Logo

Description automatically generatedLogo

Description automatically generated

**FixIt Lanka**

**PROJECT PROPOSAL**

Proposed by

Harshana Dananjaya ∙ Rasindu Bingusara ∙ Purna Lankathilake ∙ Shanidu Hadunneththi ∙ Sandali Schokman

|  |  |
| --- | --- |
| General Project Information | |
| Project Name | FixIt Lanka |
| Duration | 5 weeks |
| Project Start-date | 3rd September 2024 |
| Project end-date | 8th October 2024 |
| Project Team | |
| Role | Name |
| Project Manager | Karagoda Patabendige Rasindu Savan Bingusara CL/HDCSE/CMU/112/02 |
| Front-End Developer | Poruthotage Harshana Dananjaya Dimithri Fernando CL/HDCSE/CMU/112/05 |
| Back-End Developer | Randage Adithya Purna Lankathilake Fernando  CL/HDCSE/CMU/112/30 |
| Quality Assurance | Handunneththi Shanidu De Zoysa CL/HDCSE/CMU/112/29 |
| System Analyst | Sandali Tharusha Schokman  CL/HDCSE/CMU/112/127 |

Contents

[1. Introduction 5](#_Toc176001689)

[2. Background Study 6](#_Toc176001690)

[2.1. Existing Solutions 6](#_Toc176001691)

[2.2. Concepts and Features Already Implemented 6](#_Toc176001692)

[1. User Registration and Login 6](#_Toc176001693)

[2. Post Creation and Categorization 7](#_Toc176001694)

[3. Anonymous Posting 7](#_Toc176001695)

[4. Community Interaction 7](#_Toc176001696)

[5. Location-Based Search and GPS Integration 7](#_Toc176001697)

[6. Heatmap Visualization 7](#_Toc176001698)

[7. Admin Oversight and Content Management 8](#_Toc176001699)

[3. Problem Statement / Description to the problem. 9](#_Toc176001700)

[1. Lack of Awareness of Public Issues 9](#_Toc176001701)

[2. Inefficient Reporting Channels 9](#_Toc176001702)

[3. Delayed Response to Urgent Issues 9](#_Toc176001703)

[4. Limited Community Engagement 10](#_Toc176001704)

[5. Lack of Transparency and Accountability 10](#_Toc176001705)

[6. Geographical and Situational Awareness Deficiency 10](#_Toc176001706)

[7. Inaccessibility for Vulnerable or Remote Communities 10](#_Toc176001707)

[8. Difficulty in Tracking and Managing Reports 11](#_Toc176001708)

[4. Literature review 12](#_Toc176001709)

[1. SeeClickFix 12](#_Toc176001710)

[2. FixMyStreet 13](#_Toc176001711)

[3. 311 Apps 13](#_Toc176001712)

[4. Citizen Reporting in Disaster Management 14](#_Toc176001713)

[5. Proposed Solution 16](#_Toc176001714)

[5.1. Comprehensive Public Issue Reporting Platform 16](#_Toc176001715)

[1. Raising Awareness of Public Issues 16](#_Toc176001716)

[2. Streamlined and Efficient Reporting Process 16](#_Toc176001717)

[3. Prioritizing Urgent Issues 16](#_Toc176001718)

[4. Enhancing Community Engagement 16](#_Toc176001719)

[5. Ensuring Transparency and Accountability 17](#_Toc176001720)

[6. Improving Geographical and Situational Awareness 17](#_Toc176001721)

[7. Facilitating Reporting from Vulnerable or Remote Communities 17](#_Toc176001722)

[8. Centralized Management and Tracking of Reports 17](#_Toc176001723)

[6. Justification to the solution 18](#_Toc176001724)

[1. Raising Awareness of Public Issues 18](#_Toc176001725)

[2. Streamlined and Efficient Reporting Process 18](#_Toc176001726)

[3. Prioritizing Urgent Issues 18](#_Toc176001727)

[4. Enhancing Community Engagement 18](#_Toc176001728)

[5. Ensuring Transparency and Accountability 19](#_Toc176001729)

[6. Improving Geographical and Situational Awareness 19](#_Toc176001730)

[7. Facilitating Reporting from Vulnerable or Remote Communities 19](#_Toc176001731)

[8. Centralized Management and Tracking of Reports 19](#_Toc176001732)

[7. Scope of the System 20](#_Toc176001733)

[7.1. Areas Covered by the System 20](#_Toc176001734)

[7.2. Justification for the Selected Scope 21](#_Toc176001735)

[7.3. Functionalities of the Proposed Solution 22](#_Toc176001736)

[7.3.1. Core Functionalities 22](#_Toc176001737)

[7.3.2. Additional Functionalities 24](#_Toc176001738)

[7.4. Non-Functional Requirements 25](#_Toc176001739)

[8. Methodology & Technology 26](#_Toc176001740)

[8.1. System Development Methodology: Agile 26](#_Toc176001741)

[8.2. Programming Language: Java 26](#_Toc176001742)

[8.3. Database Management System (DBMS): MySQL 26](#_Toc176001743)

[8.4. Software Tools: Android Studio and GitHub 27](#_Toc176001744)

[8.5. Hardware Requirements 27](#_Toc176001745)

[8.6. Team Structure 28](#_Toc176001746)

[8.7. Cost Estimation 31](#_Toc176001747)

[9. Gantt Chart / Risks 32](#_Toc176001748)

[9.1. Project Phases 32](#_Toc176001749)

[9.2. Project Risks 35](#_Toc176001750)

[9.3. Risk Metrix 38](#_Toc176001751)

[9.4. Risk Management Plan 39](#_Toc176001752)

[10. Conclusion 40](#_Toc176001753)

[11. References 41](#_Toc176001754)

[CSE5015 Computing Project Proposal Approval Sheet 42](#_Toc176001755)

# Introduction

In Sri Lanka, citizens often face various public issues ranging from infrastructure deficiencies to environmental concerns. However, these problems frequently go unnoticed or unresolved due to a lack of effective communication channels between the public and relevant authorities. To address this gap, our proposed mobile application provides a dynamic platform where Sri Lankans can report issues they encounter in their daily lives, whether in villages or cities.

This app is designed to empower users to take an active role in improving their communities by reporting issues such as road damage, water supply problems, and other public inconveniences. With features that include image uploads, detailed descriptions, issue categorization, and interactive options like commenting, voting, and sharing, the app fosters a community-driven approach to problem-solving. Users can also explore issues in their vicinity using GPS-based location tracking, making it easier to identify and address local problems.

Moreover, the app includes an anonymous posting option, ensuring users can report sensitive issues without fear of reprisal. The integration of admin oversight ensures that all posts adhere to community guidelines, maintaining a safe and constructive environment for all users.

