UNIVERSITY OF ASIA PACIFIC

Department of Computer Science and Engineering(CSE)

Course Code: CSE 306
Course Title: System Analysis and Design Lab

Project Name: StormGuard Developers - A Disaster Preparedness Portal

Submitted By:
Md Yasin Hossain Akash - 22101153
Nusrata Ahmmed Maisha - 22101160
C.M Hasibul Hasan - 22101174

Submitted To:
Alif Ruslan
Lecturer
Computer Science and Engineering

Introduction

In the face of increasing disasters, effective response is vital but hindered by challenges like resource scarcity and communication barriers. Our project offers a tech-driven solution with efficient resource allocation, real-time updates, transparent donation tracking, multilingual support, offline access, and updated contacts. Leveraging modern web technologies, we aim to create an affordable, sustainable platform to enhance disaster resilience, accountability, and inclusivity for communities, responders, and aid organizations worldwide.

Problem Statement

Disaster management faces critical challenges, including resource scarcity, weather delays, outdated emergency contacts, and communication barriers. These obstacles delay relief efforts and compromise the safety and well-being of affected populations.

Objectives

- Develop a solution for efficient resource allocation during disasters.
- Provide real-time updates and alerts to stakeholders.
- Ensure transparency in donation tracking and fund utilization.
- Create a multilingual, offline-accessible platform for better communication.

Project Output

- Efficient resource distribution and management.
- Real-time updates for disaster situations.
- Transparent donation tracking.
- Improved communication through multilingual and offline support.

Target Audience

1. Local Communities
    - People directly affected by disasters who need timely information, resources, and assistance.
2. Emergency Responders
    - First responders, including firefighters, paramedics, and rescue teams, require real-time updates and resource allocation tools.
3. Government Agencies
    - Authorities are responsible for managing disaster response, recovery efforts, and coordination among stakeholders.
4. Volunteers
    - Individuals and groups offering their time and skills to support disaster relief and recovery efforts.
5. Non-Governmental Organizations (NGOs)
    - Humanitarian organizations work to provide aid, resources, and support to affected populations.
6. Donors and Philanthropists
    - Individuals or organizations contribute funds or resources and seeking transparency on their usage.
7. Media Outlets

    - News platforms spread updates, alerts, and public safety information during disasters.
8. Researchers and Academics
    - Experts studying disaster management, risk reduction, and resilience who require accurate data and analytics.
9. International Aid Organizations
    - Global entities providing large-scale disaster relief resources, funding, and expertise.
10. Technology Providers
    - Companies or individuals developing tools, APIs, and integrations for disaster management systems.

Functional Requirements

1. User Registration & Login
    - Enable secure user registration and login functionality.
2. User Roles & Permissions
    - Differentiate between user roles (e.g., admin, general user, emergency personnel) with role-specific access controls.
3. Emergency Alerts Notification
    - Send real-time disaster alerts to users, customized for the nature and severity of the disaster.
4. Location-based Disaster Information
    - Provide disaster updates and resources specific to the user's geographic location.
5. Volunteer Sign-up & Management
    - Allow users to register as volunteers and match assignments based on skills and availability.
6. Incident Reporting
    - Enable users to report ongoing disasters, with verification before inclusion in the alert system.
7. Disaster Preparedness Guides
    - Offer educational resources, guides, and best practices for disaster preparedness.
8. Multilingual Support
    - Present content in multiple languages to cater to diverse user groups.
9. Emergency Contact Directory
    - Provide a directory of emergency contact information for services like police and fire departments.
10. Resource Allocation Tracking
    - Track the distribution and usage of disaster relief resources.

11. Social Media Integration
    - Share alerts and updates on social media platforms for wider reach.
12. Push Notifications
    - Deliver updates and alerts directly to users' devices.
13. Donation Tracking
    - Allow users and administrators to track donations and their distribution to affected areas.
14. Automated Donation Suggestions
    - Recommend specific items or funds based on the disaster's immediate needs.
15. Data Backup and Restore
    - Ensure critical data is backed up and recoverable in case of system failure.

Non-Functional Requirements

1. Scalability
    - Handle high user loads, especially during emergencies, without performance degradation.
2. Performance
    - Respond to user requests within 2 seconds under normal conditions.
3. Availability
    - Maintain 99.9% uptime, ensuring system reliability during disasters.
4. Security
    - Encrypt all user data and comply with relevant data protection laws.

5. Usability
    - Provide a user-friendly and intuitive interface suitable for users with minimal technical skills.
6. Reliability
    - Deliver accurate and timely disaster alerts consistently.
7. Maintainability
    - Ensure the system is modular and easy to update or repair.
8. Interoperability
    - Integrate seamlessly with third-party systems such as weather services and governmental platforms.
9. Responsiveness
    - Optimize the portal for desktops, smartphones, and tablets.
10. Localization
    - Support regional settings, time zones, and localized disaster information.
11. Data Integrity
    - Ensure that data remains accurate, consistent, and protected from corruption.
12. Disaster Recovery
    - Implement a recovery plan to restore services quickly after a failure.
13. Energy Efficiency
    - Optimize system operations to reduce energy consumption, especially during peak usage periods.

Feasibility Analysis

1. Technical Feasibility

- Existing Technologies: The system can be built using widely available technologies such as Django or Node.js for the backend, and React or Angular for the frontend. APIs for weather data, geolocation, and disaster monitoring are readily available.
- Scalability: Cloud services such as AWS, Google Cloud, or Azure provide the necessary infrastructure to scale during high-traffic periods.
- Data Sources: Reliable data can be sourced from government databases, weather services, and NGOs for disaster statistics and resources.
- Development Resources: Expertise in web and mobile application development, along with integration of AI and machine learning for resource allocation and donation suggestions, is achievable with existing tools.

2. Economic Feasibility

- Development Costs: Initial costs can be minimized by leveraging open-source tools and frameworks.
- Operational Costs: Cloud-based services with pay-as-you-go models reduce upfront investment.
- Funding Opportunities: NGOs, government grants, and international aid organizations can support the project.
- Return on Investment (ROI): The system's value lies in its ability to save lives, improve disaster response efficiency, and increase donor transparency, justifying the cost of implementation.

