

Rainy

A crowdsourced platform for
collecting and analyzing
rainfall data

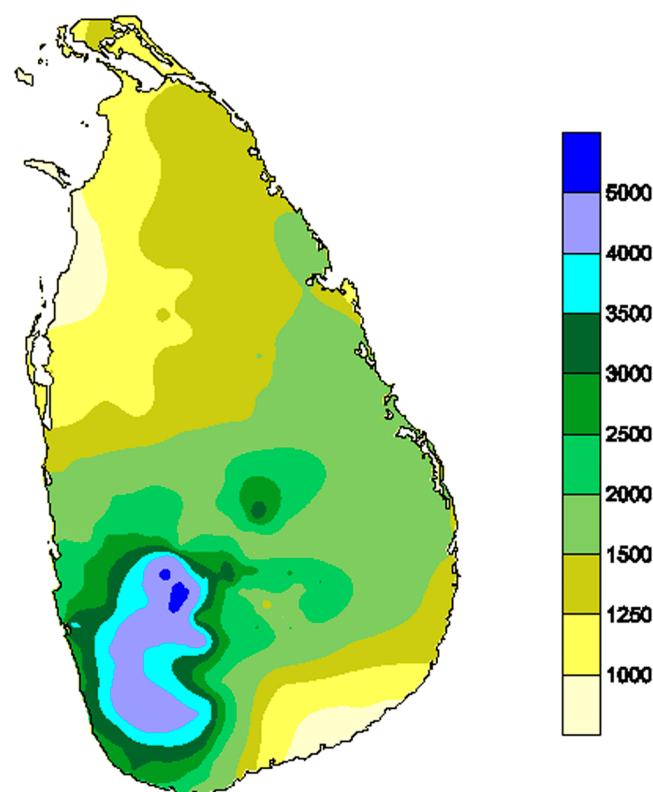


Background

Our day-to-day life can be heavily impacted by the weather. In a tropical country like Sri Lanka weather prediction can be quite difficult. But by collecting real time rainfall data, predictions can be significantly improved.

Official data from the Sri Lanka Department of Meteorology provides insight into seasonal rainfall variations, which can be integrated into our system to improve prediction accuracy. From these we can see that Sri Lanka's climate is influenced by two main monsoon seasons and two inter-monsoon periods. This can result in distinct rainfall patterns.

Knowing the weather is useful for everyone, but agriculture, transportation and infrastructure planning are particularly impacted by it. Sri Lanka's agricultural sector can benefit significantly from more accurate rainfall predictions.



Annual rainfall in Sri Lanka
Source: Department of Meteorology

Project Description

The system will allow users to report rainfalls by recording the start and end times, GPS location, and a rain intensity measurement on a five-value scale as Very heavy rain, Heavy rain, Moderate rain, Light rain, and Very light rain. The collected data will be stored in a repository, enabling ML models to refine weather predictions.

Also, there will be a point system to motivate users which allows them to select gift items.

- ◊ Enhance weather prediction by using user data with seasonal rainfall patterns.
- ◊ Educate users on expected rainfall trends to improve quality assessments.
- ◊ Validate and cross-check user submissions against other user data by sending notifications.
- ◊ Send collected data to a database server that can be used in a ML/AI software product.

Importance and Usefulness

Enhances weather predictions with real-time, local data.

Strengthens agricultural decision making by offering accurate rainfall insights.

Supports flood prevention and water resource management.

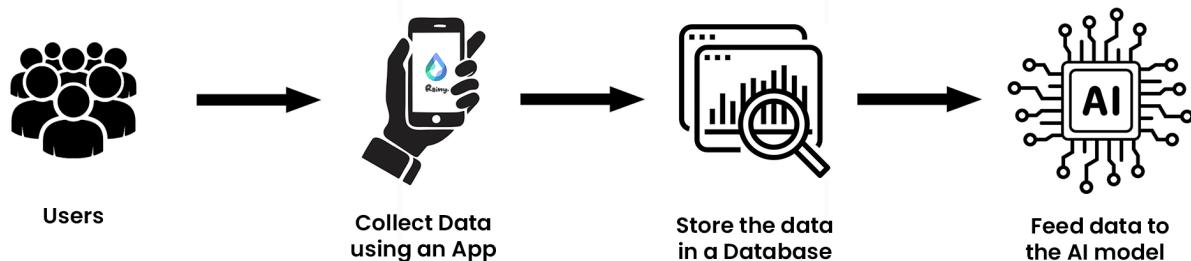
Improves transportation planning by predicting bad weather.