By leveraging modern technology, this app aims to bridge the communication gap between citizens and authorities, making it easier to highlight and resolve the issues that affect daily life in Sri Lanka. Through this platform, we envision a more engaged and proactive citizenry, leading to more responsive governance and improved public services.

# Background Study

In recent years, Sri Lanka has seen a growing demand for effective communication channels between citizens and authorities to address public issues such as infrastructure damage, environmental concerns, and other community-related problems. Traditional methods of reporting issues, such as through government offices or local councils, have often proven inefficient, leading to delays in addressing critical public concerns. In response to this need, mobile applications have emerged as a powerful tool for bridging the communication gap between the public and governing bodies.

## Existing Solutions

Globally, there are several mobile applications and platforms that have successfully implemented the concept of crowd-sourced issue reporting. Applications such as “SeeClickFix” in the United States and “FixMyStreet” in the United Kingdom have empowered citizens to report non-emergency issues directly to local governments, facilitating faster resolution times. These platforms typically allow users to report issues by uploading images, providing descriptions, and selecting relevant categories. The issues are then mapped and made visible to other users and local authorities, creating a transparent and accountable system for managing public complaints.

In Sri Lanka, however, there is still a significant gap in the adoption of such technologies. The need for a localized solution tailored to the specific challenges and needs of Sri Lankan citizens has become increasingly apparent. While social media platforms like Facebook and Twitter are sometimes used to highlight public issues, these platforms lack the structured approach needed for effective issue tracking, categorization, and resolution.

## Concepts and Features Already Implemented

### User Registration and Login

* + The app provides a straightforward registration process where users can sign up using their basic details—first name, last name, email, phone number, and profile picture. To ensure user authenticity, the app implements an OTP (One-Time Password) system for phone number verification. This feature is crucial in establishing a secure and trustworthy user base.

### Post Creation and Categorization

* + Users can create posts to report issues by uploading images and providing detailed descriptions. The ability to categorize these issues (e.g., Infrastructure, Environmental, Public Services) allows for better organization and easier navigation for both users and authorities. For instance, user’s report on the malfunctioning streetlights is categorized under “Infrastructure,” making it easy for others to find and relate to similar issues.

### Anonymous Posting

* + Recognizing the need for privacy, the app allows users to post anonymously. This feature ensures that users can report sensitive issues without fear of reprisal, encouraging more active participation. All anonymous posts undergo admin review before being published to maintain the integrity of the platform.

### Community Interaction

* + The app fosters community engagement by enabling users to comment on, vote for, and share posts. These interactions help prioritize issues based on community concern, as seen when users upvote and share one of users post about streetlights. This feature also creates a sense of collective responsibility, motivating users to participate actively in addressing community issues.

### Location-Based Search and GPS Integration

* + The app leverages GPS technology to provide users with location-specific posts. Users can search for issues in nearby areas, filter them by district or province, and sort by the number of votes or recent dates. This feature enhances the app’s usability by allowing users to focus on issues that are most relevant to their locality.

### Heatmap Visualization

* + A heatmap feature provides a visual representation of the most problematic areas based on the concentration of reported issues. This tool helps both users and authorities identify and prioritize regions that require urgent attention. user’s exploration of this feature reveals districts with a high number of reported problems, highlighting areas that may need more focused governmental intervention.

### Admin Oversight and Content Management

* + To ensure that the platform remains free from misuse, the app includes an admin role. Admin reviews all posts marked for anonymous publication and monitor flagged posts and comments. This oversight helps maintain the quality and reliability of the content on the app while ensuring that community guidelines are upheld. The admin role also includes user management, which is essential in preventing the spread of false information and maintaining the app’s credibility.

# Problem Statement / Description to the problem.

The public issue reporting app we're developing is designed to address several key problems in communities by providing a platform for citizens to report issues and enabling authorities or relevant organizations to respond more effectively. Here’s a clear explanation of the problems this app would solve:

## 1. Lack of Awareness of Public Issues

* **Problem**: Often, authorities and the broader community are unaware of local issues such as broken infrastructure, environmental hazards, or safety concerns because these problems are not reported or are reported through inefficient channels.
* **Solution**: The app allows users to report issues as soon as they encounter them. By capturing photos or videos and providing descriptions, citizens can quickly bring attention to problems that need to be addressed. The platform makes it easy for these reports to be shared with both the public and relevant authorities.

## 2. Inefficient Reporting Channels

* **Problem**: Traditional methods of reporting issues (e.g., calling government offices, filling out forms, or sending emails) are often time-consuming, inconvenient, and lack transparency.
* **Solution**: The app streamlines the reporting process, enabling users to submit reports directly from their smartphones. The structured format ensures that all necessary information (location, category, description) is provided, making it easier for authorities to understand and act on the issues.

## 3. Delayed Response to Urgent Issues

* **Problem**: Without a centralized system, urgent issues may go unnoticed or unresolved for long periods, leading to further deterioration or harm.
* **Solution**: The app prioritizes issues based on urgency, trending status, and user votes. This feature ensures that critical problems receive immediate attention, allowing authorities or relevant organizations to allocate resources efficiently.

## 4. Limited Community Engagement

* **Problem**: Citizens often feel disconnected from local governance and may not believe their concerns will be heard or acted upon.
* **Solution**: The app fosters community engagement by allowing users to vote on the importance of issues, comment on reports, and share posts within their networks. This interaction encourages collective action and gives citizens a sense of involvement in the well-being of their community.

## 5. Lack of Transparency and Accountability

* **Problem**: When issues are reported through traditional channels, there’s often little transparency in how they are handled, and citizens may not receive updates on the status of their reports.
* **Solution**: The app provides real-time updates on the status of reports, from submission to resolution. This transparency builds trust and ensures that authorities or responsible parties are held accountable for addressing the issues reported.

## 6. Geographical and Situational Awareness Deficiency

* **Problem**: Authorities may struggle to identify problem areas geographically, making it difficult to allocate resources effectively or understand the broader impact of certain issues.
* **Solution**: The app’s integration with GPS and mapping services allows users to pinpoint the exact location of issues. It also aggregates data to highlight problem areas, helping authorities identify hotspots and address recurring issues more strategically.

## 7. Inaccessibility for Vulnerable or Remote Communities

* **Problem**: Vulnerable or remote communities often face difficulties in reporting issues due to isolation, lack of resources, or technological barriers.
* **Solution**: By leveraging mobile technology, the app provides a convenient reporting tool that can be used even in remote areas. The ability to post anonymously also protects users who may fear retaliation for reporting certain issues, thereby encouraging broader participation.

## 8. Difficulty in Tracking and Managing Reports

* **Problem**: Authorities and organizations may struggle to track multiple reports, especially when they come through different channels or lack consistent formats.
* **Solution**: The app centralizes all reports in a single system, where they can be categorized, tracked, and managed efficiently. The admin panel allows for easy review, prioritization, and action on reports, improving overall response times and management of public issues.