3. Social Feasibility

- User Acceptance: The platform addresses critical pain points like lack of communication, outdated contact directories, and resource mismanagement, making it highly valuable for affected communities.
- Accessibility: Multilingual support and mobile compatibility ensure widespread adoption across diverse demographics.
- Stakeholder Engagement: Emergency responders, government agencies, NGOs, and donors will benefit from the transparency and efficiency offered by the platform.

4. Legal Feasibility

- Compliance: Adhering to data protection laws (e.g., GDPR) and local disaster management regulations ensures the system operates legally.
- Donor Transparency: Clear tracking of donations complies with regulations for financial accountability.

5. Operational Feasibility

- Ease of Use: A user-friendly interface with offline capabilities ensures accessibility even in challenging situations.
- Training Needs: Minimal training is required due to the intuitive design, but guides and FAQs will be provided.
- Maintenance: Regular updates, bug fixes, and integration of new features are manageable with a modular design.

6. Environmental Feasibility

- Energy Efficiency: Optimizing backend processes and using cloud services reduces the energy footprint of the platform.

- Disaster Impact Mitigation: By improving resource allocation and communication, the platform reduces the overall environmental and social impact of disasters.

Technical Requirements

1. Software Requirements

- Frontend Development:
  - Frameworks: React.js, Angular, or Vue.js for building an interactive and user-friendly interface.
  - Tools: HTML5, CSS3, JavaScript, and responsive design techniques.
- Backend Development:
  - Frameworks: Django (Python), Node.js, or Ruby on Rails for robust backend functionality.
  - Database: PostgreSQL, MySQL, or MongoDB for managing user data, disaster information, and resources.
  - APIs: Integration with external APIs for weather data, geolocation, and communication tools (e.g., Google Maps API, OpenWeatherMap).
- AI and Machine Learning:
  - Libraries: TensorFlow, PyTorch, or Scikit-learn for implementing AI-powered resource allocation and donation suggestions.
  - Models: Predictive algorithms for disaster impact assessment and real-time data analysis.
- Cloud Services:
  - Hosting: AWS, Google Cloud, or Microsoft Azure for scalable and reliable hosting.

    - Storage: Cloud-based storage solutions for managing large datasets and backups.
- Mobile Development:
    - Framework: Flutter or React Native for cross-platform mobile app development.
- Push Notifications:
    - Services: Firebase Cloud Messaging or OneSignal for real-time notifications.
- Social Media Integration:
    - APIs: Facebook Graph API, and Twitter API for sharing updates and alerts.

2. Hardware Requirements

- Servers:
    - Minimum: A quad-core processor, 16GB RAM, and 1TB SSD are required for initial deployment.
    - Recommended: Cloud-based scalable infrastructure for handling large traffic.
- End-user Devices:
    - Compatibility with desktop, mobile, and tablet devices, including low-end smartphones.

3. Network Requirements

- Bandwidth: High-speed internet with at least 100 Mbps bandwidth for the server to handle concurrent users.
- Connectivity: Reliable connections for API integrations and real-time data syncing.

4. Security Requirements

- Data Encryption:

    - SSL/TLS for secure communication between users and the server.
    - AES encryption for stored sensitive data.
- Authentication and Authorization:
    - OAuth2.0 or JWT for secure user authentication.
    - Role-based access control (RBAC) for managing user permissions.
- Data Protection:
    - Compliance with GDPR and other local data privacy regulations.

5. Scalability Requirements

- Load Balancing: Tools like NGINX or HAProxy to distribute server traffic efficiently.
- Horizontal Scaling: Capability to add more servers to handle increased user loads during disasters.

6. Backup and Recovery

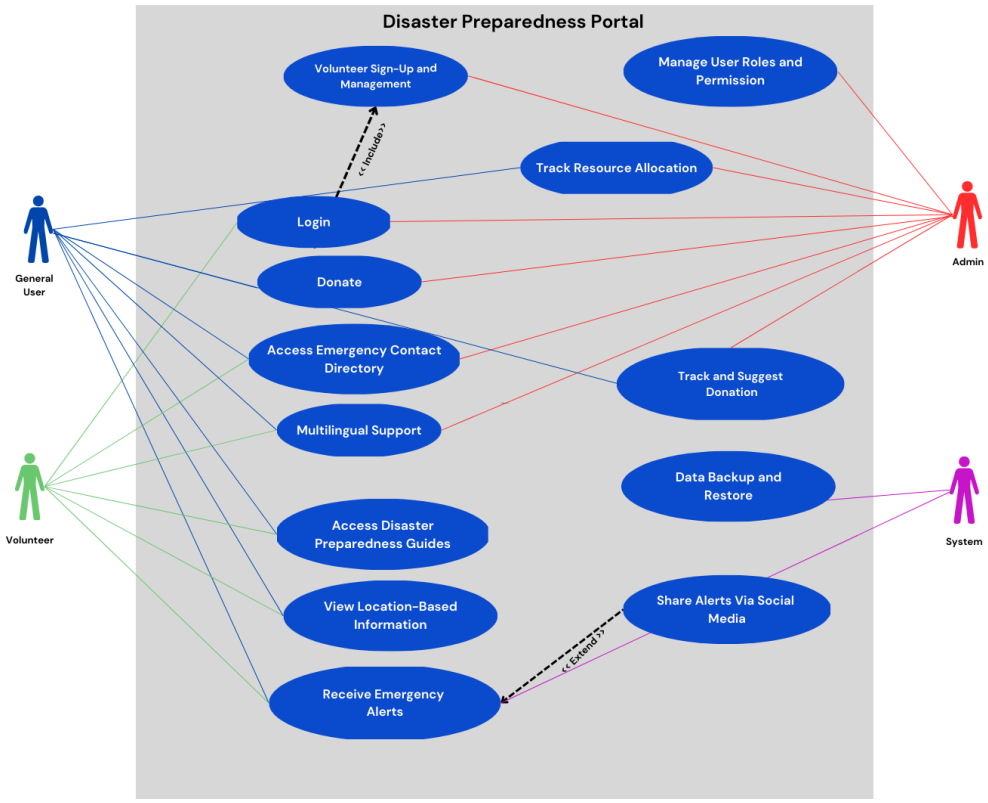
- Automated Backups: Daily backups of critical data to cloud storage.
- Disaster Recovery: A recovery plan ensuring system restoration within 30 minutes of failure.

