

American International University-Bangladesh (AIUB)  
**Department of Computer Science  
Faculty of Science & Technology (FST)**

**AUTOMATED FLOOD WARNING AND RAPID RELIEF SYSTEM**

A Software Requirement Engineering Project Submitted

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Semester: Spring\_24\_25** | | **Section:F** | **Group Number:5** | |
| SN | Student Name | Student ID | Contribution in % (100%) | Individual Marks |
| 12 | SIAMIR RAHMAN SIAM | 21-44620-1 | 100% |  |
| 23 | RIFAT BIN AMAN | 21-45475-3 | 100% |  |
| 26 | TANVIR MUSTABI SIFAT | 21-45870-3 | 100% |  |
| 33 | TAUSHIF SHARIAR | 22-46405-1 | 100% |  |

The project will be Evaluated for the following Course Outcomes

|  |  |  |
| --- | --- | --- |
| **EVALUATION CRITERIA** | Total Marks (50) | |
| 40 | |
| Introduction [Section 1.1-1.4], Format, Submission, Defense | [10 Marks] |  |
| System Overall Description & Functional Requirements | [10 Marks] |  |
| System Quality Attributes and Project Requirements | [10 Marks] |  |
| UML and E-R Diagram with Data Dictionary | [10 Marks] |  |
| UI/UX Prototyping | [10 Marks] |  |

Software Requirements Specification

for

< AUTOMATED FLOOD WARNING AND RAPID RELIEF SYSTEM >

Version 2.0 approved

Prepared by

< GROUP-5 >

<organization>

< JUNE 11,2025>

Table of Contents

[Revision History 3](#_Toc201043639)

[1. Introduction 4](#_Toc201043640)

[1.1 Purpose 4](#_Toc201043641)

[1.2 Document Conventions 5](#_Toc201043642)

[1.3 Intended Audience and Reading Suggestions 6](#_Toc201043643)

[1.4 References 7](#_Toc201043644)

[2. Overall Description 8](#_Toc201043645)

[2.1 Product Perspective 8](#_Toc201043646)

[2.2 Product Functions 9](#_Toc201043647)

[2.2.1 Account 9](#_Toc201043648)

[2.3 User Classes and Characteristics 12](#_Toc201043649)

[2.4 Hardware and Operating Environment 14](#_Toc201043650)

[2.5 Design and Implementation Constraints 16](#_Toc201043651)

[2.6 User Documentation 17](#_Toc201043652)

[3. System Requirements 18](#_Toc201043653)

[3.1 System Features 18](#_Toc201043654)

[3.1.1 Account 18](#_Toc201043655)

[3.1.2.1 Account Login 18](#_Toc201043656)

[3.1.3.1 Search Account / View Profile 19](#_Toc201043657)

[3.1.4.1 Update Account Information 19](#_Toc201043658)

[3.1.5.1 Log Out 19](#_Toc201043659)

[3.1.6.1 Flood Forecasting 19](#_Toc201043660)

[3.1.7.1 Alert Notification System 20](#_Toc201043661)

[3.1.8.1 Shelter Locator and Navigation 20](#_Toc201043662)

[3.1.9.1 Relief Tracking and Distribution 20](#_Toc201043663)

[3.1.10.1Citizen Reporting & Feedback 20](#_Toc201043664)

[3.1.11.1 Admin Dashboard (BWDB) 21](#_Toc201043665)

[3.1.12.1 Emergency Response Coordination 21](#_Toc201043666)

[3.2 Non-Functional/Quality Requirements 21](#_Toc201043667)

[3.3 Project Requirements 23](#_Toc201043668)

[4. Design and Interface Requirements 26](#_Toc201043669)

[4.1 UML Diagrams 26](#_Toc201043670)

[4.2 Data Dictionary 30](#_Toc201043671)

[4.3 UI/UX Design Specification 30](#_Toc201043672)

[4.3.1 Design Tool 30](#_Toc201043673)

[4.3.2 DESIGNS ARE BELOW 30](#_Toc201043674)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Changes | Version |
| TAUSHIF SHARIAR | 26.05.25 | Edited in 1.1 Purpose | 1.1 |
| TAUSHIF SHARIAR | 29.05.25 | Added some new features | 1.2 |
| SIAMIR RAHMAN SIAM | 11.06.25 | Finalize Features (add/remove some fetures) | 2.0 |

# Introduction

## Purpose

The **Automated Flood Warning and Relief System** is designed to mitigate the impacts of floods by providing real-time flood forecasting and relief coordination. Developed in collaboration with the **Bangladesh Water Development Board (BWDB)** and the **Ministry of Disaster Management and Relief**, the system will utilize real-time data from water monitoring sources and issue alerts to affected areas using various communication channels. The system will ensure timely evacuation, aid distribution, and post-flood recovery by coordinating local disaster management authorities, humanitarian organizations, and emergency services.

**Scope of the Product**

The **Automated Flood Warning and Relief System** is a digital solution created to **predict floods, warn people early**, and **coordinate emergency relief operations** to reduce loss of life and property during flood disasters in Bangladesh.

This system will **use real-time data** from water level sensors, weather forecasts, and satellite imagery to detect when a flood is likely to occur. Once detected, it will **send alerts directly to citizens’ mobile phones** (via SMS and app notifications), giving them enough time to evacuate or prepare.

Citizens will also be able to **locate nearby shelters**, check if there is space, and get directions to reach safety. If one shelter is full, the system will automatically suggest others nearby.

On the administrative side, the system will help **BWDB officials and local disaster management teams** monitor where relief supplies (like food, water, medicine) are going. It will show **which areas need aid**, **what has been sent**, and whether it has reached the right destination—helping prevent corruption or resource hoarding.

Citizens can also use the system to **report issues**, like blocked roads, missing people, or if they didn’t receive relief. These reports will be visible to disaster response teams so they can act quickly.

Additionally, the system will **coordinate efforts among different organizations**—such as NGOs, government teams, and telecom partners—so that everyone works together using the same information. A **dashboard interface** will allow officials to view flood maps, resource allocation, and emergency response status in real time.

**Business Requirements:**

The key business objectives of the system are:

* **Provide Early Flood Warnings**  
  Deliver timely flood alerts using real-time water level and weather data to help people evacuate and prepare.
* **Notify the Public Quickly and Effectively**  
  Send alerts using multiple channels like SMS, app notifications, and voice calls—especially to people in flood-prone areas.
* **Coordinate Relief Distribution**  
  Help disaster response teams and aid organizations send food, water, medicine, and shelter support to the right locations efficiently.
* **Ensure Transparency and Accountability**  
  Track relief operations digitally to prevent fraud, hoarding, or mismanagement of supplies.

This business requirement section sets the **strategic goals** and **operational expectations** of the system before the technical design starts.

## Document Conventions

This Software Requirements Specification (SRS) document adheres to the following standards and conventions:

1. **Standards:**
   * All headings follow a hierarchical numbering system (e.g., 1, 1.1, 1.2) to maintain consistency and clarity.
   * Functional and non-functional requirements are clearly identified and grouped for better readability.
2. **Typography:**
   * **Bold Text:** Used for section headers, key terms, and emphasis on important points.
   * Blue color text are hyperlinks.
3. **Requirements Prioritization:**
   * Each requirement is labeled with a priority level (High, Medium, Low) based on its criticality to the system's functionality.
   * High-level requirements inherit their priority by default unless otherwise specified.
4. **Cross-Referencing:**
   * Requirements are cross-referenced where applicable to ensure traceability and consistency.
5. **Document Navigation:**
   * A table of contents will be included to help readers quickly locate sections of interest.
   * Readers are encouraged to review the introduction and purpose sections first, followed by specific areas relevant to their role (e.g., developers, testers, project managers).