# Literature review

Reviewing similar systems and technologies used in public issue reporting apps can provide valuable insights into best practices and potential challenges. Below are a few well-known systems and the technologies they use, along with a review of their features and relevance to your project.

## 1. SeeClickFix

Overview: SeeClickFix is a widely-used civic engagement platform that allows residents to report non-emergency issues (like potholes or broken streetlights) directly to their local government. It’s used in various cities in the U.S. and internationally.

**Key Features**

• Issue Reporting: Allows users to report issues by taking photos, adding descriptions, and categorizing them.

• Location-Based: Uses GPS to tag the location of the issue.

• Real-Time Notifications: Users receive updates when their issue is acknowledged or resolved.

• Community Interaction: Other users can comment on or vote for reported issues.

• Integration with Government Systems: Directly integrates with municipal work order systems for efficient resolution.

**Technology Stack**

• Frontend: React.js and mobile apps using native development (Swift for iOS, Kotlin/Java for Android).

• Backend: Node.js, Ruby on Rails, and Python for various services.

• Database: PostgreSQL and MongoDB for storing user data, reports, and interactions.

• Mapping: Google Maps API or OpenStreetMap for geolocation and map rendering.

Relevance to Your Project: SeeClickFix is a close match for your app’s intended functionality, especially in terms of issue reporting, user interaction, and location-based features. The technology stack used in SeeClickFix is modern and scalable, making it a good reference point for your app.

## 2. FixMyStreet

Overview: FixMyStreet is a UK-based platform that allows citizens to report problems like potholes, broken streetlights, and other local issues directly to their local authorities. It has also been adapted for use in other countries.

**Key Features**

• Simple Reporting: Users can report issues by specifying a location on the map and providing a description.

• Anonymity: Offers the option for users to report issues anonymously.

• Integration: Reports are automatically sent to the relevant local authority.

• Feedback Loop: Users can see updates on the status of their reports.

**Technology Stack**

• Frontend: Ruby on Rails for the web platform, with a mobile-responsive design.

• Backend: Ruby on Rails and PostgreSQL for managing user data and reports.

• Mapping: Uses OpenStreetMap and Mapbox for location services.

Relevance to Your Project: FixMyStreet’s focus on simplicity and user experience could be beneficial in designing your app’s interface. The use of open-source mapping solutions like OpenStreetMap and Mapbox might also be an effective alternative to Google Maps, depending on budget and needs.

## 3. 311 Apps

Overview: Many U.S. cities have developed their own 311 apps, which serve as non-emergency contact points for residents. These apps allow users to report issues, request services, and get information.

**Key Features**

• Multi-Functional: Apart from reporting issues, these apps often provide information on city services and allow users to submit service requests.

• Real-Time Updates: Users can track the status of their requests in real-time.

• Personalized Experience: Users can create accounts to track their interactions with the city over time.

• Push Notifications: Alerts users about the status of their reports and important city updates.

**Technology Stack**

• Frontend: Often built with React Native for cross-platform mobile apps.

• Backend: Typically relies on city-specific backend systems, often involving RESTful APIs connected to municipal databases.

• Database: Relational databases like PostgreSQL, with some using NoSQL for scalability.

• Mapping: Google Maps API or ArcGIS for geographic information.

Relevance to Your Project: 311 apps are a good example of integrating issue reporting with other civic services, providing a more comprehensive tool for residents. The use of cross-platform mobile development frameworks like React Native could be advantageous if you’re aiming for both Android and iOS support.

## 4. Citizen Reporting in Disaster Management

Overview: During disasters, apps like these allow citizens to report incidents, request help, and provide information to authorities in real-time. Examples include Ushahidi, which was initially developed for crisis reporting in Kenya.

**Key Features**

• Crowdsourcing: Gathers reports from multiple users to create a comprehensive view of the situation.

• Real-Time Mapping: Displays reports on a map in real-time, helping authorities identify hotspots.

• Multimedia Reports: Allows users to submit photos, videos, and text descriptions.

• Data Analysis: Aggregates data to provide insights into trends and help prioritize responses.

**Technology Stack**

• Frontend: Often built with a mix of web technologies (HTML, CSS, JavaScript) and mobile frameworks.

• Backend: Python or Ruby on Rails for processing reports and integrating with other systems.

• Database: MySQL, PostgreSQL, or MongoDB for handling large volumes of data.

• Mapping: Uses tools like Leaflet, Google Maps API, or custom GIS solutions.

Relevance to Your Project: The use of crowdsourcing and real-time mapping in disaster management apps aligns with your need to highlight problem areas and prioritize issues. Ushahidi’s open-source model might also be an interesting framework to explore if you’re looking for customizable solutions.

# Proposed Solution

## Comprehensive Public Issue Reporting Platform

Our app provides a robust solution to several pressing problems in communities by creating an accessible, transparent, and efficient system for reporting, managing, and resolving public issues. Here’s how your solution addresses each problem:

### 1. Raising Awareness of Public Issues

* The app empowers citizens to report issues they encounter in their daily lives directly from their smartphones. By allowing users to upload photos, videos, and descriptions, the app ensures that these issues are visible not only to local authorities but also to the broader community. The categorization feature makes it easy to classify and understand the nature of each issue, whether it’s a broken road, a public safety concern, or environmental degradation.

### 2. Streamlined and Efficient Reporting Process

* The app simplifies the process of reporting issues by offering a user-friendly interface where reports can be submitted quickly and easily. With just a few taps, users can take a photo or video, select the appropriate category, add a description, and submit the report. This streamlined process replaces more cumbersome traditional methods (such as calling or filling out forms) and ensures that reports contain all the necessary information for authorities to act swiftly.

### 3. Prioritizing Urgent Issues

* The app includes features that allow users to vote on the importance of reports, which helps prioritize issues based on community feedback. The “trending” status and filters by urgency mean that authorities can quickly identify which issues need immediate attention. This prioritization mechanism ensures that the most critical problems are addressed first, reducing the risk of issues escalating due to delays.

### 4. Enhancing Community Engagement

* To foster a sense of community involvement, the app allows users to comment on reports, vote on their significance, and share them within their networks. By enabling this level of interaction, the app encourages citizens to take an active role in improving their communities. The ability to save reports and share them on a personal profile wall also gives users a sense of ownership and pride in their contributions.

### 5. Ensuring Transparency and Accountability

* The app’s built-in tracking and notification features ensure that users are kept informed about the status of their reports. They can see when a report has been acknowledged, reviewed, and resolved. This transparency builds trust between the community and authorities, and the clear documentation of issues and their resolution process holds authorities accountable for taking action.

