Sprint 1: Initial Setup

Requirements Artifact for WeatherSensor AI - Evans Chigweshe

1. System Overview

The WeatherSensor AI aims to provide users with real-time weather information enhanced by natural-language summaries generated through a large language model.

2. Functional Requirements

User Story: As a user, I want the application to fetch and display current weather data for a specific location to the console.

1.1 - Define System Architecture - Evans Chigweshe

- FR 1.1.1: The system architecture documentation shall provide a synopsis of the project.
- FR 1.1.2: The system architecture documentation shall describe an overview of the system.
- FR 1.1.3: The system architecture documentation shall provide a summary of the three layers.
- FR 1.1.4: The system architecture documentation shall describe the data flow when a user interacts with the site.
- FR 1.1.5: The system architecture documentation shall describe the component architecture with descriptions for each.
- FR 1.1.6: The system architecture documentation shall provide UML Diagrams for Use Case, Components, and Data Flow.

Progress Summary: Created a comprehensive architecture document of WeatherSensorAI, including the architectural layers, data flow, component architecture, and three UML diagrams showing user interaction, components, and data flow.

Sprint 1: Initial Setup

Requirements Artifact for WeatherSensor AI - **Jackson Yanek**

1. **System Overview**

The WeatherSensor AI aims to provide users with real-time weather information enhanced by natural-language summaries generated through a large language model.

2. **Functional Requirements**

User Story: As a user, I want the application to fetch and display current weather data for a specific location to the console.

1.2 - Create Repository and Initial File Structure - **Jackson Yanek**

- FR 1.2.1: The repository must be publicly accessible.

Progress Summary: Public Repository Created

1.4 - Create and complete the README file in github - **Jackson Yanek**

- FR 1.4.1: The README file must have a clean and readable format.

Progress Summary: Basic Readme skeleton pushed to Github Repository.

Sprint 1: Initial Setup

Requirements Artifact for WeatherSensor AI - **Manu Redd**

1. **System Overview**

The WeatherSensor AI aims to provide users with real-time weather information enhanced by natural-language summaries generated through a large language model.

2. **Functional Requirements**

User Story: As a user, I want the application to fetch and display current weather data for a specific location to the console.

1.3 - Set Up Environment Variables for API keys -

- FR 1.1.1: Generate Gemini Pro API key through KU education account
- FR 1.1.2: Create an environment variable to keep API key private
- FR 1.1.3: Write sample code to test API request

Progress Summary: I researched the most optimal gemini models for our use-case and learned how to use an API key for sending requests for text generation. I used sample code from google and the API requests are working.

Sprint 1: Initial Setup

Requirements Artifact for WeatherSensor AI - **Cole Cooper**

1. **System Overview**

The WeatherSensor AI aims to provide users with real-time weather information enhanced by natural-language summaries generated through a large language model.

2. **Functional Requirements**

User Story: As a user, I want the application to fetch and display current weather data for a specific location to the console.

1.5 - Create simple skeleton website (github hosting) - **Cole Cooper**

- FR 1.5.1: Create HTML Skeleton providing base
- FR 1.5.2: Create CSS file providing style
- FR 1.5.3: Enable Github Pages
- FR 1.5.4: Push to Github and ensure basic functionality

Progress Summary:

Sprint 1: Initial Setup

Requirements Artifact for WeatherSensor AI - Riley England

1. **System Overview**

The WeatherSensor AI aims to provide users with real-time weather information enhanced by natural-language summaries generated through a large language model.

2. **Functional Requirements**

User Story: As a user, I want the application to fetch and display current weather data for a specific location to the console.

1.6 - Evaluate and Select Weather Data Source - Riley England

- FR 1.6.1: Research available APIs (OpenWeatherMap, Visual Crossing, WeatherAPI).
- FR 1.6.2: Compare Data fields (temperature, winds, UV)
- FR 1.6.3: Choose a provider supporting all required parameters.

Progress Summary: Evaluated multiple weather APIs and selected Open-Meteo for initial integration due to its coverage, documentation, and free support. No API key needed, validated a simple query.

Sprint 1: Initial Setup

Requirements Artifact for WeatherSensor AI - **Jackson Y and Manu Redd**

1. **System Overview**

The WeatherSensor AI aims to provide users with real-time weather information enhanced by natural-language summaries generated through a large language model.

2. **Functional Requirements**

User Story: As a user, I want the application to fetch and display current weather data for a specific location to the console.

1.7 -

- **PERSON 2 and PERSON 3**

- FR 1.7.1: Researched open access weather databases with API capability
- FR 1.7.2: Decided to go with The National Weather Service (NWS) API
- FR 1.7.3: Set up the code with documentation from the NWS
- FR 1.7.4: Tested the code and were able to retrieve weather data.

Progress Summary: We researched potential databases for our use-case of real-time weather information and decided on using the NWS API. We successfully implemented the code.