

Requirements

Concept

Our concept is to utilize Raspberry Pis and external weather APIs to capture accurate weather information for remote locations.

Goal

To create a swarm of Raspberry Pis that push weather data to a central API. The API collects and organizes into actionable information that is displayed to an end-user.

Tools and Technologies

- Raspberry pi 3 model B+
- Raspberry pi sense hat
- Various sensors
- Ansible (automation)
- DietPi (OS Distribution)

Project Plan

- **Sprint 1 (ends 9/13)**
 - Project planning (All)
 - Requirements capture (All)
 - Tools and Tech (All)
 - Establish Roles (All)
 - Communications Method (All)
 - Initial Roadmap (All)
 - Configure Frameworks (All)
 - Acquire Hardware (Alex + Jacob)
 - Design Mockups (Dan + Troy)
 - Presentation (All)
- **Sprint 2 (ends 9/25 or 9/27)**
 - Meeting with TARDEC
 - Define Requirements with TARDEC
 - Initial API Configuration
 - Distribution selection for Raspberry Pi
 - Ansible configuration
 - Weather sensor configuration
 -
- **Sprint 3 (ends 10/11 or 10/13)**
 - Weatherproofing/containment for Raspberry Pi
 - Placement of each weather station
- **Sprint 4 (ends 10/23 or 10/25)**
 -

- **Sprint 5 (ends 11/6 or 11/8)**
 -
- **Sprint 6 (ends 11/20 or 11/22)**
 -

Languages

- Python
- JavaScript
- Java

Frameworks

- NodeJS
- Express
- ReactJS

Config. Management

Server Setup

Development Environment

- GitHub
- Local workstations
- Cloud Dev Server
- Visual Studio (Code) / Vim / Various Text editors
- MongoDB Compass
- Discord

Raspberry Pi team: Jacob and Alex

SSH or Remote into each Raspi

- How to handle changing IP address
- Information to collect from the Raspi:
 - a. Thermometer (temperature)
 - b. Barometer (atmospheric pressure)
 - c. Hygrometer (humidity)
 - d. Anemometer (wind speed)
- Control configs of multiple Pi
 - a. Use ansible to control configs
- Get approval to place Raspberry Pi somewhere on campus
- Explore possibility of solar panel for self-contained power
- Security for physical devices
- Weatherproofing of device/sensors

- Network security
- Configure Pi Image
 - a. Determine which distribution we will use
- Collection times for the weather data?

Server & Database Team: Dan and Matt

Format => <%Sprint Completed By%>: <%Requirement%>

- Sprint 3: Import Data transmitted by the Pi DataBase
- Sprint 2: Import API and Scrapped HTML data to DataBase
- Sprint 4: Aggregate data and compare Pi reading to collected data
- Sprint 1: Use MongoDB to create a SQL less Database architecture
- Sprint 1: Use a group created API to control data flow into and out of the server
- Sprint 5: Use Docker and MongoDB to ensure information is backed up quickly and easily
- Sprint X: Provide weather alerts to users based on given parameters

Front End & UI Team: Dan and Troy

- Develop base for website integration with PI and Database
- Graph showcasing the changes in gathered data over time
- The system will collect data from the raspberry pis and display it to others using a online dashboard
- The dashboard will show a history of the recorded weather as a graph, and current weather readings
- The system will at her information from other weather sources and display it alongside SkyPi's data
- User accounts must be able to be created or deleted from the front and back end of the system
- Alerts can be sent out to the front end webpage from the back end based on specified parameters
- The frontend dashboard should be viewable on either desktop or mobile browsers