Making infrastructure planning convenient by considering rainfall impacts.

System Requirements Analysis

The rainfall app outlines key components: **user authentication**, **real-time rain data collection through sensors**, **GPS for location-based data**, **data transmission to a server**, **efficient database management**, and **AI/ML model integration** for improved weather predictions. These elements ensure the app delivers accurate and localized rainfall data for better forecasting.

Overview of the System



- 1. User Authentication:** The app shall enable secure user authentication to maintain data integrity and prevent unauthorized access.
- 2. Rain Data Collection:** Users can record rain events by logging start and end times along with GPS location. A scale of five will classify rainfall intensity.
- 3. GPS and Location Services:** The system will utilize GPS to provide precise location tracking for recorded rainfall events.
- 4. Data Transmission:** The system will transmit all collected data to a central database.
- 5. Database Management:** Rainfall data will be securely stored in a cloud-based database, allowing easy retrieval and analysis.
- 6. AI/ML Model Integration:** The system will analyze rainfall data using AI/ML models to generate weather predictions and trends.

System Requirements Specifications

The rainfall app focuses on developing a system to analyze rainfall data in Sri Lanka. It outlines the necessary features for real-time data collection using smartphone sensors, integration with AI/ML models for weather prediction, and a user-friendly interface to display accurate rainfall patterns, helping improve weather forecasting accuracy.

Functional Requirements

Users should be able to log rainfall information via a smartphone app.
GPS location must be captured at event start and end.
Users must classify rainfall intensity on a scale of five.
Collected data must be sent to a central database securely.
System should allow retrieval of past data.
The system should use official meteorological datasets to improve validation.
ML/AI models should process rainfall data to predict future weather.

Non-Functional Requirements

The app should have a user friendly, intuitive interface.
Data must be securely transmitted and encrypted to ensure privacy.
The system should provide real time updates with low latency.
Scalability to handle increasing users and data.
Reliability and availability are essential for ensuring consistent and accurate data collection and reporting.

Software Requirements Analysis

A good UI makes the app easy to use and understand. Each screen has a specific job to help users record and track rain. The design includes buttons, forms, and clear messages to guide users smoothly. From logging rain data to checking profiles, every part of the app is made to be simple and helpful.

