Milestone 4

Updated Project Features List

- Python Integration - Non-Functional
- Database Management - Non-Functional
- Hardware, Weather collection instruments - Non-Functional
- Hardware, Mobile Drone Platform - Non-Functional
- Website
    - User database interaction (Functional)
    - Up to the minute information (Non-functional)
- Live Streaming Capability - Functional
- Login and Registration - Functional

Hardware - Weather collection instruments (Non-Functional)
Priority : 1

- The weather station will be using a combo of SparkFun CCS811 and BME280 sensors to measure the barometric pressure, humidity, temperature, TVOCs, and equivalent CO2 (or eCO2) levels in the air. The analogue data gathered by the sensor will be converted to digital using the AVR 14, since the Pi cannot interpret analogue data. This digital data will then be recorded and transmitted to the internet using the Pi and the corresponding wifi hotspot that will exist within the enclosure.

Hardware - mobile drone platform (Non-Functional)
Priority : 6

- The weather station will be capable of measuring pollution levels at 50 ft of elevation and should be mobile enough to reliably travel 1 m/s in the x, y and z direction when in use on the drone. The weather station will be attached to a DJI Phantom 3 Drone. The station will be attached using 3D printed parts, and will be located inside a 3D printed enclosure. The weight of the components used to fixture the weather station to the drone will be minimized as much as possible to ensure the altitude and speed we require are attainable. The center of mass of these additional components must be considered as well, since changing the center of the mass of the drone too much will make it very difficult if not impossible to fly, and we will not be able to reliably control its direction.

Website (Functional)
Priority : 4

- The website will be required to display information based on user input. This will likely be implemented through the incorporation of JavaScript and NodeJS. Data will be pulled

from a database that is constantly updated based on the live data from the drone. Due to the dynamic nature of computer IP addresses, this server will be hosted online likely through Heroku. This information stored in this database will provide the main function of the software, as providing up to date weather information is the softwares main intent. For this to function, user input must be included in calls to our database to provide the desired information from our sensors. While the intention of the software is to provide up to the minute weather sensing, it is a non-functional requirement for the website to communicate with the database on a minute to minute basis. While this is a goal of the software, a longer time period for weather updates would still accomplish the software's goal of providing weather information from a drone mounted sensor.

Python Integration (Functional)
Priority : 3

- The functional reading of the weather sensor data will be handled via Python. Basic operation and usage of the Raspberry Pi's weather data will be handled here. Processing and calculating data will all be done within a Python script and from there it will be available for the database implementation to store. This will be accomplished entirely as backend functionality and hidden from the user.

Database Management (Functional)
Priority : 2

- The functional database of weather data will be managed by Python code analyzing data from the Raspberry Pi and sending it to a cloud-based server on Heroku. This way, as the drone records data and the Raspberry Pi processes it, it will be updated to our database in a quick time-frame and we will be able to retrieve it via NodeJS queries in real time. The database will also include the Python calculations post-flight, and data retrieval via user inputted requests will be handled through NodeJS, that way the user doesn't have direct access to the database or the calculations.

Live Streaming Capabilities (Functional)
Priority : 7

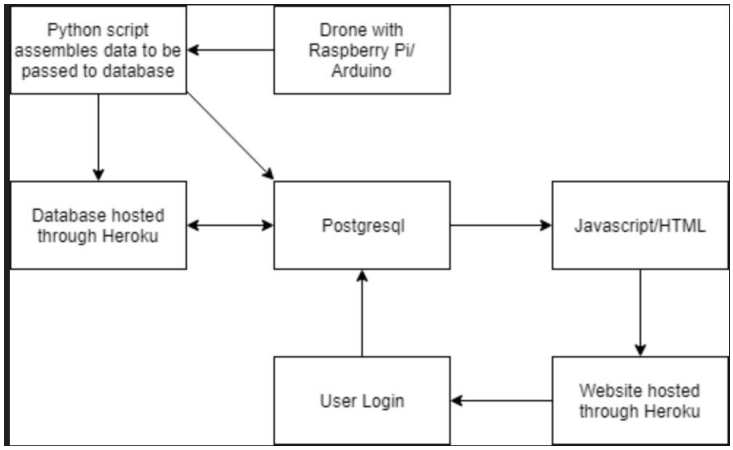
- The functional live streaming of the recorded data will be managed through the use of a wifi link to the Pi and the use of a cell phone wifi hotspot. With this, the pi will process the data from the sensors and through wifi be able to send it to the database via NodeJS. The goal is to have this data submission happen in real time as the drone flies its set course. On the website and server end, the database should also update in close to real time and be able to send the latest processed data to user requests while the drone is in flight.