These conventions aim to ensure that the document is accessible, comprehensible, and easy to use for all stakeholders involved in the PetCare project.

## Intended Audience and Reading Suggestions

**Primary Audiences:**

* **Government Authorities**  
  (e.g., Ministry of Disaster Management and Relief, BWDB)  
  → Focus on (Business Requirements), (Functional Requirements), and 13(Business Rules) for an understanding of the system’s objectives and compliance considerations.
* **System Developers and Engineers**  
  → Read (Functional Requirements), (System Requirements), and (System Architecture) for detailed technical specifications and implementation guidance.
* **Disaster Response Teams and Aid Organizations**  
  → Refer to Sections (User Requirements) and (User Interface) to understand how the system supports field operations and communication with affected citizens.
* **Telecom and Technical Integration Partners**  
  → Focus on Sections (Network Requirements) and (Integration Layer) for communication protocols and external system integration points.
* **Quality Assurance Teams**  
  → Sections (Non-functional Requirements) and (Requirement Analysis) provide quality expectations and priority levels to validate performance and reliability.
* **End Users (Citizens in Flood-Prone Areas)**  
  → While not the primary readers of this document, insights into user interaction can be found in (Citizen App Interface)

**Reading Suggestions:**

* Begin with the **Introduction** and **Purpose** sections to understand the application’s objectives and scope.
* Proceed to **Overall Description** and **System Requirements** for technical and functional details.
* Specific roles should focus on the sections most pertinent to their responsibilities as outlined above.

## References

• [830-1984 - IEEE Guide for Software Requirements Specifications](https://ieeexplore.ieee.org/document/278253)

• <https://www.scribd.com/document/388726428/Sample-SRS-Document>

• Software Requirement Engineering Power Point slide given by course teacher.

• Tutorials for SRS from YouTube.

• [Draw.io](https://app.diagrams.net/)

# Overall Description

## Product Perspective

The Automated Flood Warning and Relief System is a **new, standalone solution** designed to improve flood prediction, alert distribution, and relief coordination in Bangladesh. It is not a replacement system but integrates with existing infrastructure such as **BWDB sensors**, **weather APIs**, and **telecom networks** for real-time operations.

The system provides a **citizen-facing mobile app** and an **admin dashboard** for authorities, enabling early warnings, shelter guidance, and transparent aid tracking. It serves as a central coordination platform, linking citizens, government bodies, and aid organizations during flood emergencies.

**Problem Background**

Flooding is a recurring natural disaster in Bangladesh, especially during the monsoon season, causing widespread loss of life, property damage, and disruption to daily life. Although institutions like the **Bangladesh Water Development Board (BWDB)** and the **Ministry of Disaster Management and Relief** are responsible for managing flood response, their efforts are often hindered by:

* Lack of timely and localized flood forecasts
* Poor communication with citizens
* Manual and delayed relief coordination
* Limited transparency in aid distribution

There is a **critical need for an automated, real-time, and intelligent flood management system** that not only forecasts disasters but also helps coordinate relief operations efficiently and transparently.

**Context and Origin**

The **Automated Flood Warning and Relief System with Rapid Aid Support** is a **new, standalone software product**, not a replacement or extension of a prior system. However, it is designed to **integrate with several existing platforms**, including:

* **BWDB Sensor Network** (for water level data)
* **Weather Forecasting APIs** (e.g., meteorological services)
* **Telecom Gateways** (for SMS and alert distribution)
* **Government and NGO Aid Systems** (for relief coordination)

This system will serve as the **central coordination and communication platform** during flood emergencies. It provides a **unified dashboard for authorities** and a **mobile app for citizens**, creating two-way interaction, real-time insights, and data-driven decisions.

**Business Objectives**

* **Provide a Unified Platform:** Consolidate all aspects of pet care, from health management to social engagement, into one application.
* **Enhance User Convenience:** Streamline the process of accessing pet-related services and tracking essential health metrics.
* **Promote Responsible Pet Ownership:** Educate users through timely reminders, tips, and community interactions.
* **Drive Market Differentiation:** Position the product as a leader in the pet care industry through innovative features and user-centric design.

This section sets the foundation for understanding how PetCare aligns with both user needs and broader business strategies.

## Product Functions

### Account

**Account Register/ Sign Up**

**Account Login**

**Search**

**Update Account Information**

**Log Out.**

### Flood Forecasting & Risk Management

* Collect real-time water level, rainfall, and weather data from BWDB sensors.
* Analyze flood risks using predictive models and AI.
* Classify risk zones dynamically based on updated data.

### Alert & Notification System

* Send geo-targeted SMS and app alerts to citizens.
* Issue voice call alerts in rural/low-literacy areas.
* Push multi-language notifications.
* Provide real-time updates as situations change.

### Shelter Locator and Navigation

* Show nearby shelters based on user’s location.
* Display real-time shelter capacity and resources.
* Provide alternative routes if roads are blocked or shelters are full.

### User Management

* Register users with phone number, address, and location.
* Enable secure login for citizens and officials.
* Allow status updates: "Safe", "Need Help", "Evacuated".

### Relief Inventory & Tracking

* Manage relief supply inventory (food, water, medicine, etc.).
* Track aid dispatch, transit, and delivery in real time using GPS.
* Generate reports on relief sent vs. received.

### Relief Demand Estimation

* Calculate required aid for each region based on population and flood severity.
* Predict future needs using historical and real-time data.

### Prediction Accuracy & Performance Monitoring

* Generate accuracy reports for flood predictions.
* Continuously refine models using feedback and new data.

### Two-Way Citizen Communication

* Allow citizens to report floods, roadblocks, missing persons.
* Route citizen reports to local authorities or emergency responders.
* Enable in-app chat or emergency call features.

### Transparency & Corruption Control

* Log all aid transactions and movements.
* Detect discrepancies in aid distribution.
* Enable anonymous citizen corruption reporting.
* Maintain audit trails for all official actions.

### Feedback & Review System

* Collect citizen ratings for shelter quality and relief experience.
* Log system usability feedback for continuous improvement.

### Coordination & Integration Tools

* Sync operations with NGOs, local disaster teams, and telecoms.
* Integrate weather APIs, satellite imagery, GIS mapping, and SMS gateways.
* Support third-party dashboard views for authorized partners.

### Analytics and Decision Support

* Predict potential flood areas and damage impact.
* Recommend resource deployment locations.
* Visualize heatmaps of risk, aid demand, and shelter load.

### Admin Dashboard (BWDB & Relief Authorities)

* View real-time flood data and citizen reports.
* Manage shelters and relief logistics.
* Generate exportable reports (PDF, CSV).
* Customize alert thresholds and risk criteria.

### Mobile App (Citizen-Facing)

* Display flood alerts, shelter routes, and safety tips.
* Work in offline mode (e.g., store last known shelter info).
* Support multi-language UI and emergency numbers.

### System Reliability and Redundancy

* Support offline/low-bandwidth alert fallback mechanisms.
* Ensure automatic failover to backup systems.

## User Classes and Characteristics

**1. Citizens (General Public)**

* **Frequency of Use:** High during flood seasons.
* **Technical Expertise:** Varies widely; includes low-literate and rural users.
* **Functions Used:** Receive alerts, find shelters, report emergencies, provide feedback.
* **Security/Access Level:** Basic; limited to personal information and location data.
* **Educational Level:** Varies (primary to higher education).
* **Experience:** Likely first-time or infrequent users during emergencies.
* **Key Needs:**
  + Easy-to-use mobile interface or SMS-based alerts.
  + Real-time, localized flood and shelter information.
  + Offline support or low-bandwidth functionality.