UI Breakdown

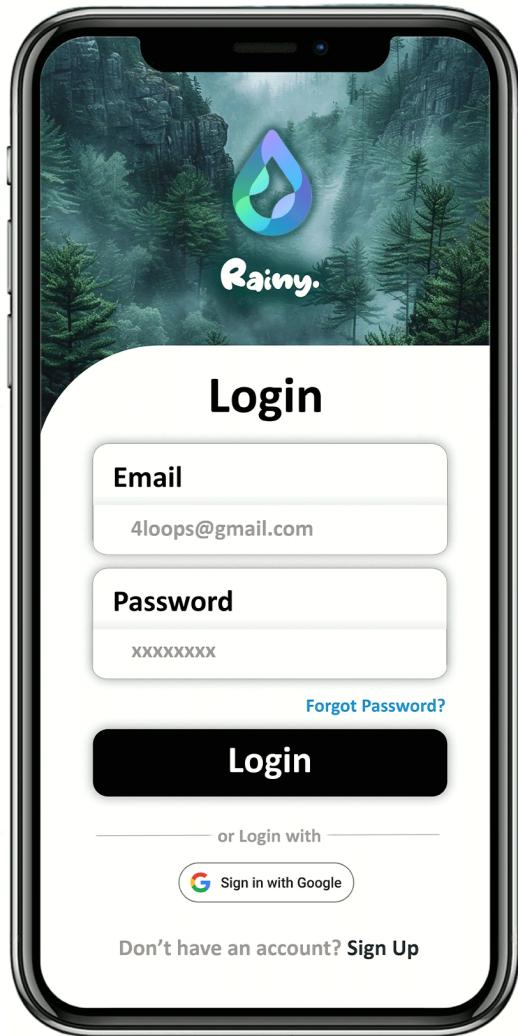


*App Icon in the
App Drawer*

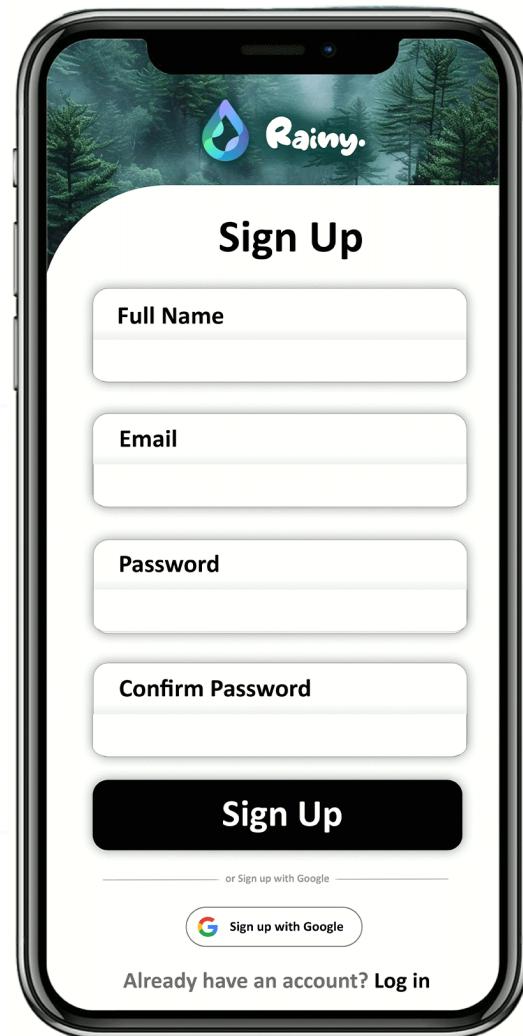


Welcome Screen

Software Requirements Analysis



Login Screen



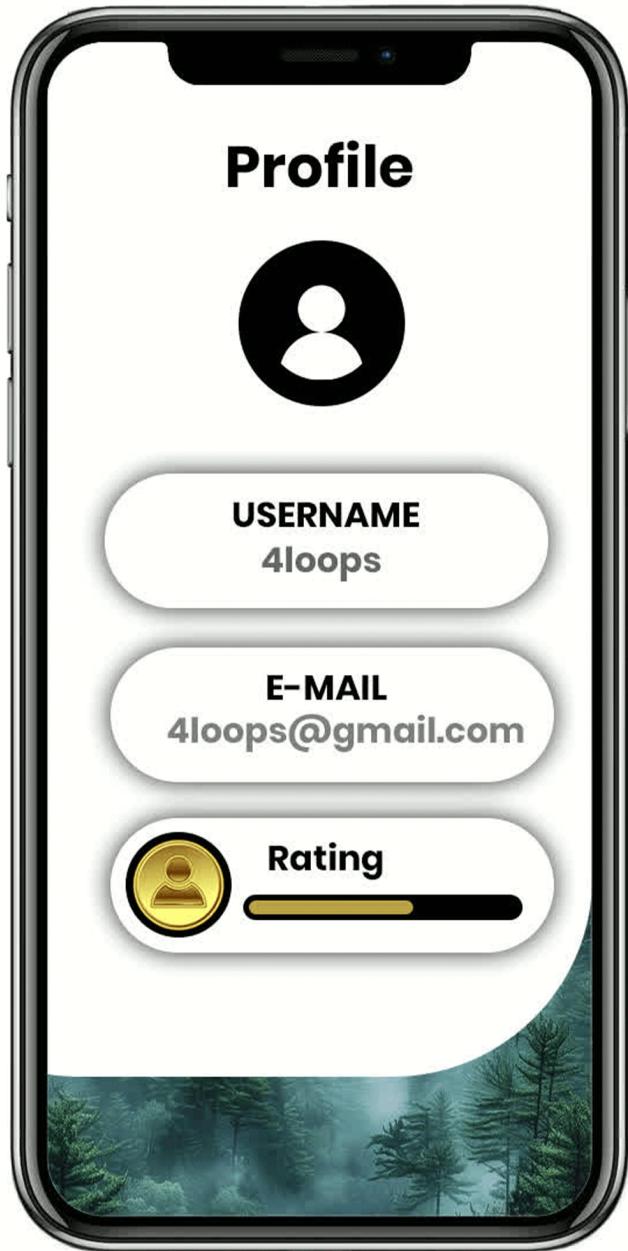
Sign up Screen

There should be a login screen and a sign up screen for registration of new users.

The login page should contain the option to login with email or with Google Account. It should include a Forgot Password? link to reset the password and a link to create a new account.

The Sign up page should contain the option to sign up with email or with Google Account and a link to go back to the Login screen.

Software Requirements Analysis



Profile Screen

The Profile screen will show the username and email of the user.

Additionally, it will show the user's rating based on the point system. This will allow the user to keep track of how helpful their records have been.

Software Requirements Analysis



Rain Recording Screens

The Rain recording screens will allow the users to select the type of rain and confirm the time period of the rain. It will also show the location of the user and allow the user to change it if it is inaccurate.