Project Designing Tools

In developing the project, we utilized key project designing tools to structure and streamline the development process. Below are the essential diagrams and their purpose in the system's design:

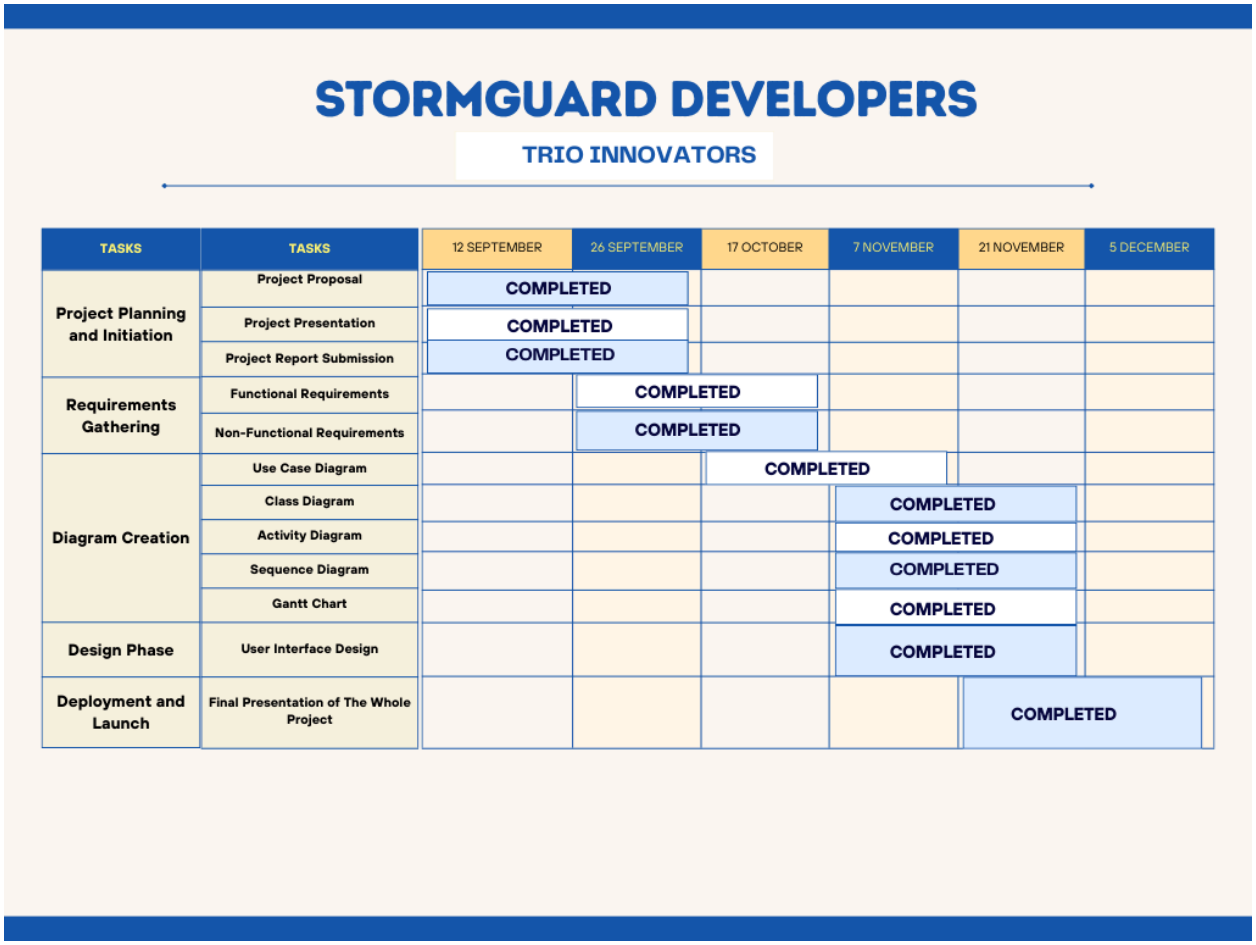
Use Case Diagram

Purpose: Represents the interactions between users and the system, identifying the actions they can perform.