**Importance Level:** **High Priority**  
This is the primary beneficiary group; failure to meet their needs would defeat the system’s core purpose.

**2. Local Disaster Management Authorities**

* **Frequency of Use:** Regular during flood events; periodic during preparedness phases.
* **Technical Expertise:** Moderate; comfortable with basic dashboard usage.
* **Functions Used:** Respond to citizen reports, manage shelters, monitor alerts, dispatch aid.
* **Security/Access Level:** Medium; requires access to local-level shelter and relief data.
* **Educational Level:** Generally educated and trained in local administration.
* **Experience:** Familiar with emergency protocols but not always with digital tools.

**Importance Level:** **High Priority**  
They are responsible for local-level action and decision-making based on system insights.

**3. BWDB and Ministry Officials**

* **Frequency of Use:** Frequent during system monitoring and relief coordination.
* **Technical Expertise:** High; capable of handling predictive models, analytics, and configurations.
* **Functions Used:** Manage forecasts, trigger alerts, track aid distribution, generate reports.
* **Security/Access Level:** High; access to sensitive system-wide data.
* **Educational Level:** High; typically engineers, scientists, or administrators.
* **Experience:** Experienced in disaster management, technical systems, and policy enforcement.

**Importance Level:** **High Priority**  
Their input, monitoring, and coordination are essential to the overall effectiveness of the system.

**4. Aid Organizations and NGOs**

* **Frequency of Use:** Medium to high during relief efforts.
* **Technical Expertise:** Varies; generally moderate.
* **Functions Used:** View needs assessments, plan logistics, coordinate with authorities.
* **Security/Access Level:** Restricted; read-only or partner-level access.
* **Educational Level:** Moderate to high.
* **Experience:** Experienced in humanitarian logistics and aid delivery.

**Importance Level:** **Medium Priority**  
While not system operators, their access supports synchronized relief distribution.

**5.Telecom Partners**

* **Frequency of Use:** Continuous (automated systems).
* **Technical Expertise:** High (backend integration and communication systems).
* **Functions Used:** Deliver alerts via SMS and voice.
* **Security/Access Level:** API-level access only.
* **Educational Level:** Technical/Engineering background.
* **Experience:** High experience with network infrastructure and public communication.

**Importance Level:** **Medium Priority**  
Essential for alert delivery but require minimal direct interaction with the user interface.

## Hardware and Operating Environment

The *Automated Flood Warning and Relief System* will operate across multiple platforms and environments, supporting both administrative and citizen-facing functionalities. The environment includes mobile devices, desktop systems, servers, and cloud infrastructure. The following outlines the required hardware and software environments:

**1. Server Environment (Backend)**

* **Hardware Platform:** Cloud-based virtual servers (AWS / Google Cloud / Azure)
* **Processor:** Minimum 4-core CPU, scalable depending on data load
* **Memory:** Minimum 8 GB RAM (scalable)
* **Storage:** SSD storage with backup and redundancy (minimum 500 GB)
* **Operating System:** Linux (Ubuntu 20.04 or later)
* **Web Server:** Apache or Nginx
* **Database:** MySQL / PostgreSQL
* **Other Components:**
  + GIS software for mapping (e.g., QGIS, Mapbox)
  + APIs for weather, satellite data, and telecom integration
  + Containerization (Docker) and orchestration (Kubernetes)

**2. Client Environment – Mobile Application**

* **Target Devices:** Android smartphones (version 6.0 and above), iOS devices (version 12 and above)
* **Supported Features:**
  + Push notification service (Firebase Cloud Messaging / Apple Push Notification Service)
  + GPS for location tracking
  + Offline storage for alerts and shelter maps
* **User Interface:** Lightweight, multilingual, optimized for rural connectivity and low memory devices

**3. Administrative Dashboard (Web)**

* **Target Devices:** Desktop/laptop computers used by BWDB and relief agencies
* **Browser Compatibility:** Chrome, Firefox, Microsoft Edge (latest 2 versions)
* **Operating Systems Supported:** Windows 10+, Linux, macOS
* **Security Measures:** HTTPS with TLS encryption, role-based access control

**4. Network Requirements**

* **Internet:** Required for real-time data synchronization, alert delivery, and dashboard access
* **Mobile Networks:** Integration with telecom operators for SMS/voice alerts
* **Cloud Infrastructure:** Scalable backend for handling disaster load surges

## Design and Implementation Constraints

The design and implementation of the *Automated Flood Warning and Relief System with Rapid Aid Support* must adhere to several technical and organizational constraints, which influence development choices and system architecture.

**1. Government Infrastructure Compliance**

* The system must be compatible with existing **BWDB systems and databases**.
* All data sharing must comply with regulations set by the **Ministry of Disaster Management and Relief**.

**2. Device and Connectivity Limitations**

* The **citizen-facing app must work on low-end Android devices** and support **offline functionality** in areas with limited or no internet.
* Must support **low-bandwidth data usage**, especially in rural zones.

**3. Telecom Dependency**

* Alert delivery via SMS and voice calls depends on **third-party telecom providers**.
* System must integrate with **telecom APIs** and maintain alert reliability even during disaster-related network outages.

**4. Hosting and Scalability**

* The system must be **cloud-hosted** to ensure reliability and scalability during peak usage (e.g., natural disaster events).
* System design must support **auto-scaling** to handle sudden user surges.

**5. Security and Privacy**

* Must implement **end-to-end encryption** for all sensitive data (e.g., user location, relief records).
* Role-based access must be enforced across all modules (citizens, local authorities, BWDB officials).

**6. GIS and Mapping Integration**

* The system must integrate with external **GIS platforms** (e.g., Google Maps, Mapbox) to visualize shelters, flood zones, and aid distribution.
* Real-time location tracking must be optimized for both urban and rural geography.

**7. AI Model Constraints**

* Prediction models must be **interpretable and auditable** for use by government agencies.
* The system must allow easy updates of ML algorithms without system downtime.

## User Documentation

The following user documentation components will be provided:

1. **User Manuals:**
   * Step-by-step instructions for setting up and using the software.
   * Available in PDF format and accessible via the application help section.
2. **Online Help:**
   * Context-sensitive help embedded within the application.
   * Covers common tasks, troubleshooting, and FAQs.
3. **Video Tutorials:**
   * Short videos demonstrating key features like pet profile creation, appointment scheduling, and order management.
   * Available on the official website and YouTube channel.
4. **Quick Start Guide:**
   * A concise guide included in the mobile app onboarding process to familiarize new users with essential features.
5. **API Documentation (for developers):**
   * Detailed documentation of all external APIs used in the system for third-party integration.

These resources will ensure that all user classes, from pet owners to service providers, can effectively use the PetCare application.

# System Requirements

## System Features

### Account

#### SIGN UP

**1.1** The system shall allow users to register by entering their name, phone number, address, and password.  
**1.2** The system shall send an OTP (One-Time Password) to the user’s phone number for verification.  
**1.3** Upon successful verification, the system shall create a new user account and store the information in the database.

**Priority Level:** High

**Precondition:** User must provide a valid phone number.

**Cross-references:**

## ****3.1.2.1**** Account Login

**1.1** The system shall allow users to log in using their phone number and password.  
**1.2** The login credentials shall be validated against the user database.  
**1.3** If the login is successful, the user will be redirected to their home/dashboard page.  
**1.4** If login fails three times, the system shall lock the account for one hour (optional security function).  
**1.5** In case of a wrong password, a verification code may be sent via email or SMS to allow retry.