Software Requirements Analysis



The confirmation page for rain recording will show the summary of the recording and allow the user to check for any additional changes in the recording and change them if needed.

*Rain Recording
Confirmation Screen*

Software Requirements Analysis



Notification Alert

Additionally, our app should support notification alerts asking users if it's raining outside. This is based on data submitted by other users in the same area and time.

This will allow us to confirm data that is submitted by a certain user and will allow us to confirm the legitimacy of the data submitted by a user.

We can make it less annoying for the user by including 2 buttons in the notification which the user can just tap without being taken in to the app.

Software Requirements Specifications

Functional Requirements

Technology Stack

- Flutter – A cross-platform framework for building the app on both Android and iOS.
- Dart – The programming language used for Flutter development.
- VS Code – The primary code editor for writing and debugging the app.
- FlutterFlow.io – A visual UI builder for rapid interface design.

Database

- Supabase – A cloud-based backend solution for storing and managing user data, rain records, and authentication details.

Libraries & Dependencies

- Flutter Geolocator – To get the user's location for weather-related features.
- Google Sign-In – For Google authentication, allowing users to log in with their Google accounts.
- Flutter Secure Storage / Hive – For storing user session data and keeping track of recorded rain events locally.
- http / Dio – For handling API requests to fetch or send data to the backend.

Other Useful Tools

- Firebase Messaging – For push notifications (e.g., alerts about rain updates).
- Provider / Riverpod – For state management, keeping the UI responsive and smooth.
- Shared Preferences – To store simple user settings locally.
These tools and libraries will ensure the app is functional, efficient, and user-friendly.

Software Requirements Specifications

Non-Functional Requirements

1. Security

- Encrypted Data Transfer – Uses HTTPS with TLS for secure communication.
- Secure Authentication – Google Sign-In with OAuth 2.0 for safe login.
- Encrypted Local Storage – User data stored using AES encryption.

2. Privacy & Data Protection

- Minimal Data Collection – Only essential data (location, rain records, profile) is stored.
- User-Controlled Data – Users can delete their data anytime.

3. Performance & Reliability

- Optimized API Calls – Cached data reduces load times.
- Efficient State Management – Uses Provider / Riverpod for smooth UI.
- Offline Mode – Records data without internet and syncs later.

4. Maintainability & Scalability

- Modular Codebase – Clean architecture for easy updates.
- Cloud Backend – Supabase scales automatically.

This ensures Rainy is secure, private, fast, and scalable.

Final Remarks

Rainy is much more than a rainfall logging app; instead, it is a community-based project to advance weather forecasting by providing real data in real-time to researchers, meteorologists, and the public. By using crowdsourced data, we are able to develop a low-cost, scalable system that will serve the needs of everyone from farmers who depend on weather forecasts to urban planners needing to mitigate the risk of flooding.

Even though users contribute voluntarily, they are rewarded through our point system, ensuring engagement and accuracy while maintaining fairness. At the end of the day, everyone wins—users get incentives, and Sri Lanka gets better weather predictions.

Funding this project accelerates modernization in weather tracking, people's empowerment, and the assurance of a smarter, more connected system in the future. Let's hack together Rainy and make weather forecasting better for everyone.

References

Sri Lanka Department of Meteorology

<http://www.meteo.gov.lk>

Used for annual rainfall data and seasonal rainfall patterns in Sri Lanka.

"How to Build a Weather App Using Flutter and Supabase"

<https://techwithsam.com/build-weather-app-flutter-supabase>

Explains using Flutter for the front end and Supabase for the backend.

"Crowdsourcing Weather Data: A Beginner's Guide"

<https://weathertechinsights.com/crowdsourcing-weather-data>

Discusses how crowdsourcing can collect real-time weather data.

"Integrating GPS and Maps in Flutter Apps"

<https://fluttermapp.com/integrate-gps-maps-flutter>

Tutorial on using Google Maps API and GPS in Flutter apps.

"Using AI/ML for Weather Prediction: A Simple Guide"

<https://datascienceforall.com/ai-ml-weather-prediction>

Overview of how AI/ML models can improve weather predictions.

"Why Crowdsourcing is the Future of Environmental Monitoring"

<https://ecotechdaily.com/crowdsourcing-environmental-monitoring>

Discusses the benefits of crowdsourcing for environmental data collection.

The Team

4Loop

Binara Herath (230244G)

Dineth Sankalpa (230522H)

Muditha Herath (230242A)

Malindu Bandara (230078B)

Thank you for reading.