance

- The system shall be designed to support many concurrent users.
- Response times for user interactions, including reporting and verification, shall be minimized.

4.2 Security

- User data, including victim information, contribution details, and verification records, shall be securely stored.
- The system shall implement role-based access control, ensuring that only authorized users can access specific functionalities.

4.3 User friendly

- Users regardless of identity should be able to easily use and understand the features of the software. So, all features should be clear-cut and ordered.

- They should be able to easily navigate through all the sections available.

4.4 Reliability

- The system shall have an uptime of at least 99.9%

- Data backup and recovery mechanisms shall be in place to prevent data loss.

4.5 Maintainability

- The system shall be designed with modular components for ease of maintenance

- Regular maintenance shall be scheduled for updates and enhancements.

4.6 Portability

- The user interface shall be responsive, ensuring accessibility from various devices.

- The system shall be compatible with various OS.

4.7 Reusability

- Code components shall be documented and designed for reusability in future disaster management projects.

- System modules shall be structured for easy integration with other systems if needed.

5. Interface Requirements

5.1 User Interfaces

5.1.1 Victim Interface

- The victim interface shall provide a secure and intuitive login mechanism.
- Victims shall have a dashboard displaying options for reporting losses, updating needs, and accessing resources.
- The reporting interface shall include fields for detailed information, such as affected areas, property damage, and personal losses.
- Resource inventory management shall be accessible through a user-friendly interface, allowing easy addition, modification, and deletion of resources.

5.1.2 Government Agency Interface

- Government agencies shall have a comprehensive dashboard for monitoring and coordinating relief efforts.
- The interface shall include sections for approving and allocating contributions, viewing reports, and managing system settings.

- Government agencies shall have access to a contribution tracking interface for transparent monitoring.

5.1.3 Administrative Interface

- The administrative interface shall provide system administrators with tools for managing user roles, permissions, and system configurations.
- System logs and audit trails shall be accessible through the administrative interface for security monitoring.

6. Constraints

- The development of the DRMS shall adhere to a predefined budget and timeline.
- The system shall be compatible with modern web browsers, and responsive design principles shall be implemented.

7. Data Requirements

7.1 Database Design

7.1.1 Database Overview

The DRMS will utilize a relational DBMS to store and manage data. The database will consist of a set of tables designed to capture and organize the necessary information for effective disaster relief management.

7.1.2 Table Definitions

-Six tables are utilized for storage in DBMS.

User	
Name	Varchar(100)
Phone no	Number(30)
Email	Varchar(255)
Username	Varchar(100)
Password	Varchar(100)

Citizen	
House no	Varchar(10)
House name	Varchar(50)
Address	Varchar(300)
Aadhar no	Number(12)
Ration card no	Number(20)
Account no	Number(20)
IFSC code	Varchar(20)
Description	Varchar(500)
Survey result	Boolean

Staff	
Name	Varchar(100)
Phone no	Number(30)
Email	Varchar(255)
Position	Varchar(100)
StaffID	Number(5)
Password	Varchar(100)

Contributions	
ID	Varchar(20)
Name	Varchar(100)
Organisation name	Varchar(100)
Organisation id	Varchar(30)
Organisation type	Varchar(50)
Aadhar no	Number(12)
Amount	Number(30)

Transfer	
Name	Varchar(100)
Email	Varchar(255)
Phone no	Number(30)
Account no	Number(20)
IFSC code	Varchar(20)
Amount send	Number(30)

Survey	
Survey ID	Number(30)
House no	Number(10)
House name	Varchar(50)
Address	Varchar(300)
Survey_desp	Varchar(500)
Estimated loss	Number(20)

8. Preliminary schedule and budget

-Initially the mini project to be completed within 3 months and no budget.

9. Appendices

- 1.OS: Operating System
- 2.DBMS: Data Base Management System
- 3.DRMS: Disaster Relief Management System

10. Future Enhancements

- Integration with external emergency alert systems for real-time updates.
- Including Geographical area and maps to show affected regions and categorize according to heavily and less affected regions.
- Development of a mobile application to enable victims to report losses on-the-go.

11. Signatures

(To be signed by relevant stakeholders, including project managers, developers, and representatives from government agencies and relief organizations, to indicate agreement with the requirements specified in this document.)