**Priority Level:** High

**Precondition:** User must be registered with valid credentials.

**Cross-references:** 3.6(QA4)

## ****3.1.3.1**** Search Account / View Profile

**1.1** The system shall allow users to search and view their own account information after logging in.  
**1.2** The displayed information shall include name, phone number, address, and registered location.  
**1.3** The system shall retrieve the account information from the database in real-time.

**Priority Level:** Medium

**Precondition:** User must be logged in.

**Cross-references:** 3.1.1.1,3.1.2.1

## ****3.1.4.1**** Update Account Information

**1.1** The system shall allow users to update their profile information (name, address, location, password).  
**1.2** Upon submission, the system shall verify the input and update the corresponding data in the database.  
**1.3** A success message shall be shown after the update is completed.

**Priority Level:** Medium

**Precondition:** User must be authenticated.

**Cross-references:** 3.1.1.1,3.1.2.1,3.6(QA6)

## ****3.1.5.1**** Log Out

**1.1** The system shall allow users to log out of their account at any time.  
**1.2** After logout, the session shall be terminated, and the user shall be redirected to the login or welcome page.  
**1.3** Any cached session data shall be cleared securely to prevent unauthorized access.

**Priority Level:** High

**Precondition:** User must be logged in.

**Cross-references:** 3.6(QA4)

## ****3.1.6.1 Flood Forecasting****

**1.1** The system shall collect real-time data from BWDB sensors, including water level, rainfall, and flow rate.  
**1.2** The system shall analyze incoming data using predictive models to detect flood threats.  
**1.3** The system shall classify zones into risk levels (low, medium, high) based on data thresholds.  
**1.4** The system shall continuously update risk predictions as new data becomes available.  
**1.5** If a flood risk crosses a critical threshold, the system shall notify relevant authorities.

**Priority Level:** High  
**Precondition:** Sensor and weather data sources must be operational.  
**Cross-references:** 3.1.1,2.2.2,2.2.3

## ****3.1.7.1 Alert Notification System****

**1.1** The system shall generate flood alerts when forecasted thresholds are met.  
**1.2** The system shall deliver alerts via SMS, mobile app, and voice calls.  
**1.3** Alerts shall be geo-targeted based on user location.  
**1.4** Alerts shall be issued in both Bangla and English.  
**1.5** The system shall escalate alert frequency based on risk level.  
**Priority Level:** High  
**Precondition:** User is registered with location data.  
**Cross-references:** 2.2.2,2.2.3

## ****3.1.8.1 Shelter Locator and Navigation****

**1.1** The system shall allow users to view nearby shelters on a map.  
**1.2** The system shall display real-time shelter capacity and directions.  
**1.3** The system shall update shelter availability based on coordinator input.  
**1.4** The system shall suggest alternative shelters if one is full.  
**1.5** The system shall notify users when capacity at their selected shelter is near full.  
**Priority Level:** High  
**Precondition:** GPS/location access must be enabled.  
**Cross-references:** 2.2.4,2.2.15

## ****3.1.9.1 Relief Tracking and Distribution****

**1.1** The system shall allow authorities to input and track inventory of relief materials.  
**1.2** The system shall calculate demand per ward based on flood impact and population.  
**1.3** The system shall assign and track delivery routes with timestamps and GPS.  
**1.4** The system shall generate distribution logs and reports.  
**1.5** If discrepancies occur, the system shall flag them for audit.  
**Priority Level:** High  
**Precondition:** Relief data must be entered into the system.  
**Cross-references:** 2.2.6,2.2.7,2.2.14

## ****3.1.10.1Citizen Reporting & Feedback****

**1.1** The system shall allow citizens to report flood status, roadblocks, or emergencies.  
**1.2** Reports shall be sent to the nearest local authority for review.  
**1.3** The system shall allow users to submit anonymous feedback or corruption complaints.  
**1.4** The system shall store and analyze user feedback to improve system performance.  
**1.5** Feedback marked urgent shall trigger an alert to response teams.  
**Priority Level:** High  
**Precondition:** User must be logged in.  
**Cross-references:** 2.2.9,2.2.10,2.2.14,2.2.15

## ****3.1.11.1 Admin Dashboard (BWDB)****

**1.1** The dashboard shall display flood risk maps and real-time sensor data.  
**1.2** The dashboard shall include relief inventory and shelter status overview.  
**1.3** Admins shall be able to generate and export reports in PDF/CSV.  
**1.4** The system shall provide predictive analytics for risk planning.  
**1.5** All actions on the dashboard shall be logged for auditing purposes.  
**Priority Level:** High  
**Precondition:** Admin must log in with elevated access.  
**Cross-references:** 2.4.2,2.4.3

## ****3.1.12.1 Emergency Response Coordination****

**1.1** The system shall track available rescue personnel and equipment.  
**1.2** The system shall generate and update deployment plans for teams.  
**1.3** The system shall notify response teams via internal alerting.  
**1.4** The system shall allow real-time updates from field agents.  
**1.5** The dashboard shall visualize response coverage by area.  
**Priority Level:** High  
**Precondition:** Emergency team data must be loaded.  
**Cross-references:** 2.4.2,2.4.3,,2.2.6,2.2.8,2.2.12,2.2.13

## Non-Functional/Quality Requirements

**QA1: Usability**

A citizen with basic smartphone knowledge shall be able to complete registration and receive a location-based flood alert in an average of **3 minutes**, with a maximum of **5 minutes** in low-connesctivity areas.

**Priority Level:** High

**Precondition:** Mobile device is connected to a mobile network.

**Cross-references:** Sections 2.2.3 (Alert Notification),2.2.14 (Citizen App UI),

**QA2: Performance**

The system shall generate and send alerts to **all users within a high-risk zone** within **10 seconds** of flood detection.

**Priority Level:** High

**Precondition:** Flood detection thresholds are met.

**Cross-references:** 2.2.3 (Alert Notification),2.2.7 (Automatic Flood Detection)

**QA3: Reliability**

The system shall maintain **99.9% uptime** during flood seasons (June–October), including backend servers and alerting services.

**Priority Level:** High

**Precondition:** Infrastructure is hosted on a reliable cloud platform.

**Cross-references:** 2.2

**QA4: Security**

All user data, including location, phone number, and reports, shall be stored securely using **AES-256 encryption**, and access shall be restricted via **role-based authentication**.

**Priority Level:** High

**Precondition:** User is authenticated with proper role access.

**Cross-references:** 2.2.1)

**QA5: Scalability**

The system shall be able to handle at least **500,000 concurrent users** during peak disaster events without performance degradation.

**Priority Level:** Medium

**Precondition:** Cloud infrastructure scaling is enabled.

**Cross-references:** 2.6(QA2), 2.2.1

**QA6: Maintainability**

The system shall allow modular updates (e.g., new alert methods, language packs) to be deployed without affecting running services.

**Priority Level:** Medium

**Precondition:** Version control and containerization (e.g., Docker) are in place.

**Cross-references:** 2.6

**QA7: Throughput**

The system shall be capable of processing and distributing flood alerts to a minimum of **10,000 users per second** during peak load conditions.

**Priority Level:** High

**Precondition:** High-risk zone alert is triggered; cloud load balancer is active.

**Cross-references:** 2.6, 1.1

**QA8: Uptime**

The system shall maintain **99.9% uptime availability** during the monsoon season (June–October), ensuring uninterrupted alerting, dashboard access, and reporting.