### 6. Improving Geographical and Situational Awareness

* By integrating GPS and mapping services, the app allows users to precisely locate the issues they are reporting. This geographical data helps authorities visualize problem areas and allocate resources more effectively. The ability to sort and filter reports by location, urgency, and other criteria provides a comprehensive view of the situation, enabling better decision-making and targeted interventions.

### 7. Facilitating Reporting from Vulnerable or Remote Communities

* The app’s mobile-first approach ensures that even those in remote or underrepresented areas can report issues easily. The anonymous reporting option further encourages participation from individuals who might otherwise hesitate to report due to fear of retaliation or stigma. By making the reporting process accessible to all, the app ensures that the voices of vulnerable communities are heard.

### 8. Centralized Management and Tracking of Reports

* For authorities and administrators, the app offers a centralized system for managing all reports. The admin panel provides tools for reviewing, categorizing, and tracking reports, making it easier to manage multiple issues simultaneously. This centralization improves the efficiency of responding to reports, ensuring that no issue falls through the cracks.

# Justification to the solution

The proposed solution effectively mitigates or overcomes the identified problems by leveraging technology to enhance communication, transparency, and community engagement. Here's how each feature of the app justifies its role in addressing the specific problems:

## 1. Raising Awareness of Public Issues

* By providing a direct and accessible platform for reporting issues, the app ensures that problems which might otherwise go unnoticed are brought to the attention of both authorities and the community. The ability to capture and share visual evidence (photos, videos) with detailed descriptions increases the likelihood that these issues will be recognized and addressed promptly. The app transforms passive observations into actionable reports, raising awareness at multiple levels.

## 2. Streamlined and Efficient Reporting Process

* The app’s user-friendly interface simplifies the process of reporting, eliminating the need for citizens to navigate bureaucratic or outdated channels. This ease of use encourages more frequent reporting, ensuring that issues are documented as they arise. By automating much of the data collection (e.g., GPS location tagging), the app reduces the potential for incomplete or inaccurate reports, making it easier for authorities to respond effectively.

## 3. Prioritizing Urgent Issues

* The inclusion of a voting system and urgency filters ensures that the most critical issues are highlighted and addressed first. This feature overcomes the problem of delayed responses by allowing both the community and authorities to focus on the most pressing issues. It also empowers users to influence the prioritization process, ensuring that community concerns are directly reflected in the handling of reports.

## 4. Enhancing Community Engagement

* Community engagement is bolstered through features that allow users to comment, vote, and share reports. These interactive elements not only create a sense of ownership among users but also foster a collective approach to problem-solving. By encouraging dialogue and collaboration, the app strengthens community ties and creates a more proactive environment for addressing public issues. It mitigates the problem of citizen apathy by providing a tangible way for individuals to contribute to their community.

## 5. Ensuring Transparency and Accountability

* The app’s tracking and notification system ensures that users are informed about the status of their reports, addressing the problem of opaque processes and unaccountable authorities. By providing clear, real-time updates on the progress of each report, the app builds trust between the public and the authorities. This transparency is crucial for maintaining confidence in the system and ensuring that issues are not only reported but also resolved.

## 6. Improving Geographical and Situational Awareness

* The integration of GPS and mapping services directly addresses the problem of geographical awareness by allowing issues to be reported with precise location data. This feature enables authorities to visualize problem areas and allocate resources more effectively, ensuring that responses are targeted and efficient. The ability to filter and sort reports by location further enhances situational awareness, making it easier to identify patterns and address recurring issues.

## 7. Facilitating Reporting from Vulnerable or Remote Communities

* By making the app accessible to anyone with a smartphone, and including options for anonymous reporting, the solution ensures that even those in remote or vulnerable communities can report issues without fear or difficulty. This inclusivity addresses the problem of underreporting in isolated areas, ensuring that all voices are heard and all problems are documented. The app breaks down barriers to participation, making it easier for marginalized groups to contribute to public discourse.

## 8. Centralized Management and Tracking of Reports

* The app’s centralized system for managing reports directly addresses the challenge of tracking and responding to multiple issues. By consolidating all reports in one place, the app simplifies the management process for authorities, allowing for more organized and timely responses. This centralization also helps avoid duplication of efforts and ensures that all issues are followed up on, enhancing overall efficiency and effectiveness in addressing public concerns.

# Scope of the System

The proposed public issue reporting app is designed to cover a wide range of areas that are critical for community safety, infrastructure maintenance, and public welfare. The app serves as a comprehensive platform for reporting, managing, and resolving public issues, with functionalities that ensure user engagement, efficient problem resolution, and transparency. The scope includes both functional and non-functional aspects that are essential to its success.

## Areas Covered by the System

|  |  |  |
| --- | --- | --- |
| **Areas covered** | **Scope** | **Justification** |
| **Public Safety** | Reports related to criminal activities, safety hazards, and emergencies (e.g., robberies, criminal activities, hazardous infrastructure). | Ensuring public safety is a fundamental concern for any community. By enabling quick and easy reporting of safety-related issues, the app helps authorities respond more effectively to emergencies and threats. |
| **Infrastructure and Maintenance** | Issues related to public infrastructure, such as broken roads, damaged public property, or areas requiring reconstruction. | Well-maintained infrastructure is crucial for community functionality and quality of life. The app allows citizens to report infrastructure |
| **Environmental and Community Welfare** | Environmental concerns like areas needing water supply, protection from wildlife, or network coverage issues. | Addressing environmental and community welfare issues is essential for sustainable development. By covering these areas, the app helps ensure that communities receive the resources and attention they need to thrive. |
| **Public Services and Facilities** | Issues related to public services, such as schools needing financial help, or lack of essential amenities in certain areas. | The app supports the improvement of public services by allowing users to highlight areas in need of attention, ensuring that essential services are available to all community members. |
| **Community Engagement and Transparency** | Facilitating community interaction through comments, voting, and sharing of reports, as well as providing transparency in how reports are handled. | Engaging the community in problem-solving processes and maintaining transparency are key to fostering trust and collective responsibility. The app’s focus on these areas encourages active participation and holds authorities accountable. |

## Justification for the Selected Scope

The selected scope is justified by the need to address a broad spectrum of public issues that affect community safety, infrastructure, and well-being. By covering these areas, the app ensures that the most critical aspects of community life are monitored and maintained, leading to a safer, more functional, and engaged society. The focus on public safety, infrastructure, environmental welfare, and community engagement reflects the app's goal of creating a holistic solution that addresses both immediate concerns and long-term community development.

## Functionalities of the Proposed Solution

The app includes a wide range of functionalities designed to meet the needs of both users and administrators. These functionalities are essential for the effective operation of the system.