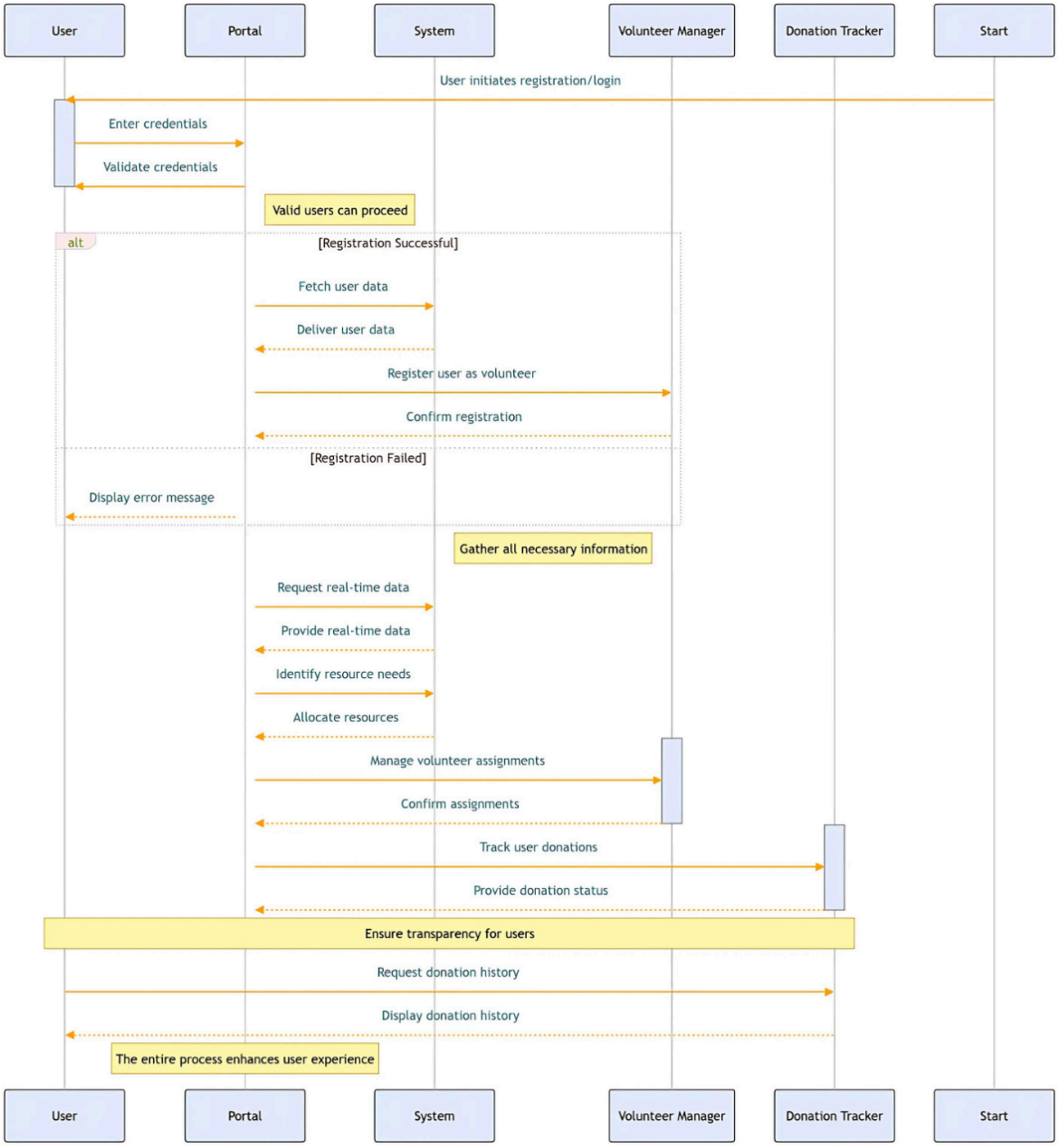
Gantt Chart

Purpose: Displays the project timeline, tracking tasks, milestones, and deadlines for project management.



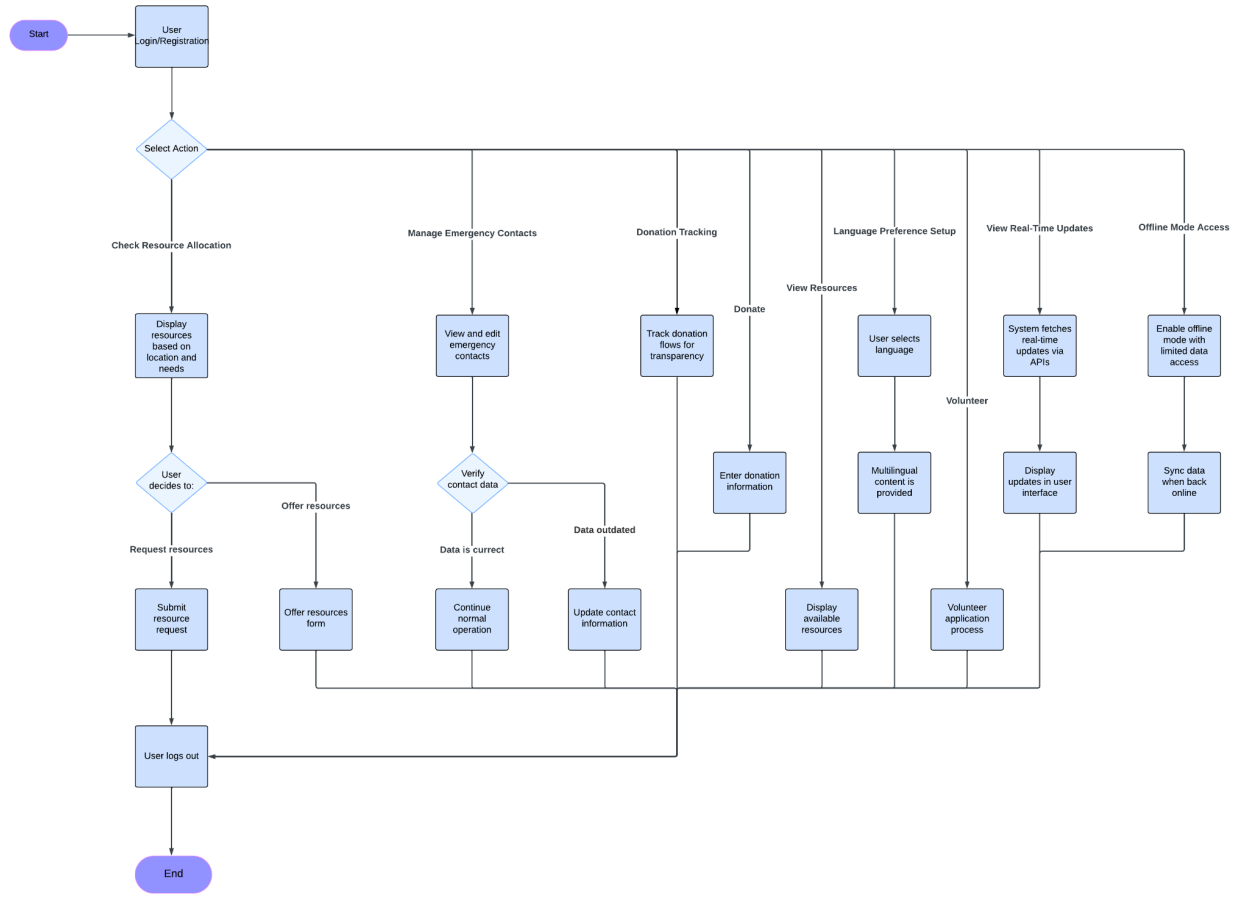
Sequence Diagram

Purpose: Shows the flow of communication and control between users, disaster sensors, and the system components.