**Priority Level:** High

**Precondition:** Infrastructure is deployed on a reliable cloud platform with redundancy enabled.

**Cross-references:** 2.2.7,2.2.8, 2.4

## Project Requirements

**Effort Estimation Using COCOMO**

To estimate the required effort and schedule for the project, the **Basic COCOMO Model** is applied.

**COCOMO Basic Formula:**

Effort (PM)=a×(KDSI)b\text{Effort (PM)} = a \times (\text{KDSI})^bEffort (PM)=a×(KDSI)b

Where:

* **Effort (PM)** = Person-Months
* **KDSI (KLOC)** = Thousands of Lines of Code (estimated at 25 KLOC for this system)
* **a = 2.4**, **b = 1.05** (for organic projects)

**Effort Estimation:**

Effort (PM)=2.4×(25)1.05≈68.4 person-months\text{Effort (PM)} = 2.4 \times (25)^{1.05} \approx 68.4 \text{ person-months}Effort (PM)=2.4×(25)1.05≈68.4 person-months

**Schedule Estimation:**

TDEV (months)=2.5×(Effort)0.38≈2.5×(68.4)0.38≈11.2 months\text{TDEV (months)} = 2.5 \times (Effort)^{0.38} \approx 2.5 \times (68.4)^{0.38} \approx 11.2 \text{ months}TDEV (months)=2.5×(Effort)0.38≈2.5×(68.4)0.38≈11.2 months

**Estimated Team Composition:**

* **Project Manager** – 1
* **Software Developers** – 5
* **QA Engineers** – 2
* **UI/UX Designers** – 2
* **System Administrator** – 1
* **Technical Writer** – 1

**Estimated Resources Summary:**

* **Total Effort**: ~68 person-months
* **Estimated Timeline**: 10–12 months
* **Team Size**: 8–10 members

**Tools**

**Selenium:**

To ensure robust testing, **Selenium** will be used starting **from Week 6** for automated testing of the system’s web components.

**Purpose of Selenium:**

* Perform functional testing across browsers and platforms
* Integrate into CI/CD pipeline for continuous testing
* Automate regression and smoke tests

**Additional Tools:**

| **Tool** | **Purpose** |
| --- | --- |
| **JIRA/Trello** | Task management and sprint planning |
| **GitHub** | Version control and code repository |
| **Postman** | API testing and automation |
| **Docker** | Containerization of microservices |
| **Slack** | Internal team communication |

**Services**

**1. Cloud Hosting and Infrastructure**

The application will be hosted on cloud platforms such as **AWS** or **Google Cloud**. Services include:

* Scalable compute and storage
* Real-time disaster alert distribution
* Secure user data management

**2. Third-Party Integrations**

* SMS gateways for alert delivery
* GIS APIs for real-time flood mapping
* Weather forecasting APIs for flood prediction models

**3. Ongoing Maintenance and Support**

* Post-deployment bug fixing
* Performance optimization
* Security patching and feature updates

**4. User Support Services**

A support platform will be used (e.g., **Freshdesk** or **Zendesk**) to:

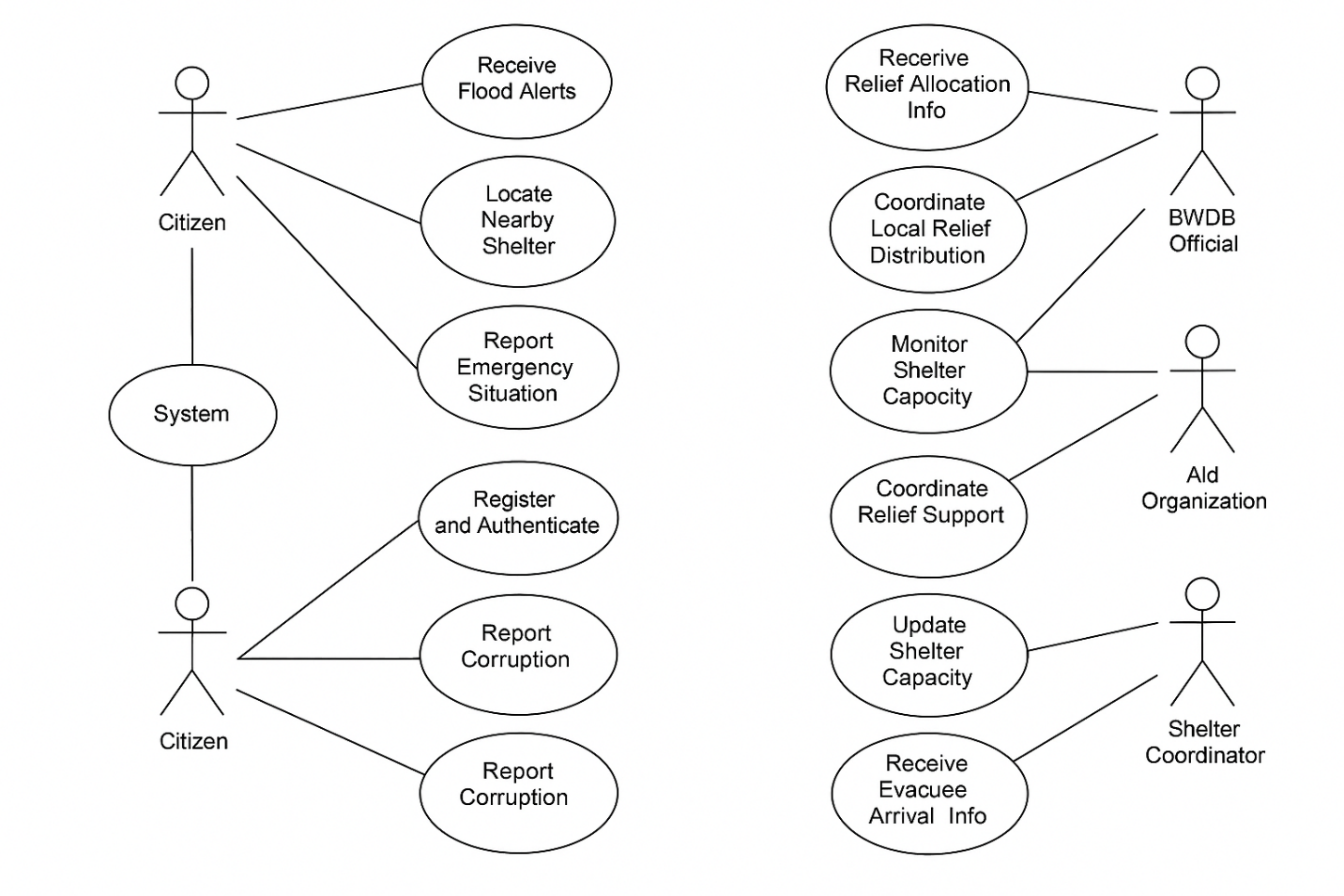
* Handle citizen queries and complaints
* Track feedback on relief distribution
* Monitor corruption reports