### Core Functionalities

|  |  |  |
| --- | --- | --- |
| **Core Functionalities** | **Functionality** | **Purpose** |
| User Authentication | Users can sign up and log in using their email and phone number (with OTP verification). | Ensures that only legitimate users can access the app, protecting against misuse. |
| **Issue Reporting** | Users can create posts to report public issues, including uploading images/videos, selecting categories, and adding descriptions. | Provides a structured and straightforward way for users to report issues, ensuring that all necessary information is captured. |
| **Vote and Comment System** | Users can vote on the importance of reports and add comments. | Facilitates community engagement and helps prioritize issues based on collective feedback. |
| Trending and Recent Posts | Displays the most trending and recently viewed posts on the home screen. | Helps users stay informed about the most significant and recent issues in their community. |
| Sharing and Saving Reports | Users can share reports on social media and save them to their profiles. | Increases the visibility of reports and allows users to keep track of issues that matter to them. |
| Admin Review and Management | Administrators can view user profiles, even in anonymous mode, and manage reports flagged for review. | Ensures that all reports are reviewed for appropriateness and that user-generated content is managed responsibly. |

### Additional Functionalities

|  |  |  |
| --- | --- | --- |
| **Additional Functionalities** | **Functionality** | **Purpose** |
| **Map Integration** | Users can search for reports by location, sort them by date, district, province, or vote count, and see reports from their area using GPS. | Enhances geographical awareness and allows users to focus on issues relevant to their specific location. |
| Anonymous Reporting | Users can choose to report issues anonymously. | Protects users who may fear retaliation, encouraging more people to report issues. |
| Profile Management | Users can manage their profiles, including their name, email, phone number, and profile picture. | Allows users to personalize their experience and manage their identity within the app. |
| Sorting and Filtering | Users can sort and filter reports by various criteria such as urgency, location, and category. | Helps users quickly find the reports that are most relevant to them. |

## Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirement** | **Requirement** | **Justification** |
| **Scalability** | The app must handle a large number of users and reports without performance degradation. | As the app grows in popularity, it must remain responsive and efficient, ensuring a smooth user experience. |
| **Security** | The app must protect user data, especially given the option for anonymous reporting, and ensure secure communication between users and the server. | Security is paramount to protect users’ privacy and data, especially when reporting sensitive issues. |
| **Usability** | The app must be easy to use, with a simple and intuitive interface. | High usability ensures that users from all backgrounds can easily navigate the app and report issues without difficulty. |
| **Reliability** | The app must be available and functioning correctly at all times. | Reliability is crucial for maintaining user trust and ensuring that reports can be made and accessed whenever necessary. |
| **Performance** | The app must load quickly and respond to user inputs without noticeable lag. | High performance is essential for a positive user experience, particularly when dealing with urgent reports. |
| **Localization** | The app should support multiple languages and localize content based on user location. | Localization ensures that the app is accessible to users in different regions and enhances usability across diverse linguistic groups. |
| **Maintainability** | The app’s codebase must be maintainable, with clear documentation and modular design. | Maintainability is important for future updates, bug fixes, and enhancements to ensure the app remains functional and up-to-date. |

# Methodology & Technology

## 8.1. System Development Methodology: Agile

* Chosen Methodology: Agile

Agile methodology was chosen due to its iterative and flexible approach, allowing for continuous feedback and improvements throughout the project lifecycle. This is particularly beneficial for expanding the project in the future, as Agile supports incremental enhancements, making it easier to adapt to new requirements or changes.

## 8.2. Programming Language: Java

* Chosen Language: Java

Java is a robust, platform-independent language that is well-suited for developing scalable and secure applications. It has a strong ecosystem, with extensive libraries and frameworks that streamline the development process. Java's object-oriented features also make it ideal for creating complex systems with maintainable code, which is essential for future expansions.

## 8.3. Database Management System (DBMS): MySQL

* Chosen DBMS: MySQL

MySQL hosted on Amazon Web Services (AWS) is an ideal choice for this project due to its robust and scalable nature, which is crucial for handling the potentially large and growing dataset generated by user interactions, posts, and comments. MySQL is a well-established relational database management system (RDBMS) that offers strong consistency, reliability, and support for complex queries, making it suitable for applications that require structured data storage with relational integrity.

AWS provides a highly available and secure environment for hosting MySQL, ensuring that the app can handle high traffic volumes and scale as needed. AWS's global infrastructure ensures low latency and quick access for users across Sri Lanka, enhancing the user experience. Additionally, MySQL's compatibility with a wide range of programming languages and its extensive community support makes it a reliable choice for developers, while AWS's managed services reduce the overhead of database maintenance, allowing the team to focus on developing and improving the app.

## 8.4. Software Tools: Android Studio and GitHub

* Chosen Tools:
  + Android Studio: The official IDE for Android development, offering powerful tools for building, testing, and debugging Android applications. It supports a wide range of features that enhance productivity, including a visual layout editor, a robust code editor, and integration MySQL.
  + GitHub: A widely used version control platform that facilitates collaboration among team members. GitHub's features, such as branching, pull requests, and issue tracking, make it easy to manage code changes and ensure that the project is well-organized and documented.

## 8.5. Hardware Requirements

|  |  |
| --- | --- |
| **Requirement** | **Specifications** |
| **Development Machines** | * High-performance laptops or desktop computers with at least 16GB of RAM. * SSD storage with a minimum of 128GB. * Multi-core processors (e.g., Intel i5). * Purpose: To handle the demands of Android development, including coding, testing, and debugging. |
| **Mobile Devices for Testing** | * A range of Android smartphones and tablets with varying screen sizes and OS versions. * Devices should include both high-end and mid-range models. * Purpose: To ensure the application performs well across different devices and screen resolutions, verifying compatibility and user experience. |
| Server Infrastructure | * Cloud-based servers with scalable resources based on project needs (e.g., AWS, Google Cloud, or Azure). * Sufficient processing power, memory, and storage to support Firebase and continuous integration/deployment (CI/CD) processes. * Purpose: To host the backend services, manage data, and support development operations. |

