

Software Plan

1. Project Overview

Project Title: Eco-Drainage Monitoring Web App

Team Members: Richie Noel Ronde, Kier M. Borne, Neo Kent B. Durano

Date Started: October 6, 2025

Expected Completion: December 1, 2025

Short Description: The Eco-Drainage Monitoring Web App helps communities prevent flooding by reporting clogged drainage, tracking maintenance, and visualizing risk areas on an interactive map. It supports proactive action through alerts and updates, promoting safer and more sustainable cities.

2. Objectives

Main Goal: Develop a web application that enables communities to monitor drainage conditions, report issues, and support proactive flood prevention.

Specific Objectives:

- Provide a platform for residents to submit drainage reports with location data.
- Enable authorities to track, manage, and respond to drainage issues efficiently.

3. Scope

In-Scope Features:

- User authentication and role-based access (admin, community, staff).
- Interactive map with geotagged drainage reports and status tracking.

Out-of-Scope Features:

- Full-scale IoT sensor integration.
- Real-time weather forecasting system.

4. Stakeholders

Primary Users: Community residents and maintenance staff.

Other Stakeholders: Local government units, environmental NGOs, and urban planners.

5. Requirements (Simplified)

Functional Requirements:

- Users can submit drainage reports with descriptions and geotags.
- Admins can update status and assign maintenance tasks.

Non-Functional Requirements:

- The system should be mobile-friendly and accessible.
- The system should handle at least 500 concurrent users.

6. System Design (High-Level)

Architecture / Modules: Authentication, Drainage Reporting, Dashboards (Community & Admin), Map Integration, Alert System, Maintenance Tracking.

Technologies / Tools: Frontend: HTML, CSS, JavaScript

Backend: Node.js with Express.

Database: MySQL or MongoDB.

Maps API: Google Maps / OpenStreetMap.

Figma

Diagram (optional): _____

7. Project Timeline

Major Milestones:

- Week 1: Requirements gathering and database design.
- Week 2: Build authentication and reporting module.
- Week 3–4: Develop dashboards, map integration, alerts, and verification features
- Week 5: Testing, deployment, and documentation.

or

Gantt Chart / Task List : _____

8. Risks & Mitigation

Possible Risks:

- Low user adoption by community.
- Technical issues with map API or notifications.

Mitigation:

- Conduct community awareness and training.
- Implement fallback notification (email if SMS fails).

9. Testing & Quality Plan

What to Test:

- Correctness of drainage reporting and map display.
- Reliability of alerts and notifications.

How to Test:

- Unit and integration testing of each module.
- User acceptance testing with selected community members.

10. Deliverables

Expected Outputs:

- Fully functional Eco-Drainage Monitoring Web App.
- Project documentation, user guide, and final presentation/demo.