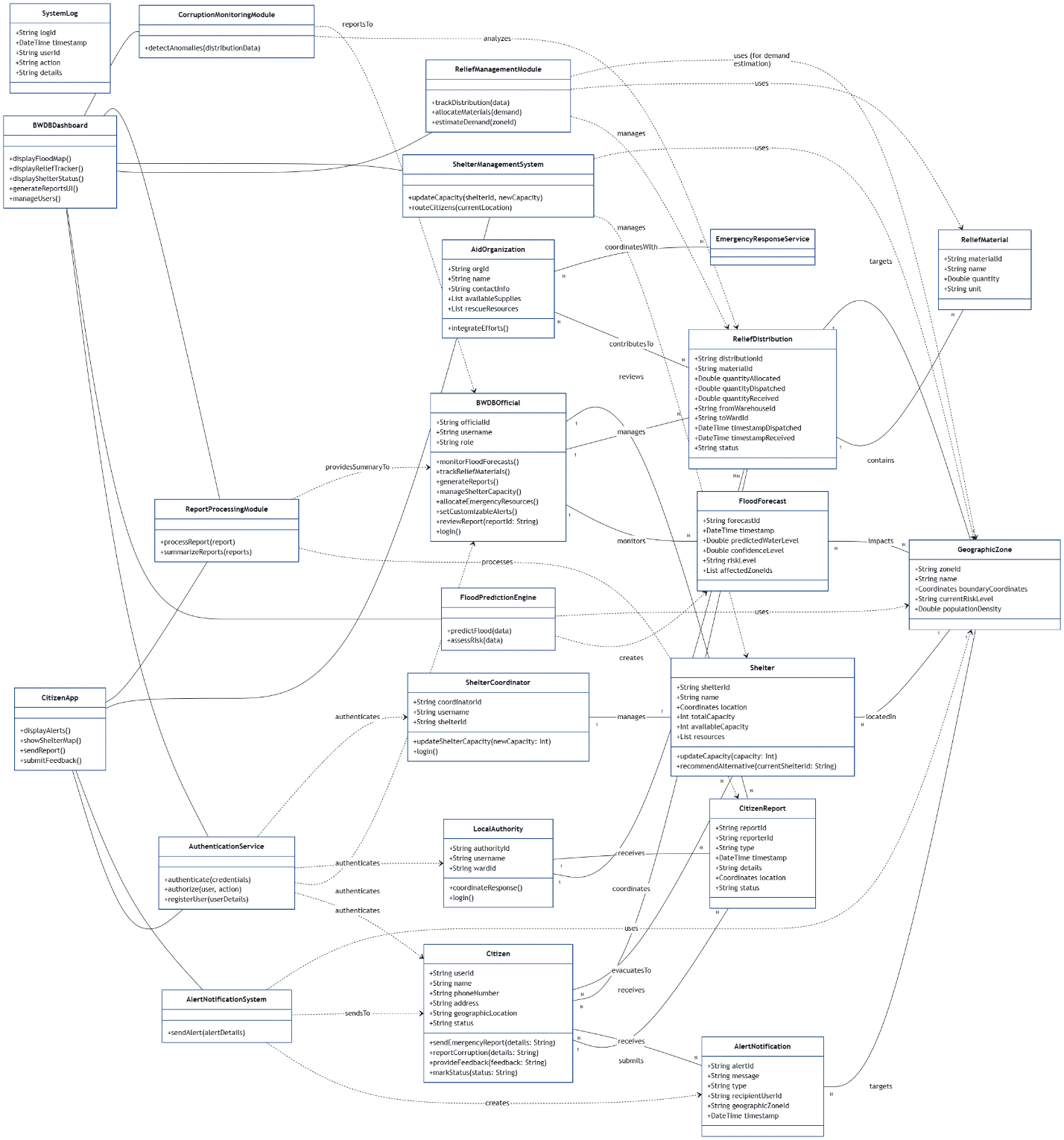
\_\_ **Budget Estimate**

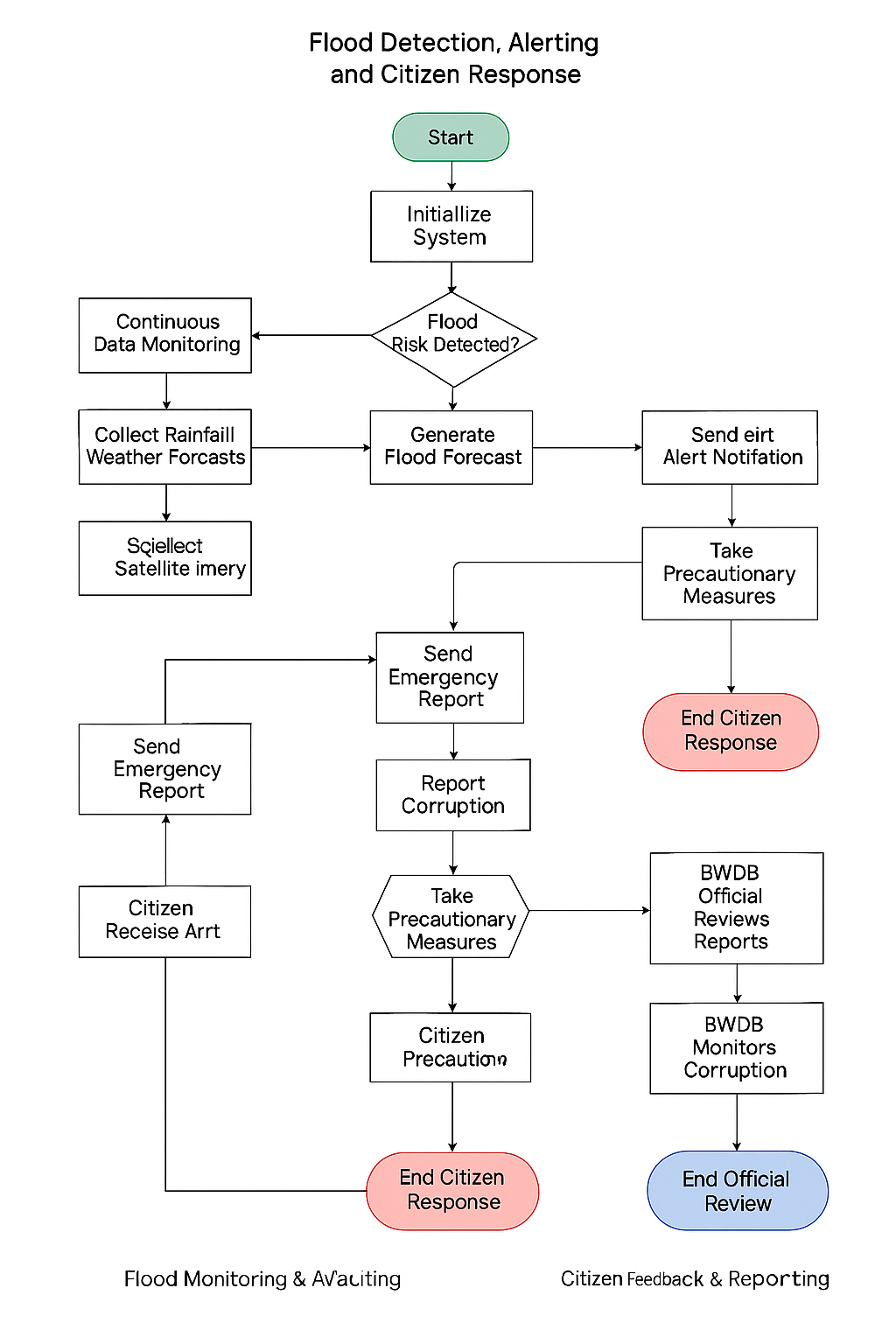
| **Budget Component** | **Estimated Cost (USD)** |
| --- | --- |
| Development (Human Resource) | $100,000 |
| Testing and Tools (Selenium, etc.) | $10,000 |
| Cloud Hosting (AWS/GCP) | $15,000 |
| Training and Documentation | $5,000 |
| Miscellaneous and Contingency | $10,000 |
| **Total Estimated Budget** | **$140,000–$150,000** |