## 8.6. Team Structure

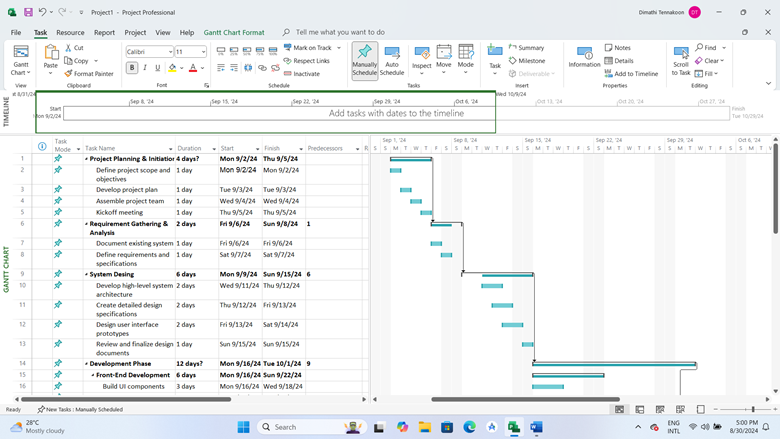
|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Key Responsibilities** |
| Rasindu Bingusara | **Project Manager**   * Oversees the entire project, ensuring alignment with goals and timelines. | * Manage project planning, scheduling, and resource allocation.   + Coordinate communication among stakeholders and the team.   + Monitor project progress and implement risk management strategies.   + Ensure the project stays within budget and meets business objectives. |
| Harshana Dananjaya | **Front-End Developer**   * Develop and implement the user interface (UI) of the application. | * + Design responsive and visually appealing UI for different devices.   + Integrate front-end components with back-end services.   + Conduct UI testing to ensure usability and performance.   + Implement interactive features like animations and transitions. |
| Purna Lankathilake | **Back-End Developer**   * + Handle server-side development, database management, and API integration. | * + Develop and maintain server logic and database interactions using Firebase.   + Ensure secure and efficient data handling between the server and client.   + Integrate third-party services and APIs as needed.   + Optimize the application for scalability and performance. |
| Sandali Schokman | **System Analyst**   * + Analyze system requirements and ensure alignment with project objectives. | * + Gather and translate user requirements into technical specifications.   + Conduct feasibility studies and analyze system design.   + Liaise between stakeholders and the development team to ensure clarity.   + Ensure that the system meets both functional and non-functional requirements. |
| Shanidu Handunneththi | Quality Assurance   * + Ensure the application meets quality standards before deployment. | * + Develop and execute comprehensive test plans and cases.   + Perform functional, regression, and performance testing.   + Identify, document, and verify the resolution of bugs and issues.   + Collaborate with developers to ensure quality and functionality. |

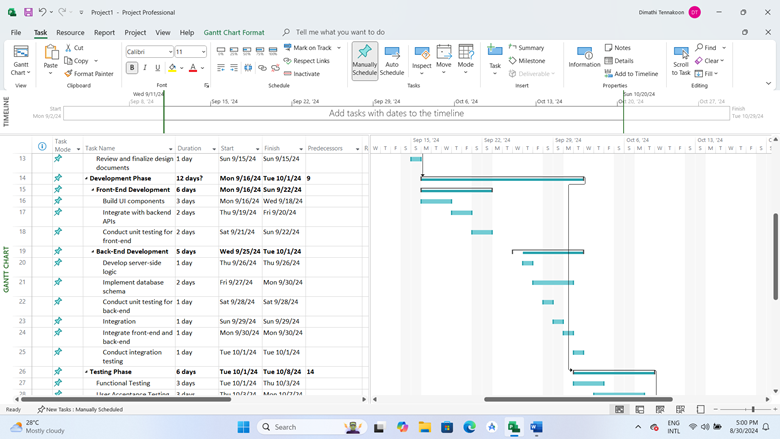
## 8.7. Cost Estimation

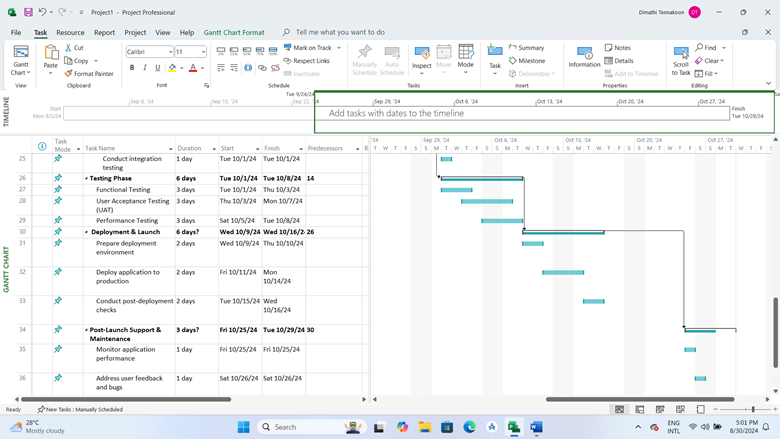
|  |  |  |
| --- | --- | --- |
| **Cost Item** | **Description** | **Cost** |
| Software Licensing | Android Studio | Free, open-source. |
| GitHub | Free for public repositories; cost incurred for private repositories depending on team size. |
| Amazon Web Service Cloudhost Database MySQL | Free tier available; costs scale based on usage. |
| Hardware | Development Machines | Approx. $1,500 - $2,000 (per machine) |
| Mobile Devices for Testing | Approx. $200 - $800 (per device, depending on the model.) |
| Cloud Servers | Approx. $100 - $500 (per month, depending on usage and scalability needs.) |
| Labor Cost | Labor cost is not applicable, as the project is being executed without external labor expenses. | $0 |

