Flood & Cyclone Early Warning System – SRS

# 1. Introduction

## 1.1 Purpose

The purpose of the Flood & Cyclone Early Warning System is to provide timely alerts, risk assessments, and predictive analytics to minimize the impact of natural disasters. Using NASA’s open datasets (e.g., precipitation, sea-level rise, cyclone paths, flood-prone regions), the system aims to help communities, government agencies, and organizations prepare and respond effectively to potential flood and cyclone events.

## 1.2 Scope

The software will provide the following capabilities:  
- Collect and process NASA datasets related to precipitation, storm patterns, and flood zones.  
- Display real-time risk maps and dashboards.  
- Provide alerts and notifications when high-risk thresholds are reached.  
- Allow users to filter by location, date range, and disaster type.  
- Generate trend analysis and predictive insights using historical data.  
- Export reports and datasets to CSV or Excel.

## 1.3 Target Users

- Disaster management authorities  
- Government agencies (environment, water resources, public safety)  
- NGOs and relief organizations  
- Citizens living in flood or cyclone-prone areas

# 2. Overall Description

## 2.1 Product Perspective

The Flood & Cyclone Early Warning System is a web-based MERN stack application (MongoDB, Express.js, React.js, Node.js). It integrates NASA’s open datasets (via APIs or bulk data sources) to generate real-time disaster predictions.  
  
Technology stack:  
- Frontend: React.js + Tailwind CSS  
- Backend: Node.js + Express.js  
- Database: MongoDB (cloud-hosted for scalability)  
- APIs/Datasets: NASA Earth Data, NASA POWER, NASA GES DISC, IMERG rainfall data  
- Visualization: Chart.js / D3.js / Mapbox for geospatial visualization  
- Editor: Visual Studio Code

## 2.2 Operating Environment

- Web-based system, accessible via any modern browser (Chrome, Edge, Firefox)  
- Server: Node.js (Express backend)  
- Database: MongoDB (Atlas preferred)

## 2.3 Design & Implementation Constraints

- Must handle large datasets (100k+ records) without performance issues  
- NASA dataset APIs may have rate limits and latency  
- Offline mode is limited (relies on cloud datasets)  
- Critical alerts must trigger within 30 seconds of event detection

# 3. Functional Requirements

## 3.1 Core Features

1. Dashboard  
 - Displays live disaster summary (current flood/cyclone alerts, risk level, affected regions)  
 - Color-coded severity levels (Green: Safe, Yellow: Caution, Red: High Risk)  
 - Quick links to maps, alerts, and reports  
  
2. Disaster Data Management  
 - Fetch, store, and update datasets from NASA APIs  
 - Filter data by location, date range, disaster type  
 - Historical data repository for analysis  
  
3. Alerts & Early Warnings  
 - Real-time flood and cyclone alerts based on thresholds  
 - Email/SMS/web notifications for subscribed users  
 - Risk categorization (Low, Moderate, Severe)  
  
4. Reports & Visualization  
 - Interactive maps (flood-prone areas, cyclone tracks)  
 - Charts: rainfall trends, cyclone frequency, flood intensity  
 - Predictive analytics using historical data  
 - Export reports to CSV/Excel

# 4. Non-Functional Requirements

- Performance: Handle large-scale datasets and thousands of concurrent users  
- Usability: Simple UI with real-time maps and alerts  
- Reliability: Data consistency ensured by MongoDB & backup strategy  
- Scalability: Cloud deployment (AWS/GCP/Azure) for high availability  
- Security: Role-based access (Admin, Researcher, Public User), HTTPS, encrypted API calls

# 5. External Interface Requirements

## 5.1 User Interface

- Left-side navigation bar: Dashboard, Alerts, Data, Reports, Settings  
- Main area: Maps, charts, alert panels  
- Mobile-responsive UI

## 5.2 Hardware

- Standard PC / Laptop / Smartphone with internet  
- Minimum 2 GB RAM for smooth usage

## 5.3 Software

- Frontend: React.js  
- Backend: Node.js + Express  
- Database: MongoDB Atlas  
- APIs: NASA datasets

# 6. System Features / Workflow

1. Dashboard: View real-time disaster risk summary, maps, and alerts  
2. Data Module: Fetch & filter NASA datasets by location/time/disaster type  
3. Alerts: Generate real-time notifications when thresholds are crossed  
4. Reports: Charts, maps, and exportable reports for decision-making

# 7. Future Enhancements

- AI/ML models for predictive disaster analytics  
- Integration with IoT sensors & weather stations  
- Mobile App for offline alerts  
- Multi-language support for global adoption  
- Collaboration with other datasets (NOAA, ISRO, ESA)