# Design and Interface Requirements

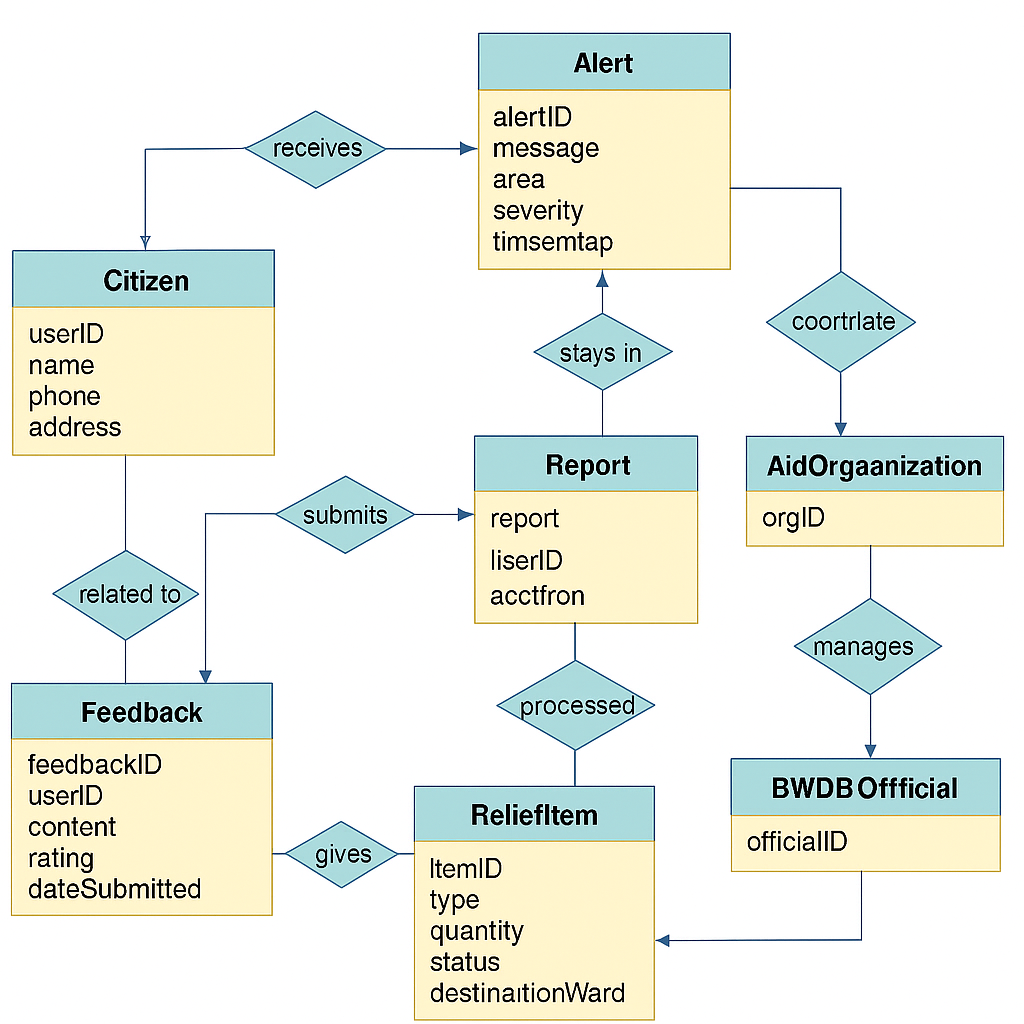
## UML Diagrams

  
 **USE CASE DIAGRAM**

**Text Box 1, Textbox**

**Text Box 1, Textbox**

**ACTIVITY DIAGRAM**

Text Box 1, Textbox

## Data Dictionary

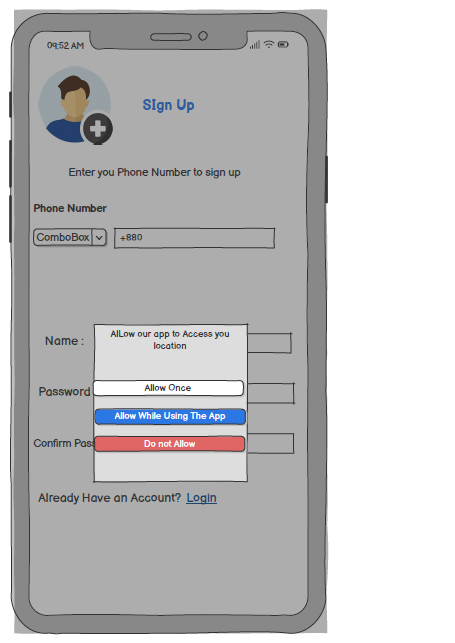
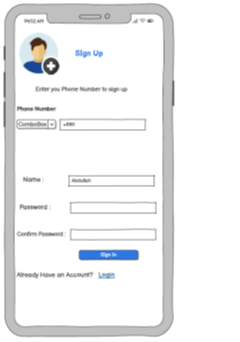
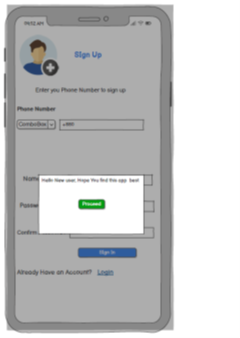


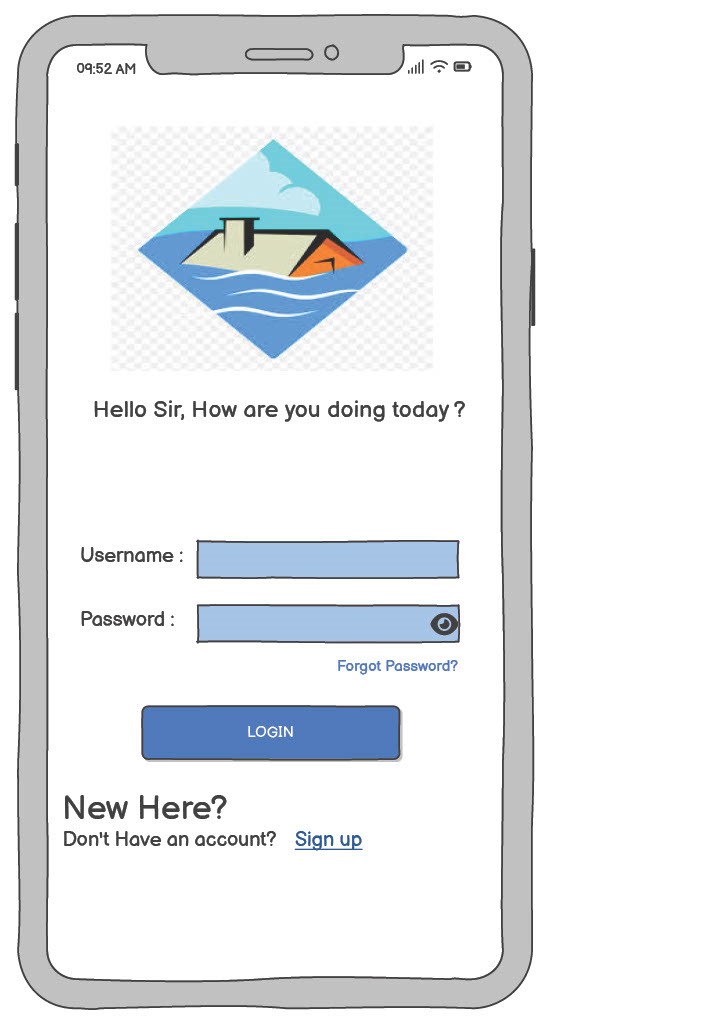
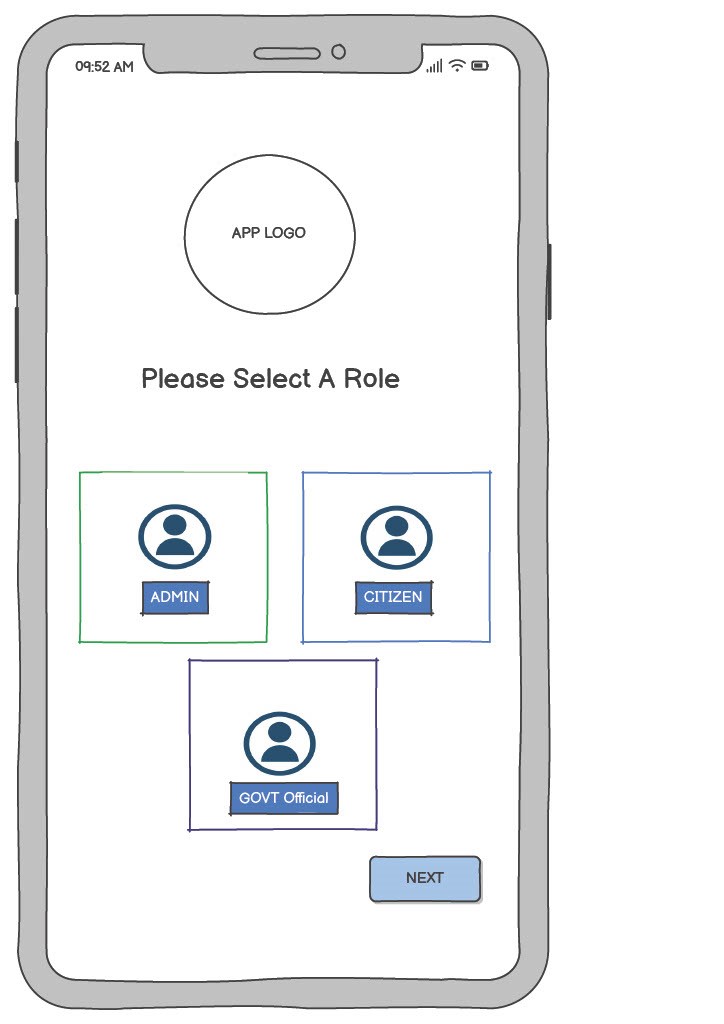
## UI/UX Design Specification