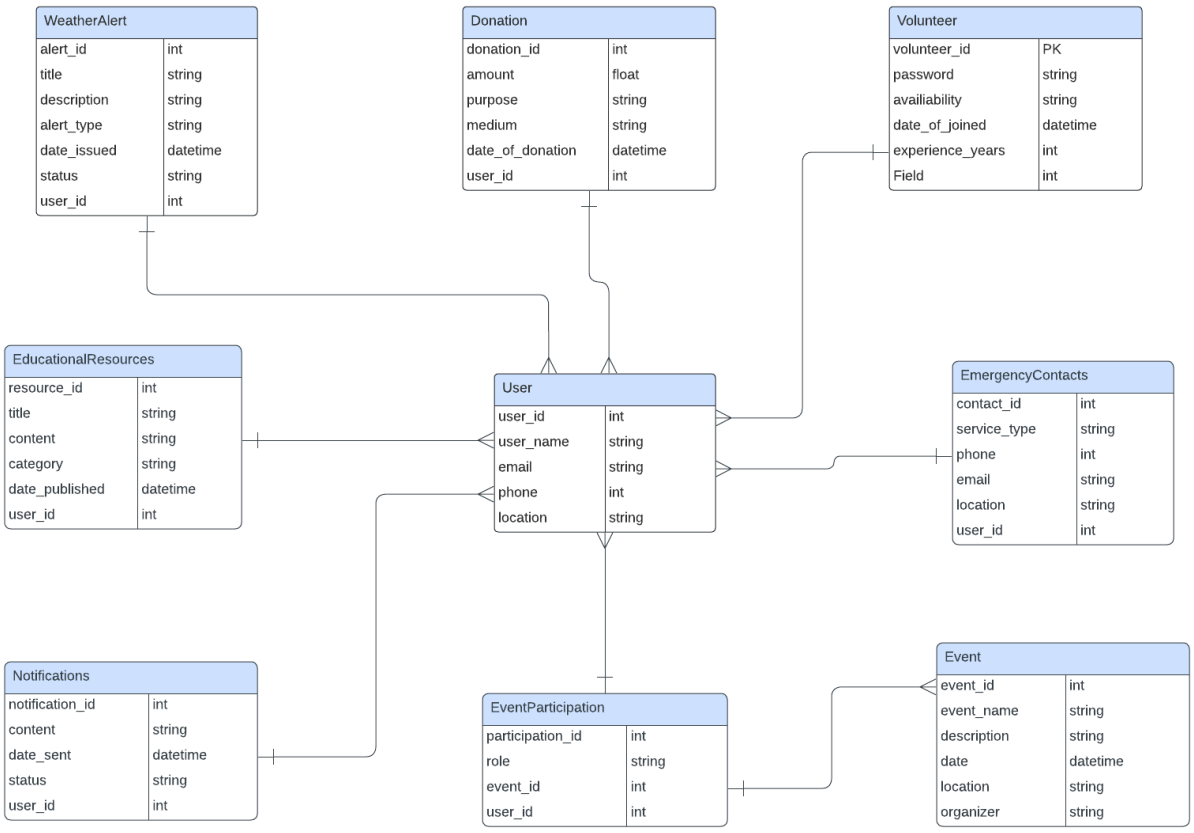
Activity Diagram

Purpose: Illustrates the workflow of disaster response processes, such as resource allocation and volunteer management.



