Team 7 Project Charter

Anna Benjamin, Kathryn Frankewich, Austin Klasa, Bridgette Kuehn, Matt Molo

Problem Statement:

Weather is currently communicated to the general population through means of weather channels and websites that display forecasts for a broad locale. However, some people may prefer more accurate, personal weather data for their specific location (i.e. people who live in rural areas). Additionally, weather enthusiasts may want to gather their own weather data. Our weather utility and corresponding web application would allow such personal monitoring of the weather. By having the weather utility interact with a web application, those using our system would be to able to also interact with each other.

Project Objective:

Open Weather Station API:

- Build a personal weather station using a Raspberry Pi and weather sensors (i.e. temperature, wind, barometer, and humidity sensors).
- Create a custom REST API that feeds data from the Raspberry Pi sensors to allow for an open API for other platforms.
- Allow other weather enthusiasts to build their own station and download our software to their personal station.
    - Provide step-by-step documentation on how-to build your own personal weather station.

Webapp Implementation of Weather API:

- Make a master server with a database that aggregates data from each Raspberry Pi so past data can be saved and used for a webapp.
- Using the custom API, create a webpage that receives and displays the weather data collected from the Raspberry Pi sensors.
- If time allows, create an iOS or Android application that synchronizes with the web application.

Stakeholders:

- Users: Those wishing to have a personal weather experience, weather enthusiasts
- Developers: Anna Benjamin, Kathryn Frankewich, Austin Klasa, Bridgette Kuehn, Matt Molo
- Team Leader: Kathryn Frankewich

Deliverables:

Open Weather Station API:

- Weather station
    - Raspberry Pi with necessary sensors temperature, wind, barometer, and humidity sensors
- A Java application that uses the sensors to deliver a REST API with temperature, humidity, pressure and any other data/sensors available.

Webapp Implementation of Weather API:

- Back-end polls weather stations[1] for data at a certain frequency, saving past data in a database for use as historical data.
- Front-end has access to all public weather stations registered[2] with it and will display current weather as well as future forecasts.
    - Selects the closest location using JavaScript geolocation.
    - Webapp will display the weather data on the webpage and also use other API's to grab forecasts, if desired from the user.
    - Webapp will be compatible with desktop and mobile users (different CSS styles).

[1] Weather stations, meaning other Raspberry Pi users that have downloaded our software and have their own station.
[2] Users will be able to choose whether it will publicise the data to the main webapp for other users to view.