### 4.3.1 Design Tool

Tool Selected tool → **Canva**

### 4.3.2 DESIGNS ARE BELOW

**4.3.2 Design are bel ow**

A login screen with a house logo

AI-generated content may be incorrect., PictureA login screen with a house logo

AI-generated content may be incorrect., Picture

A screenshot of a cell phone

AI-generated content may be incorrect., PictureA close-up of a phone

AI-generated content may be incorrect., PictureA screenshot of a phone verification

AI-generated content may be incorrect., PictureA screenshot of a phone verification

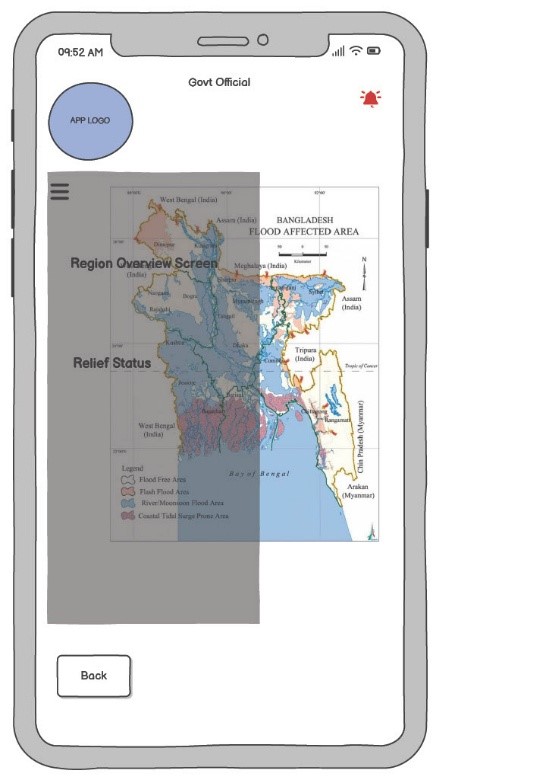
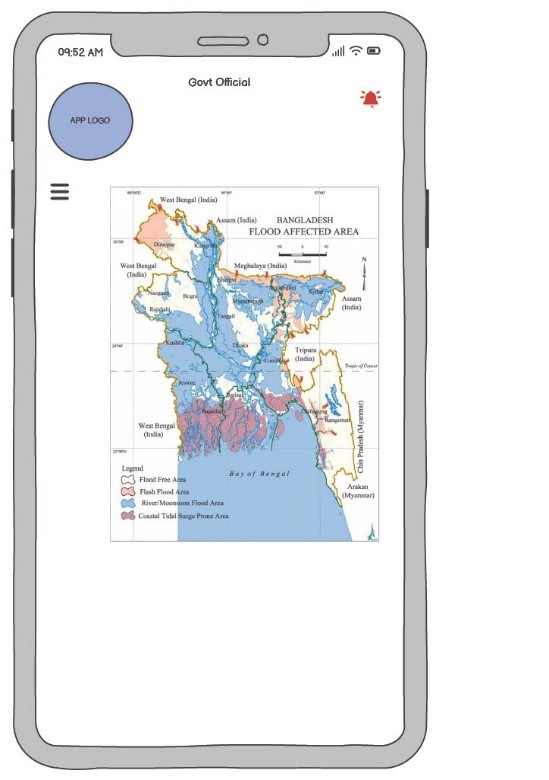
AI-generated content may be incorrect., Picture

A screenshot of a computer

AI-generated content may be incorrect., Picture

A screen shot of a phone

AI-generated content may be incorrect., PictureA close-up of a cell phone

AI-generated content may be incorrect., Picture

A close-up of a cell phone

AI-generated content may be incorrect., PictureA screenshot of a phone

AI-generated content may be incorrect., PictureA cell phone with pie chart

AI-generated content may be incorrect., PictureA close-up of a cell phone

AI-generated content may be incorrect., Picture

A close-up of a cell phone

AI-generated content may be incorrect., PictureA close-up of a cell phone

AI-generated content may be incorrect., PictureA close-up of a cell phone

AI-generated content may be incorrect., PictureA screenshot of a phone

AI-generated content may be incorrect., Picture

A screenshot of a phone

AI-generated content may be incorrect., PictureA screen shot of a phone

AI-generated content may be incorrect., PictureA screen shot of a phone

AI-generated content may be incorrect., PictureA screen shot of a phone

AI-generated content may be incorrect., Picture

A screen shot of a phone

AI-generated content may be incorrect., PictureA cell phone showing a map

AI-generated content may be incorrect., PictureA close-up of a cell phone

AI-generated content may be incorrect., PictureA close-up of a cell phone

AI-generated content may be incorrect., Picture

A screen shot of a cell phone

AI-generated content may be incorrect., PictureA close-up of a cell phone

AI-generated content may be incorrect., PictureA cell phone with a green screen

AI-generated content may be incorrect., PictureA close-up of a phone

AI-generated content may be incorrect., Picture

A close-up of a phone

AI-generated content may be incorrect., PictureA close-up of a phone

AI-generated content may be incorrect., PictureA close-up of a phone

AI-generated content may be incorrect., PictureA close-up of a phone

AI-generated content may be incorrect., Picture

A close-up of a smartphone

AI-generated content may be incorrect., Picture