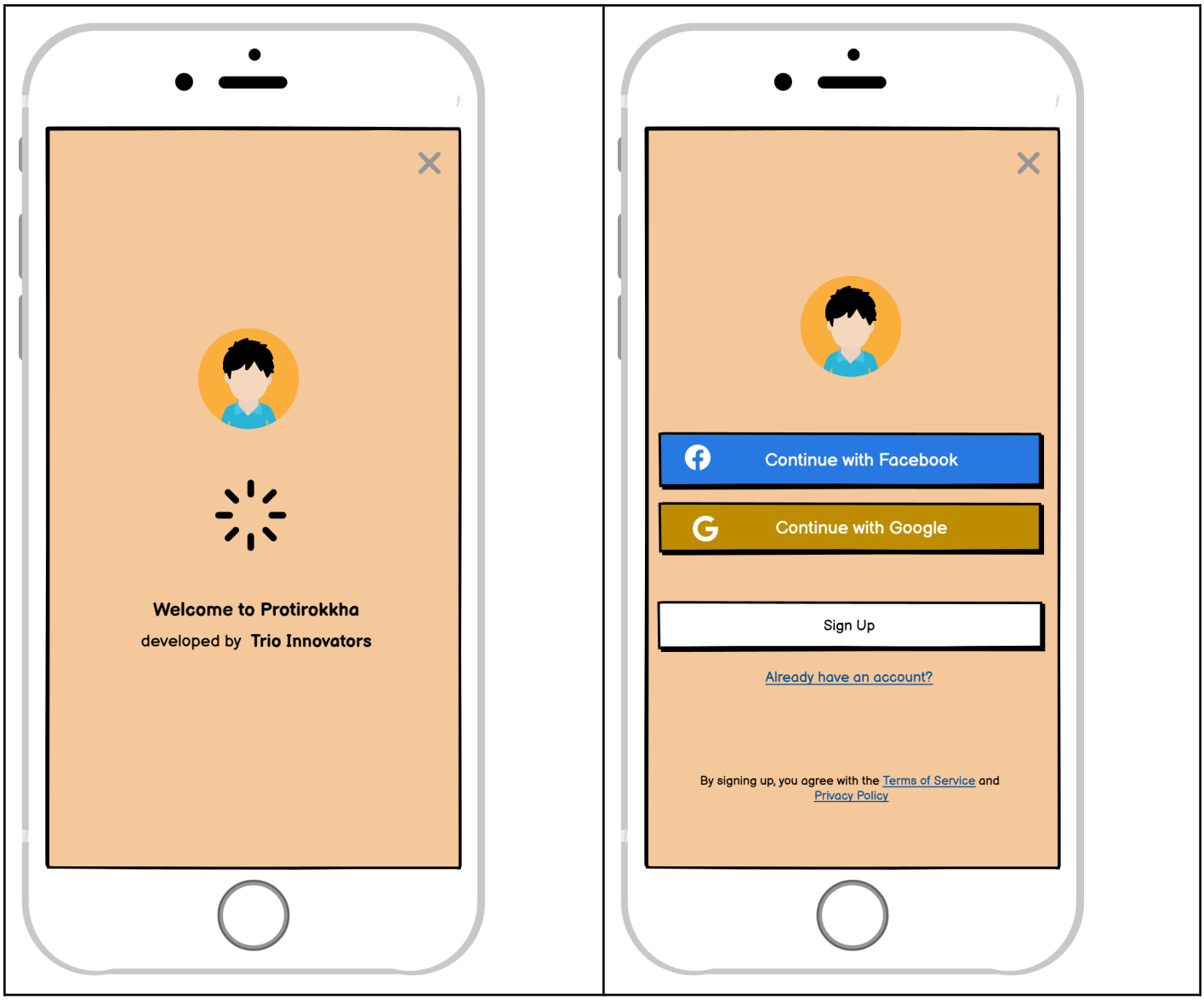
Class Diagram

Purpose: Organizes the system into structured components, defining attributes and methods for each system entity.