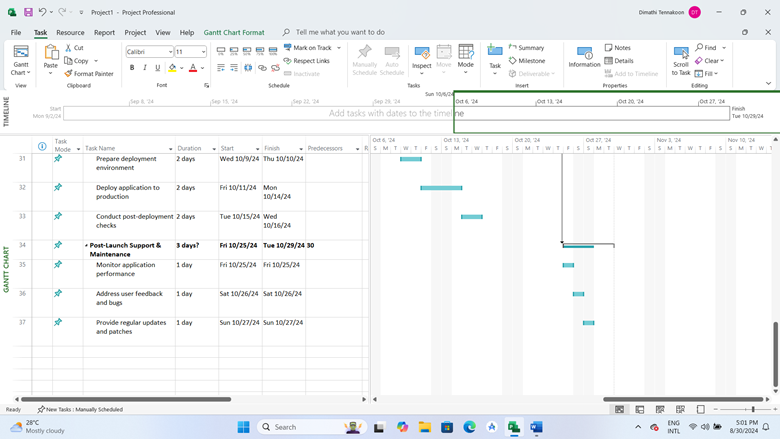
# Gantt Chart / Risks

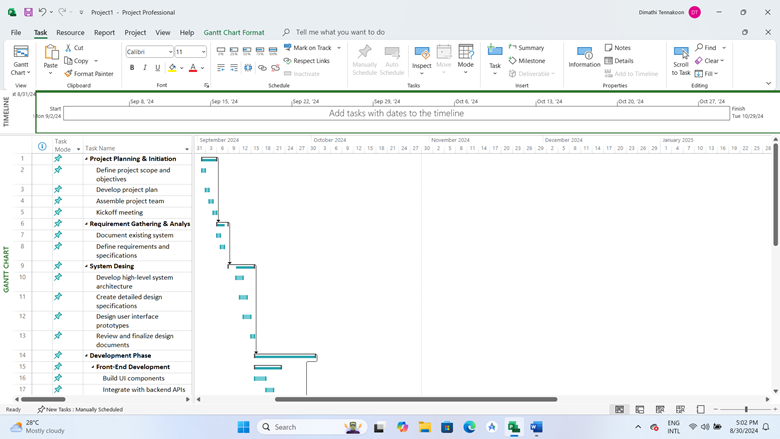
## Project Phases

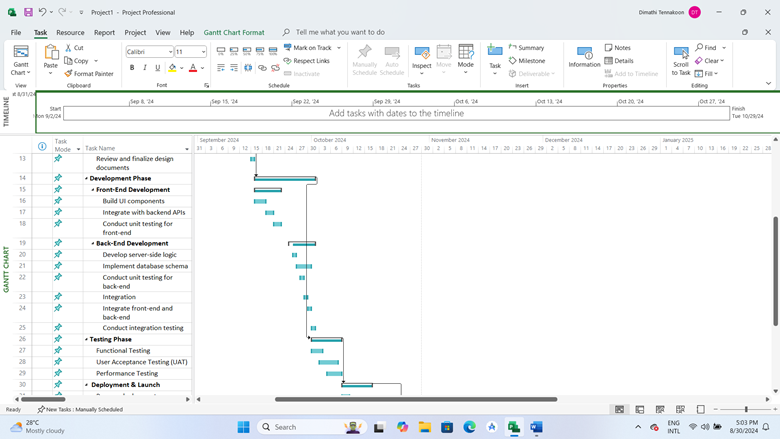


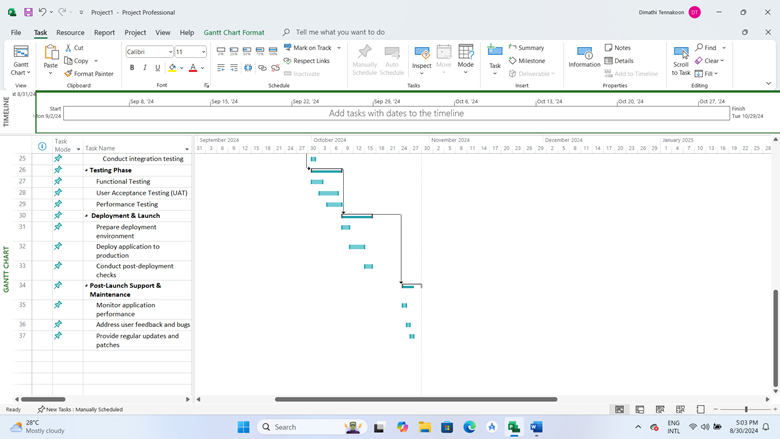












## 9.2. Project Risks

**1. Technical Risks**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Description** | **Mitigation Strategy** |
| **Integration Issues** | Difficulty integrating front-end and back-end systems might cause delays or functionality problems. | Conduct regular integration testing throughout development. Use modular design principles to isolate integration points and ensure compatibility. |
| **System Performance Problems** | The app may experience performance degradation with a high number of users or reports. | Implement performance optimization techniques, conduct stress testing, and ensure scalability in architecture design. |
| **Security Vulnerabilities** | The app may be susceptible to security breaches or data leaks, especially with anonymous reporting. | Follow best practices for security, such as encryption, regular security audits, and secure coding practices. Implement robust authentication and authorization measures. |

**2. Project Management Risks**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Description** | **Mitigation Strategy** |
| **Scope Creep** | Uncontrolled changes or additional features requested beyond the original project scope could delay delivery. | Define a clear project scope and change management process. Regularly review and validate scope with stakeholders. |
| **Resource Availability** | Key team members might become unavailable due to unforeseen circumstances, impacting project timelines. | Cross-train team members and maintain a buffer for critical roles. Have a backup plan for key personnel. |
| **Timeline Delays** | The project may not meet deadlines due to unforeseen technical or managerial challenges. | Develop a detailed project schedule with milestones. Regularly monitor progress and adjust plans as necessary. Use agile methodologies to adapt to changes. |

**3. Financial Risks**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Description** | **Mitigation Strategy** |
| **Budget Overruns** | The project might exceed the allocated budget due to unexpected expenses or mismanagement. | Prepare a detailed budget and regularly review expenditures. Implement cost control measures and keep a contingency fund. |
| **Funding Issues** | Potential issues with securing or maintaining project funding could impact development. | Establish clear funding agreements and maintain regular communication with stakeholders. Have a plan for securing additional funding if needed. |

**4. Operational Risks**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Description** | **Mitigation Strategy** |
| **User Adoption Challenges** | The app might face low user adoption or engagement despite its functionality. | Develop a user engagement plan, including marketing and educational materials. Gather user feedback and iteratively improve the app based on user needs. |
| **Regulatory Compliance** | Failure to comply with data protection regulations and local laws may result in legal issues. | Stay informed about relevant regulations and ensure compliance through regular audits and legal consultations. |

**5. External Risks**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Description** | **Mitigation Strategy** |
| **Technological Changes** | Rapid changes in technology could affect the app’s compatibility or relevance. | Stay updated with industry trends and adopt flexible, adaptable technologies. Plan for regular updates and technology reviews. |
| **Competition** | New or existing competitors might offer similar or better solutions. | Conduct market research to understand competitor offerings and differentiate the app through unique features and superior user experience. |

## Risk Metrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Impact** | **Risk Rating** |
| Integration Issues | Medium | High | High |
| System Performance Problems | High | High | High |
| Security Vulnerabilities | High | High | High |
| Scope Creep | High | Medium | High |
| Resource Availability | Medium | Medium | Medium |
| Timeline Delays | Medium | High | High |
| Budget Overruns | Medium | High | High |
| Funding Issues | Medium | High | High |
| User Adoption Challenges | High | Medium | High |
| Regulatory Compliance | Medium | High | High |
| Technological Changes | High | Medium | High |
| Competition | Medium | Medium | Medium |

## Risk Management Plan

|  |  |
| --- | --- |
| **Risk Identification** | Regularly review and update the risk register. Engage stakeholders to identify new risks throughout the project lifecycle. |
| **Risk Analysis** | Assess the impact and likelihood of each risk. Prioritize risks based on their potential effect on project objectives. |
| **Risk Mitigation** | Develop and implement strategies to minimize or eliminate risks. Assign risk owners to ensure accountability. |
| **Risk Monitoring** | Continuously monitor risks and the effectiveness of mitigation strategies. Adjust plans as needed based on risk status and project changes. |
| **Risk Communication** | Keep all stakeholders informed about risks and mitigation strategies. Ensure transparent communication about risk management efforts. |

# Conclusion

The proposed mobile app for reporting problems in Sri Lanka is designed to empower citizens by providing a convenient platform for voicing concerns about public issues. By enabling users to post images, descriptions, and categorize problems, the app fosters community engagement and transparency. It enhances the public’s ability to participate in civic activities, bringing attention to pressing issues through features like voting, commenting, and sharing.