Login and Registration (Functional)
Priority : 5

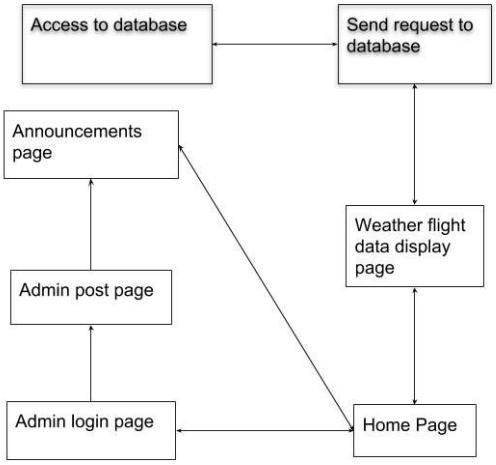
- This login will store user information in a database and create access. This page will be used only for the pages administration. It will provide a way to display announcements

and important information on the website if a person gains access as administration. This display will ask if the user is administration. This will communicate with the Heroku server in order to update and check login information. If login and password match, the user will be given administrator access and be able to post announcements on the website. This will provide security and increased accessibility to the website.

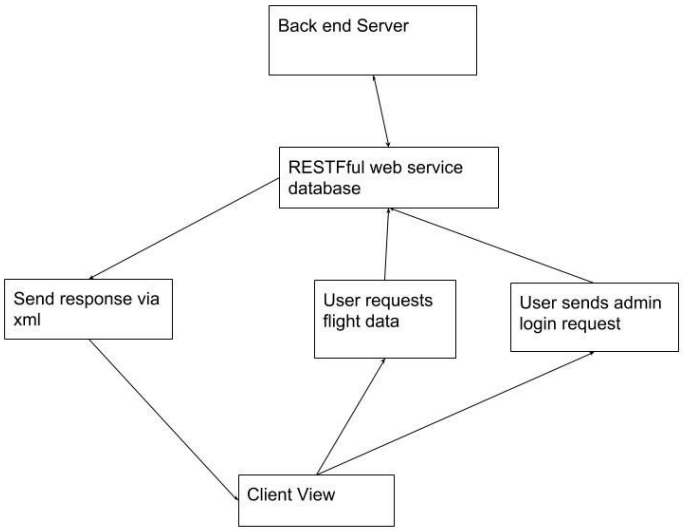
Architecture Diagram



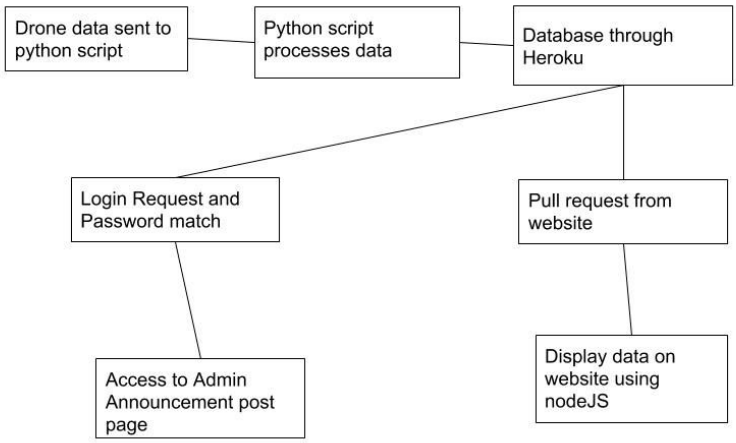
Front End Design



Web Service Design



Back End Design



Project Plan

https://trello.com/b/HrmvlfK3/weather-station