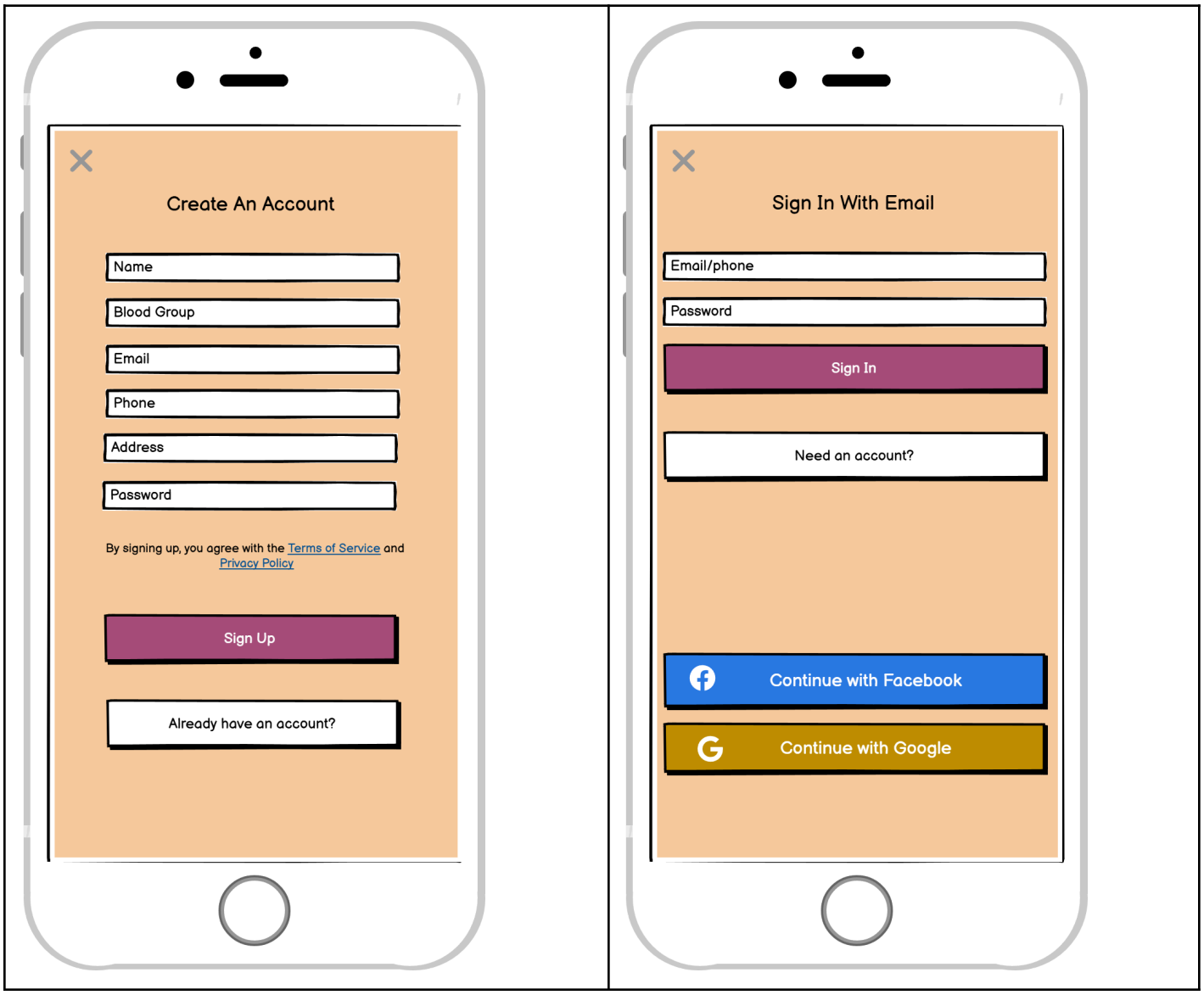


Prototype

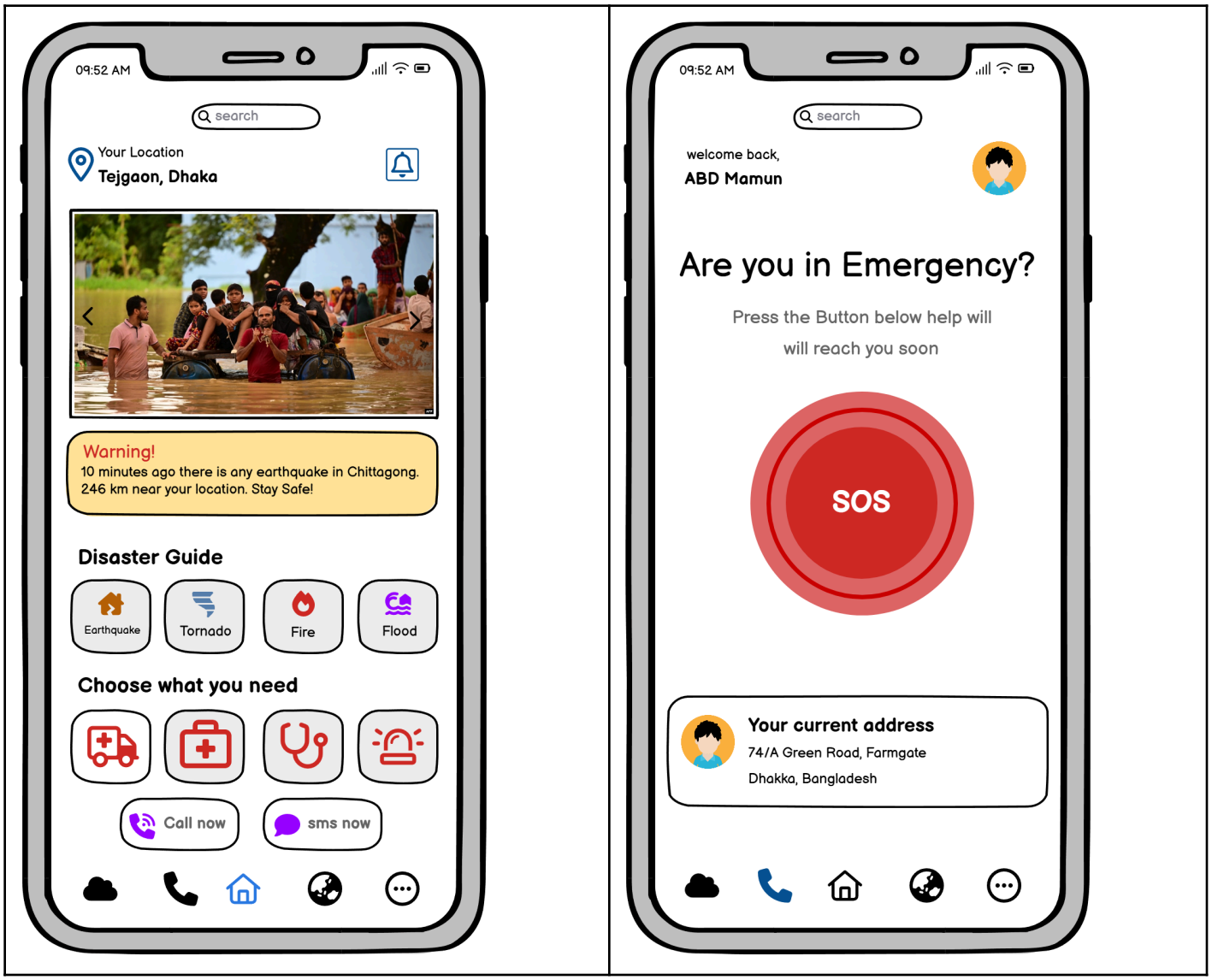
Loading Page



Login/Registration Page

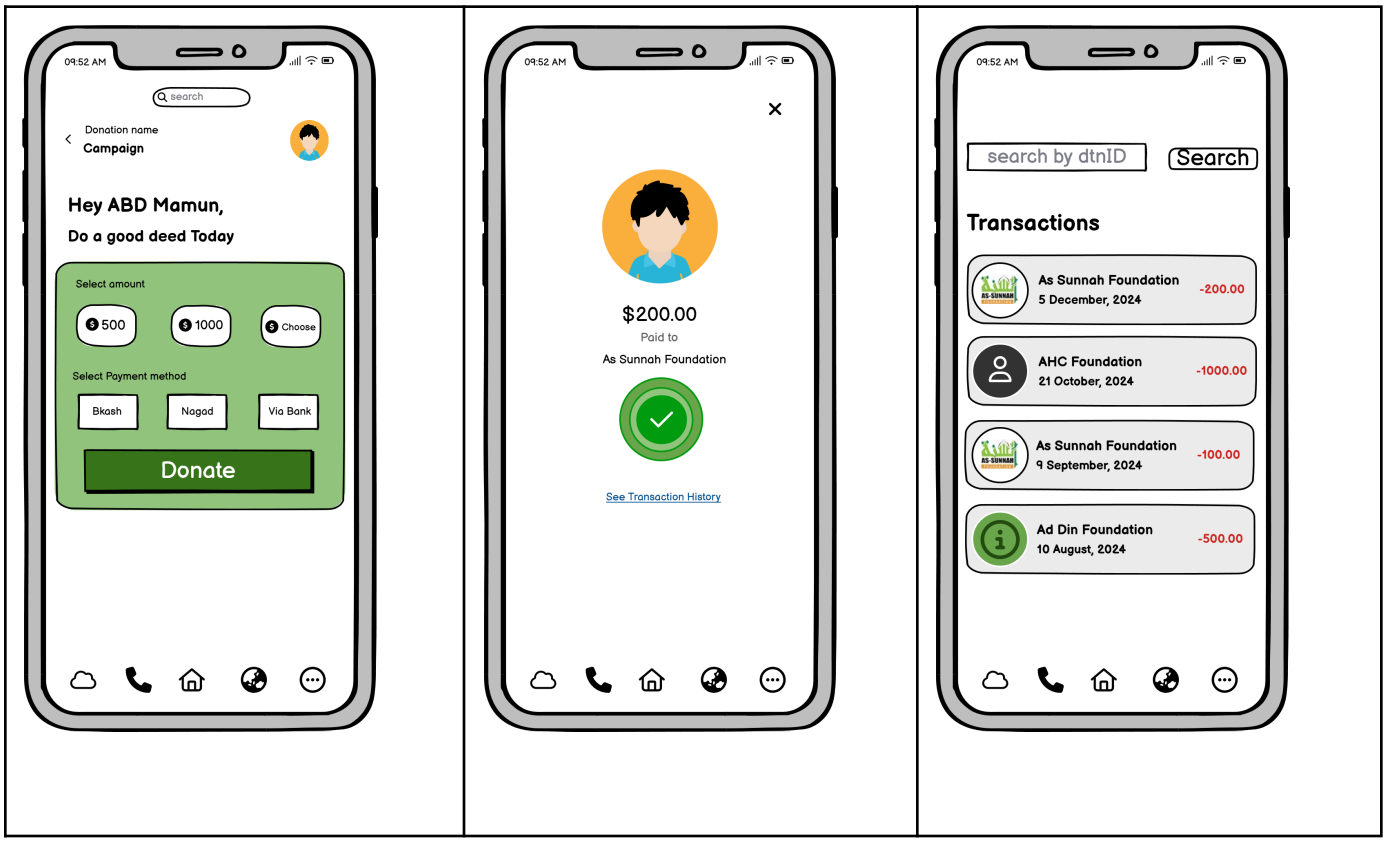


Home/Emergency Page



Weather Page

Transaction Page



CEP Mapping

How Knowledge Profiles (K's) are addressed through our project and mapping among K's, CO's, PO's:

K's	Attribute	How K's are addressed through our project	CO	PO
K4	Specialist knowledge	We need expertise in disaster management, real-time data processing, and AI to build predictive models for early warnings and risk assessment.	CO1, CO2	PO1
K5	Engineering Design	We designed the system architecture using ER diagrams, Data Flow Diagrams, and Use Case Diagrams to ensure scalability, accessibility, and efficiency in handling disaster data.	CO3, CO4	PO3, PO5
K6	Engineering Practice	We implemented the project using TensorFlow for machine learning, Python for backend processing, and cross-platform compatibility using modern tools like GitHub Copilot and VS Code.	CO4, CO9	PO4, PO5, PO6
K7	Comprehension	Our project benefits society by improving disaster preparedness, providing real-time updates, and enabling accurate resource allocation, thus minimizing loss of life and property.	CO5, CO6, CO7	PO7

How Complex Engineering Problems (P's) are addressed through our project and mapping among P's, CO's, PO's:

P's	Attribute	How P's are addressed through our project	CO	PO
P1	Depth of Knowledge Required	Requires in-depth engineering knowledge in disaster management, Machine Learning (K4), ER diagrams, Use Case, DFD (K5), Python programming (K6), and understanding the social implications of technology (K7) to develop a comprehensive preparedness portal.	CO1, CO2, CO4, CO5, CO7	PO1, PO2, PO7
P3	Depth of Analysis Required	This phase necessitates a detailed analysis of data sources, predictive algorithms, and real-time monitoring systems. Refining data models and integrating AI techniques (K4) improves accuracy, reliability, and decision-making for disaster response.	CO1, CO2, CO3, CO4, CO6	PO1, PO2, PO3, PO6
P7	Interdependence	The project involves multiple interconnected components, such as data collection, analysis, alert systems, and user interfaces. Addressing this interdependence requires a systems engineering approach for a robust disaster preparedness solution.	CO1, CO3, CO7	PO1, PO3, PO7

How Complex Engineering Activities (A's) are addressed through our project and mapping among A's, CO's, PO's:

A's	Attribute	How A's are addressed through our project	CO	PO