Through user’s user journey, the app’s functionality is demonstrated as a seamless and user-friendly tool for addressing infrastructure and service-related problems. From secure user registration with OTP verification to interactive map features and admin oversight, every aspect of the app is geared towards creating a robust, reliable, and transparent platform.

Admin functionalities further ensure that the platform remains trustworthy and compliant with community standards, maintaining a balance between user anonymity and accountability. This system promotes a collaborative environment where issues are addressed efficiently, leading to improvements in local governance and public services.

Ultimately, this app represents a significant step forward in leveraging technology to enhance civic participation and problem-solving in Sri Lanka. By providing a direct link between citizens and authorities, it facilitates more responsive and effective resolution of public issues, fostering a stronger, more connected community.

# References

* (No date a) *FixMyStreet*. Available at: https://www.fixmystreet.com/ (Accessed: 24 August 2024).
* (No date b) *Free cloud computing services - AWS free tier*. Available at: https://aws.amazon.com/free/?trk=14a4002d-4936-4343-8211-b5a150ca592b&amp=&sc\_channel=ps&amp=&ef\_id=Cj0KCQjw9fqnBhDSARIsAHlcQYRFx7F\_fi5QinYZgTtclPo16u8sVIgNLUv1e7R8cibQm9fBRsNDvmkaAqVdEALw\_wcB:G:s&amp=&s\_kwcid=AL!4422!3!453325184782!e!!g!!aws!10712784856!111477279771&amp=&all-free-tier.sort-by=item.additionalFields.SortRank&amp=&all-free-tier.sort-order=asc&amp=&awsf.Free+Tier+Types=\*all&amp=&awsf.Free+Tier+Categories=\*all (Accessed: 27 August 2024).
* Atlassian (no date) *What is agile?*, *Atlassian*. Available at: https://www.atlassian.com/agile#:~:text=The%20Agile%20methodology%20is%20a,READ%20ON%20BELOW (Accessed: 25 August 2024).
* *Get connected* (no date) *The City of San Antonio - Official City Website*. Available at: https://www.sanantonio.gov/GetConnected/Mobile-Apps/311-App (Accessed: 24 August 2024).
* *Seeclickfix* (no date) *https://seeclickfix.com*. Available at: https://seeclickfix.com/ (Accessed: 24 August 2024).
* *What is risk management?* (2021) *IBM*. Available at: https://www.ibm.com/topics/risk-management#:~:text=Risk%20management%20is%20the%20process,an%20organization’s%20capital%20and%20earnings. (Accessed: 29 August 2024).

# CSE5015 Computing Project Proposal Approval Sheet

*Student Use only;*

|  |  |  |  |
| --- | --- | --- | --- |
| Group Name | 5 | Attempt No | 1st Attempt |
| Title of the project | FixIT Lanka | | |
| System platform | Web Application Desktop Application  ✔️ Mobile Application ……………………………… | | |
| Technology |  | | |
| Programming methodology | Structured System Analysis and Design Methodology  ✔️ Object Oriented Analysis and Design Methodology | | |
| SDLC | Agile Methodology | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Student No | CL/HDCSE/CMU/112/02 | Full Name | Karagoda Patabendige Rasindu Savan Bingusara |
| Roles & Responsibilities;  Project Manager   * Manage project planning, scheduling, and resource allocation. * Coordinate communication among stakeholders and the team. * Monitor project progress and implement risk management strategies. * Ensure the project stays within budget and meets business objectives. | | | |
| Student No | CL/HDCSE/CMU/112/05 | **Full Name** | Poruthotage Harshana Dananjaya Dimithri Fernando |
| Roles & Responsibilities;  Front-End Developer   * + Design responsive and visually appealing UI for different devices.   + Integrate front-end components with back-end services.   + Conduct UI testing to ensure usability and performance.   + Implement interactive features like animations and transitions. | | | |
| Student No | CL/HDCSE/CMU/112/30 | **Full Name** | Randage Adithya Purna Lankathilake Fernando |
| Roles & Responsibilities;  Back-End Developer   * + Develop and maintain server logic and database interactions using Firebase.   + Ensure secure and efficient data handling between the server and client.   + Integrate third-party services and APIs as needed.   + Optimize the application for scalability and performance. | | | |
| Student No | CL/HDCSE/CMU/112/127 | **Full Name** | Sandali Tharusha Schokman |
| Roles & Responsibilities;  System Analyst   * Gather and translate user requirements into technical specifications. * Conduct feasibility studies and analyze system design. * Liaise between stakeholders and the development team to ensure clarity. * Ensure that the system meets both functional and non-functional requirements. | | | |
| Student No | CL/HDCSE/CMU/112/29 | **Full Name** | Handunneththi Shanidu De Zoysa |
| Roles & Responsibilities;  Quality Assurance   * + Develop and execute comprehensive test plans and cases.   + Perform functional, regression, and performance testing.   + Identify, document, and verify the resolution of bugs and issues.   + Collaborate with developers to ensure quality and functionality. | | | |

**System Functionality:**

|  |  |
| --- | --- |
| **Function** | **Description** |
| Login | Allows users to access the app by entering their credentials. |
| Signup | Enables users to create a new account with their email, phone number, and OTP verification. |
| Create Post | Allows users to create a post by adding images, descriptions, and selecting a category. |
| Add Images | Users can upload images to their posts. |
| Add Description | Users can add a text description to their posts. |
| Select Category of Issue | Users select a relevant category for the issue they are reporting. |
| Add Comment | Any user can comment on posts made by other users. |
| Vote System | Users can upvote or downvote posts. |
| Share Option | Allows users to share posts on other platforms or within the app. |
| Save Option | Users can save posts to revisit later. |
| Share Saved Post | Users can share saved posts on their profile wall. |
| Anonymous Option | Users can choose to post anonymously, hiding their identity from other users. |
| Map - Search by Location | Users can search posts based on location using the map feature. |
| Map - Sort by Date, District, Province, Vote | Users can sort posts by different criteria, including date, district, province, and vote count. |
| Map - Display Post of User’s Area | Uses GPS to show posts related to the user’s location. |
| Map - Most Problem Areas | Highlights areas with the most reported problems. |
| Home - Trending Post | Displays posts that are trending based on user interactions. |
| Home - Recent View Post | Shows the most recently viewed posts. |
| User Profile | Users can set up a profile with their username, first name, last name, email, phone number (with OTP verification), and profile picture. |
| View User Profile Details (Admin) | Admins can view user profiles, including those who post anonymously. |
| Admin Review (Admin) | Admins review and manage posts, including removing inappropriate content. |

***Official use only***;

**Status:** **Approved / Rejected**

**Date of the status:** ………………………………

**Approved by:**

……………………… …………………………….. ……………………………. ……………………….

Approval Status: **Approved / Rejected**

Date of the Status: …………………………..