A4	Range of Resources	The project required diverse resources such as **Information**: Disaster Data, Emergency Protocols, Geographical and Weather Data, User Behavior. **Technologies**: Computers, Cloud Platforms, AI Tools, Real-Time Data APIs. **People**: Developers, NGOs, Volunteers, and Communities.	CO1, CO4	PO1, PO7
A4	Consequences for Society and the Environment	The project has significant societal and environmental impacts by enhancing disaster preparedness, saving lives, and reducing property damage. It promotes awareness and preparedness while minimizing environmental harm during resource deployment and disaster relief.	CO5, CO7	PO4, PO7

Conclusion for StormGuard Developers Project

The **StormGuard Developers Project** serves as a vital tool for enhancing disaster preparedness, response, and resilience. By centralizing critical information, providing real-time alerts, and fostering community collaboration, the portal empowers individuals and organizations to take proactive measures in mitigating disaster impacts. The platform's user-friendly design, inclusivity, and multi-platform compatibility ensure accessibility for diverse users, even during emergencies.

Through this initiative, we aim to bridge the gap between technological innovation and disaster management, enabling informed decision-making and resource optimization. As we continue to refine and expand the portal's capabilities, we envision a safer and more prepared society, capable of withstanding and recovering from the challenges posed by natural and man-made disasters.

Together, we can transform preparedness into resilience and ensure that no community faces disasters unprepared.

Contribution of Members to Project

Md Yasin Hossain Akash	Project Proposal, Prototype
Nusrat Ahmmed Maisha	Use Case Diagram, Gantt Chart, Presentation Slide
CM Hasibul Hasan	Class Diagram, Sequence Diagram, Activity Diagram

System Design and Prototype Overview

- Use Case Diagram:

https://www.canva.com/design/DAGT0ifafeU/rN5Wg0v_xLcxwWd09nUCbw/edit?utm_content=DAGT0ifafeU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

- Activity Diagram:

https://lucid.app/lucidchart/5920a3ba-c32a-46d9-9c98-ce0f01c11aea/edit?viewport_loc=-345%2C1936%2C5274%2C2461%2C0_0&invitationId=inv_e3aa5b68-cbcd-403b-a388-0dcd98358213

- Sequence Diagram:

https://www.canva.com/design/DAGXDJhKTkk/Y-znfcVd382SJp6Dny3zYA/edit?utm_content=DAGXDJhKTkk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

- Class Diagram:

https://lucid.app/lucidchart/634dd15c-c5d7-4b06-8626-c827388a4b40/edit?page=0_0&invitationId=inv_13f0b057-c464-40aa-a92f-969835a0cc66#

- Gantt Chart:

https://www.canva.com/design/DAGWnUTZVjY/YubM1XgdFyO1r6WcXD1ZHw/edit?utm_content=DAGWnUTZVjY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton