Weather Station Software Requirements Specifications

February 8

REPRESENTATIVES
Brandon Jackson
Brian Atiyeh
Jeswanth Kodali
Trevor Malarkey

Revision History

Date	Version	Author	Comments
01/26/2018	v1.0	Brandon Jackson	Document skeleton
02/09/2018	v1.1	Brandon Jackson, Brian Atiyeh, Trevor Malarkey, Jeswanth Kodali	Rough draft to be sent to TA for feedback
2/10/2018	v1.2	Brandon Jackson, Brian Atiyeh, Trevor Malarkey, Jeswant Kodali	Revised version based on TA feedback
2/11/2018	v1.3	Brandon Jackson, Brian Atiyeh, Trevor Malarkey, Jeswanth Kodali	Final version

Document Approval

Printed Name	Title	Date
Brandon Jackson	Team Lead/Presentation Lead	
Signature: X_____		
Brian Atiyeh	Architecture Lead	
Signature: X_____		
Trevor Malarkey	Hardware Lead	
Signature: X_____		
Jeswanth Kodali	Documentation Lead	
Signature: X_____		
Aaron Willcock	Teachers Assistant	
Signature: X_____		
Ryan Wood	Client	
Signature: X_____		
James Mason	Client	
Signature: X_____		
Gregory Czerniak	Client	
Signature: X_____		
Matthew Stein	Client	
Signature: X_____		

Table Of Contents

1. Introduction

1.1 Purpose

The purpose of the software requirements specification document is to provide a detailed overview of the requirements to complete Weather Station project. This document contains the explanations of what features are going to be implemented, layout the interfaces to be used, and what constraints the application has. The detailed requirements provided by this document are agreed upon by the client and the developers.

1.2 Scope

The Weather Station project is a portable weather station to be used by The Department of Defense to acquire weather data in areas where this data is not regularly available or needs to be more accurate than what is available. This will be completed with temperature, atmospheric pressure, and humidity sensors connected to a Raspberry Pi. The information then gathered by the sensors will be displayed to users on a website along with available weather data from Open Maps Weather API. Along with displaying the data to users that data will be stored in to a database for historical information.

The Raspberry Pi weather station as described above will contain a thermometer, barometer, hygrometer that can be deployed in nearby areas to get the weather data. Other sensors include a gyrosphere and accelerometer to let the user know the position and speed the station is currently in. It will be connected to a local area internet connection thru onboard wifi to send information to the website as well as receive any needed information. It will be possible to have more than one Raspberry Pi deployed at a time sending data to get a better analysis of the current weather.

The Weather Station site will allow for authenticated users to be created and logged in to view the current weather. They can view the relative weather of their location that is being taken for the Open Weather Maps API based on the location they are at. To get the specific weather of their location they can view the data coming in from the connected Raspberry Pi's. This data coming in will constantly be live updated with the information so any changes are immediately available to see. The website will also allow for the

connection of multiple Raspberry Pi's at one time. Lastly there will be a page that displays the historical data that has been be retrieved from the Raspberry Pi's. This data will be taken from the MySQL database that has stored the historical data.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
Barometer	Sensor used to measure atmospheric pressure. Units are measured in pounds per square inch (p.s.i).
Thermometer	Sensor used to measure temperature. Units are measured in fahrenheit or celsius.
Hygrometer	Sensor used to measure humidity. This value is measured in a percentage that denotes the amount of water the air can still absorb.
Accelerometer	Sensor used to measure acceleration. This value is measured in meters per second squared.
Gyroscope	Sensor used to determine the orientation of the device. This value is expressed as a degree.
Department of Defense	Branch of the United States government whose primary duty is supervising national security and the United States Armed Forces.
TARDEC	Tank Automotive Research, Development, and Engineering Center.
React	React is a frontend Javascript framework that allows small parts of each web page to be built using components. These components make it easier to split up a web page and keep each individual piece bug free.
Node	Node is a backend Javascript framework built to use a non blocking I/O system. This means that while the Node server is either retrieving or sending data, it will be able to continue working instead of being blocked as that interaction is happening.
Raspberry Pi	An affordable computer chip that is made up of one serial board.

API	Application Programming Interface, we use this to grab information from another website and set up URLs the Raspberry Pi can use to send data.
Github	Web-based hosting service for version control using git, it helps manage multiple user's code on to one repository.
Model	Each model corresponds to a table in a database.
Server	Program that waits for a message to either push or pull data. The server determines if the request should be allowed access to its data, and responds if the request is allowed.
Client	Program that sends a message to a server. There can be multiple clients sending messages to the server at the same time.
HTTP	HyperText Transfer Protocol. Generally used to send data across the web.
Mbps	Megabits per second.

1.4 References

[1] "Raspberry Pi 3 Model B - Raspberry Pi", Raspberry Pi, 2018. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 9- Feb- 2018].

[2] A. Industries, "Raspberry Pi Sense HAT", *Adafruit.com*, 2018. [Online]. Available: https://www.adafruit.com/product/2738?gclid=EAIaIQobChMIxLHk1MmV2QIVRx6GCh376A7yEAQYASABEgI7lvD_BwE. [Accessed: 9- Feb- 2018].

[3] Google Developers. (2018). *Pricing and Plans | Google Maps APIs Pricing and Plans | Google Developers*. [online] Available at: https://developers.google.com/maps/pricing-and-plans/#details [Accessed 11 Feb. 2018].

[4] Openweathermap.org. (2018). *Price - OpenWeatherMap*. [online] Available at: https://openweathermap.org/price [Accessed 11 Feb. 2018].

[5] Google Developers. (2018). *Usage Limits | Gmail API | Google Developers*. [online] Available at: https://developers.google.com/gmail/api/v1/reference/quota [Accessed 11 Feb. 2018].

[6] Bookshelfjs.org. (2018). *Bookshelf.js*. [online] Available at: http://bookshelfjs.org/index.html# [Accessed 12 Feb. 2018].

[7] Knexjs.org. (2018). *Knex.js - A SQL Query Builder for Javascript*. [online] Available at: http://knexjs.org/ [Accessed 12 Feb. 2018].

1.5 Overview

This document states the software requirements of the Weather Station project. The next section of the document will go over a general description of the product perspective, function, user characteristics, constraints, and assumptions. The third section of the document give a description of the specific requirements which will include interfaces (user, hardware, software, and communications), functional, non-functional, design constraints, logical database, and all other requirements. The final section of the document will contain the analysis models including the data flow diagram.

2. General Description

2.1 Product Perspective

The Raspberry Pi weather station project is intended to create a lightweight, inexpensive, and portable weather station that can readily transmit data on the current weather conditions. The data that is picked up by the Raspberry Pi weather station will be sent to a server where the data is processed and then sent to a database to be stored. The scope of this project is to encompass the ability to support a server/client architecture where multiple Raspberry Pi's are able to send data to the server at the same time. By supporting such a framework, the user will be able to collect information on the weather over a large area of space in a relatively cheap and very fast manner.

2.2 Product Functions

The main functionality of the weather station site is to create a simplistic and intuitive experience for the user that will allow them to be able to view the data that has been collected by the weather stations.

To create an account on the weather station site users will need to input their Username, Password, and Email Address. After submitting this information to the website an admin will need to approve the account in order for them to actually be able to login with their newly created account.

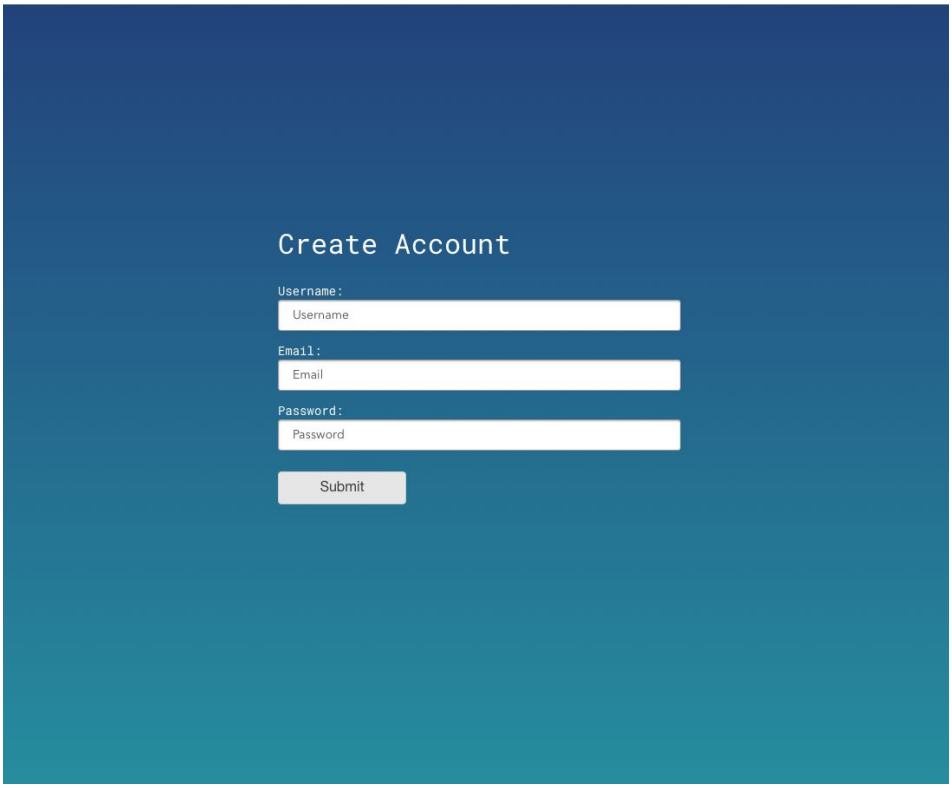


Figure 2.2.1:Create Account View

From the profile page the user has the ability to edit any of the current information that they have associated with their account. If they wish to edit their username, password, email address, or add/change their phone number they are able to from this page. All they need to do is enter the new information in the box next to the associated field and click apply.

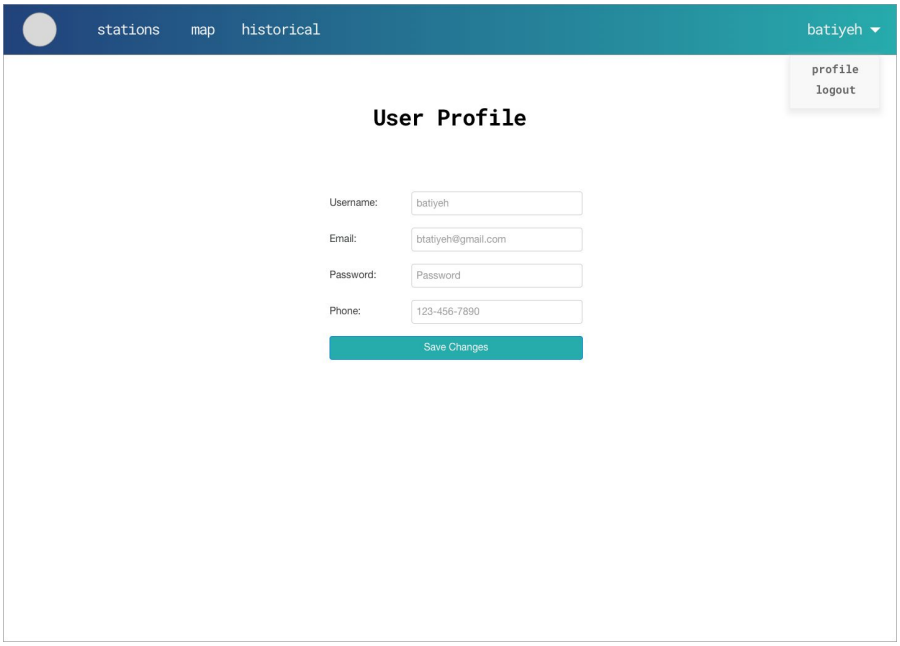


Figure 2.2.2: User Profile View

If the user also has admin privileges, they will see an additional table at the bottom of their user profile page. From this table, admins will be able to click on the privilege type of a user and change it to admin. Once this action occurs, that user will gain admin privileges.

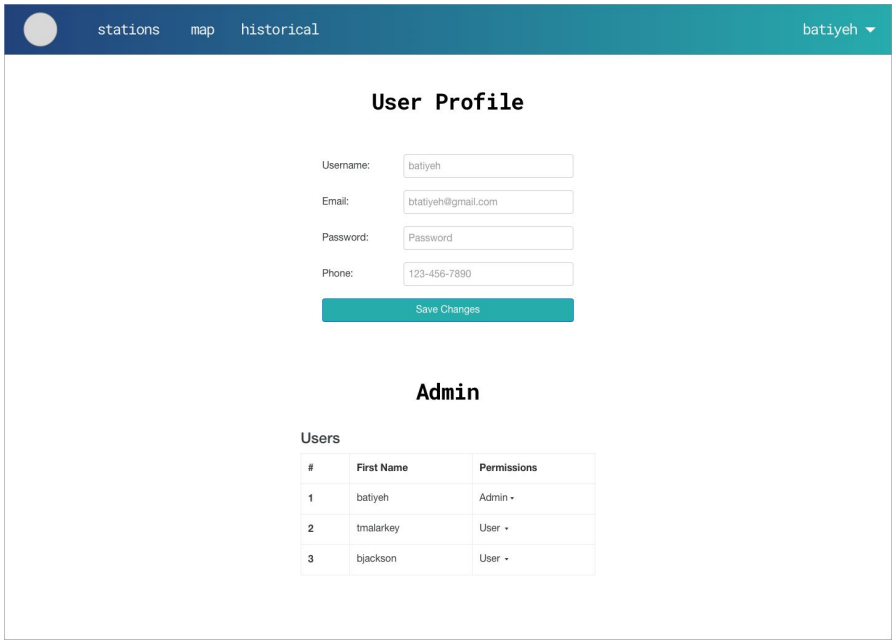


Figure 2.2.3: Admin Profile View

The weather station site has the ability to display a list of all the currently connected weather stations to the user. On this list, the user can see information such as temperature, humidity, atmospheric pressure, wind speed, wind direction, and visibility. If the user wishes to filter this list of stations to find a specific one they can enter the name of the station they want to see in the filter bar.

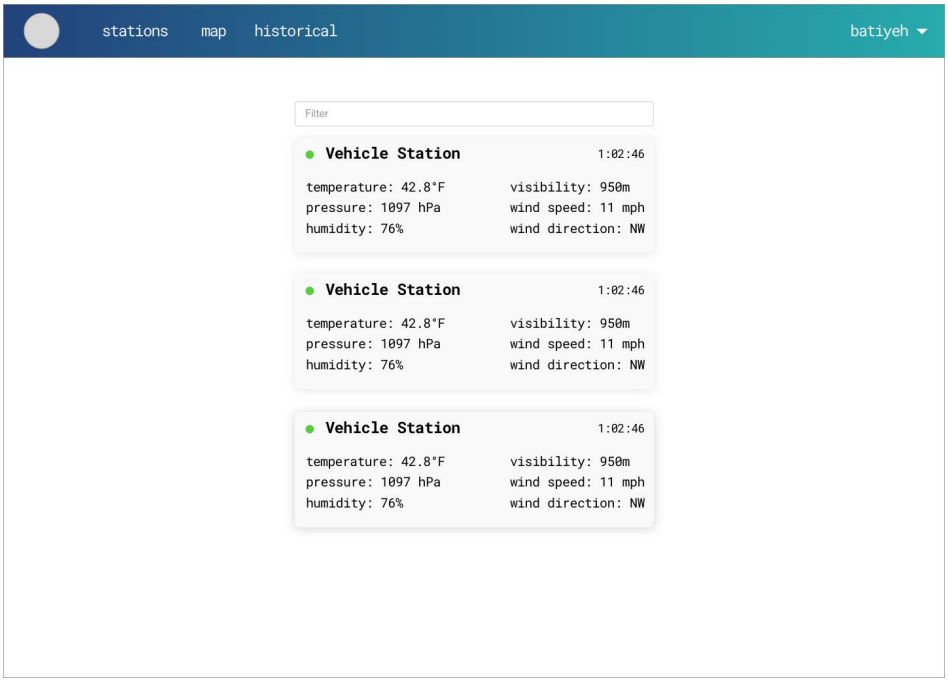


Figure 2.2.4: Connected Stations View

If a user wishes to view a more detailed list of information regarding the station itself they can click on the station and a new menu will come up. On this menu the user will be able to see the MAC address, the connection quality, and the GPS location of the station.

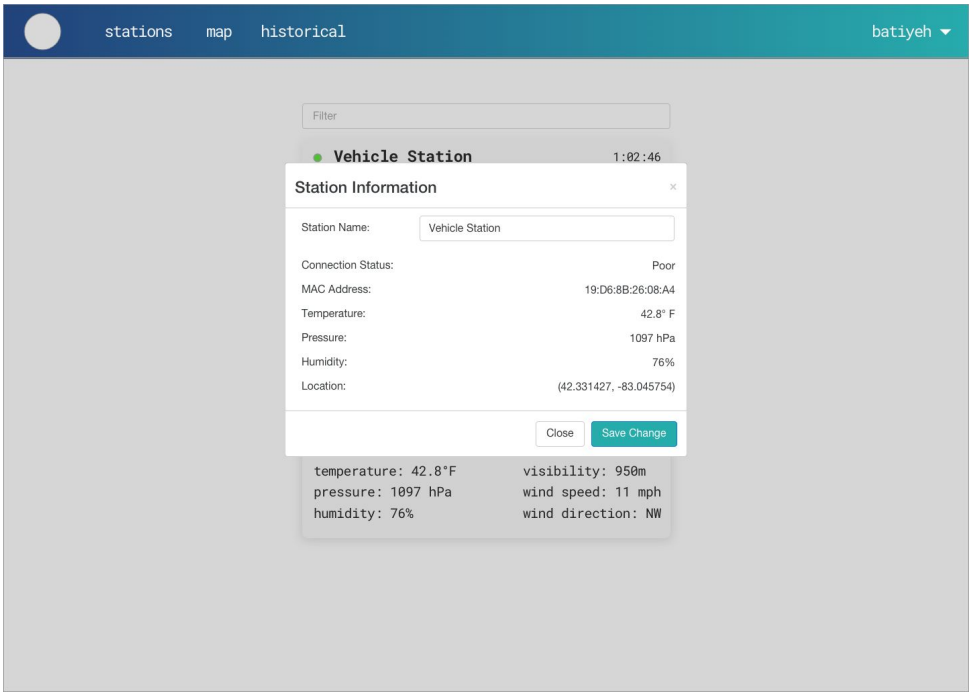


Figure 2.2.5: Connected Stations Additional Information Modal

The weather station site has the ability to map out the location of all the currently connected weather stations and display it to the user. From the maps page the user will see all the locations of the currently connected stations mapped out for them. From their the user can filter this map by clicking on the stations name on the left or by entering in the stations name on the filter bar.

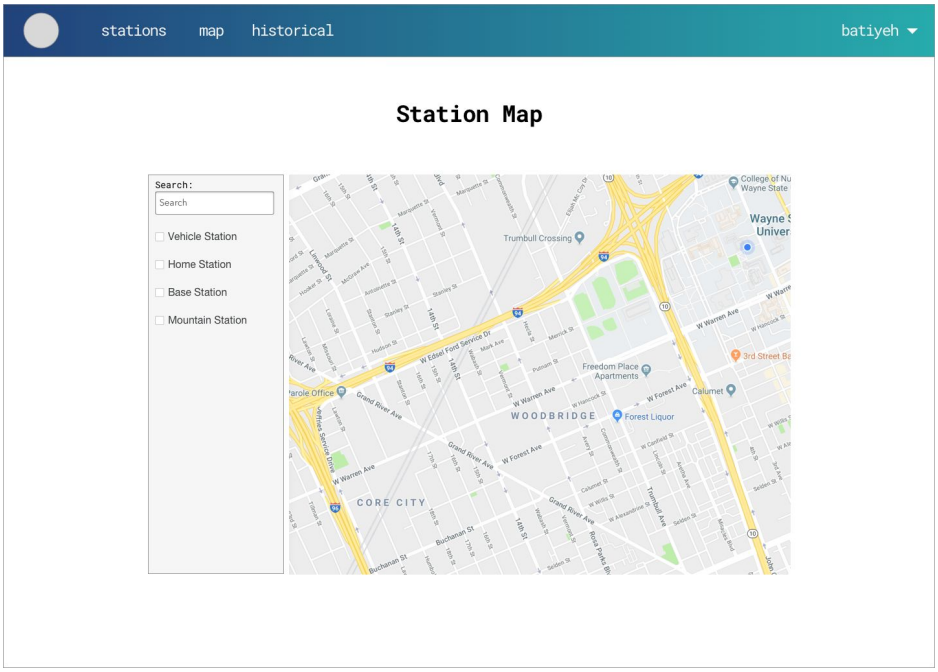


Figure 2.2.6: Station Map View

The weather station site has the ability to display historical information that has been collected by all previously and currently connected stations. From this historical page the user will be greet with a graph that is plotting the past collected at those times. The user can click on plotted points to view more information about what the weather was like at that time. The user can change the time range of the graph by entering in a new time range.

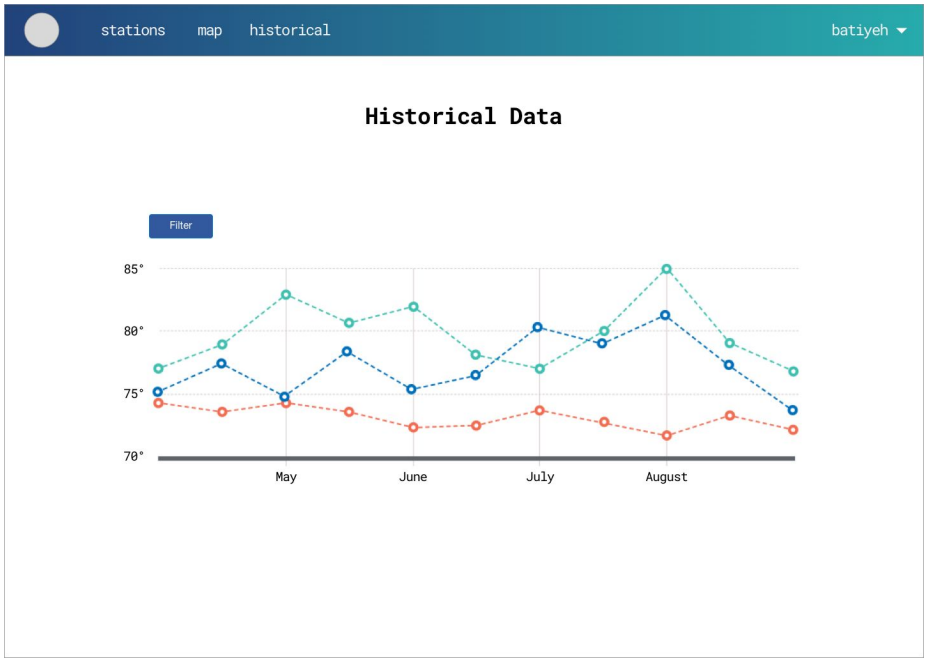


Figure 2.2.7: Historical Data View

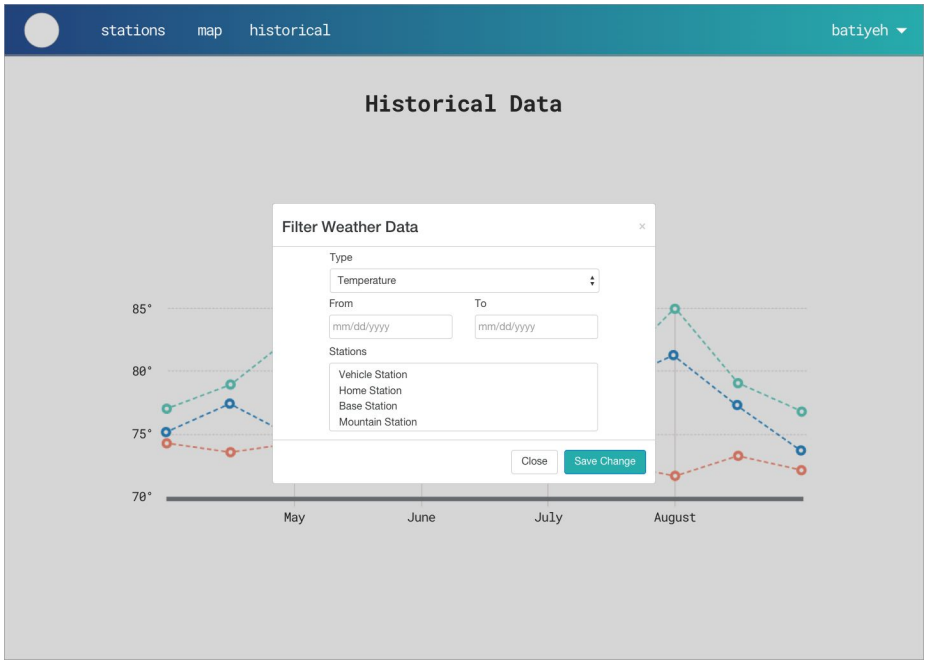


Figure 2.2.8: Historical Data Filter Modal

2.3 User Characteristics

The weather stations will be run by the Department of Defence and as such we can expect that the typical user of the weather station site will be members of the military. Soldiers, operators, or just members of an operating base are some of the people that may be using the weather station site. An admin of the weather station site will be some sort of higher up official in the military.

The weather station website features two different kind of users, a normal user and an admin user. Normal users have the ability to view several pieces of information about the weather and about the weather stations. They can see the current weather at their location, the current weather at the location of all the stations, the location of all the connected stations, and the weather data that was collected by weather stations in the past.

Admin users function the similarly to normal users except they have the ability to grant other users admin privileges. They also have the ability to approve or decline accounts from accessing the website when they are created. Once a user has been approved, they will be able to login normally as one would expect.

2.4 General Constraints

The use of the Raspberry Pi is very substantial to this project, as it records the weather using the sensors attached to them. One potential constraint to the use of a Raspberry Pi is environment. The Raspberry Pi has little to no protection against water and dirt while unprotected and thus must be kept in a clean and dry area.

Another constraint of the Raspberry Pi is the portability. Each station will only be able to run on battery for approximately 20 hours of run time with a 20,000mAh battery pack. Once the battery life is over, the battery would need to be recharged and the Raspberry Pi would have to be plugged into a wall outlet in the meantime. With a battery pack attached to the Raspberry Pi, there is also a constraint of the battery overheating in the sun which could cause the battery to burst. To keep the battery from getting too hot, the pack must be kept away from direct sunlight.

Another constraint for this project is internet connection. For this device to work, the Raspberry Pi needs a consistent internet connection that can push weather data to the server. This connection has to be strong in any condition or place that the device is placed in.

2.5 Assumptions and Dependencies

It is assumed that a user will have access to a computer that has an internet connection. This internet connection needs to be able to support a least a minimum of 5 megabits per second. This type of internet connection will give the user an optimal experience navigating the site.

The user will also have a basic knowledge of using a computer and has a modern version of the current browser that they are using to access the website. For a list of supported browsers and versions please see list 2.5.1 below. Browsers that are not on this list are not guaranteed to work as expected, but may have some functionality. Earlier versions of the browsers on list 2.5.1 are not guaranteed to work as expected either.

Browser	Version
Google Chrome	64.0+
Safari	11.0+
Microsoft Edge	16.0+
Firefox	58.0+

List 2.5.1: Supported Browsers

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

When navigating to the web application, a user will have the opportunity to either login or navigate to register an account. The view on this page will show two fields for username and password, and two buttons for either logging in or going to the register page.

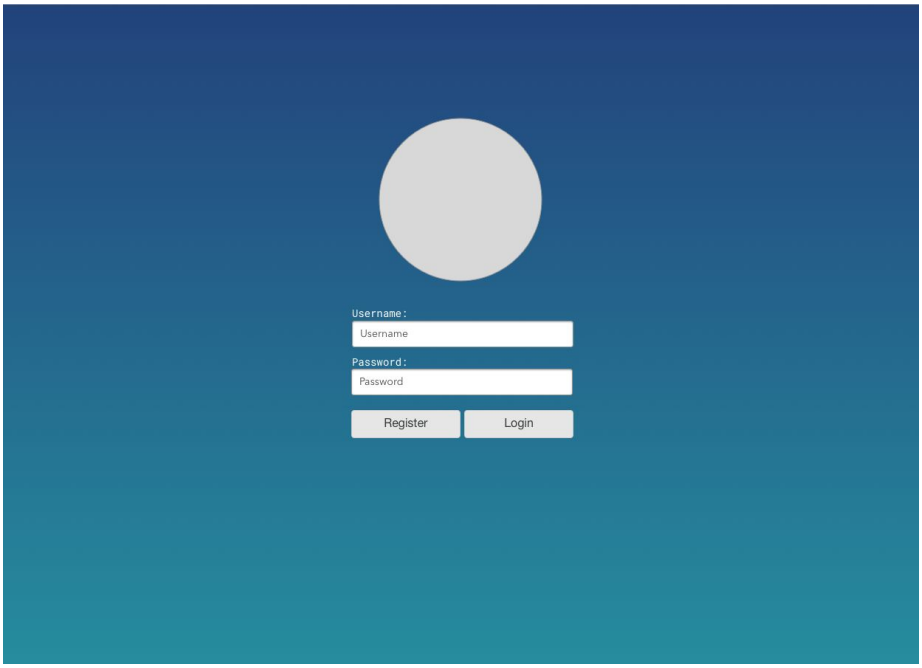


Figure 3.1.1-1: Login View

If a user wants to create an account, they can click the register button on the login page which will redirect them to the create account page. This page will show a form for the user to enter their username, email, and password. After creating their account, they will be redirected to the login page.

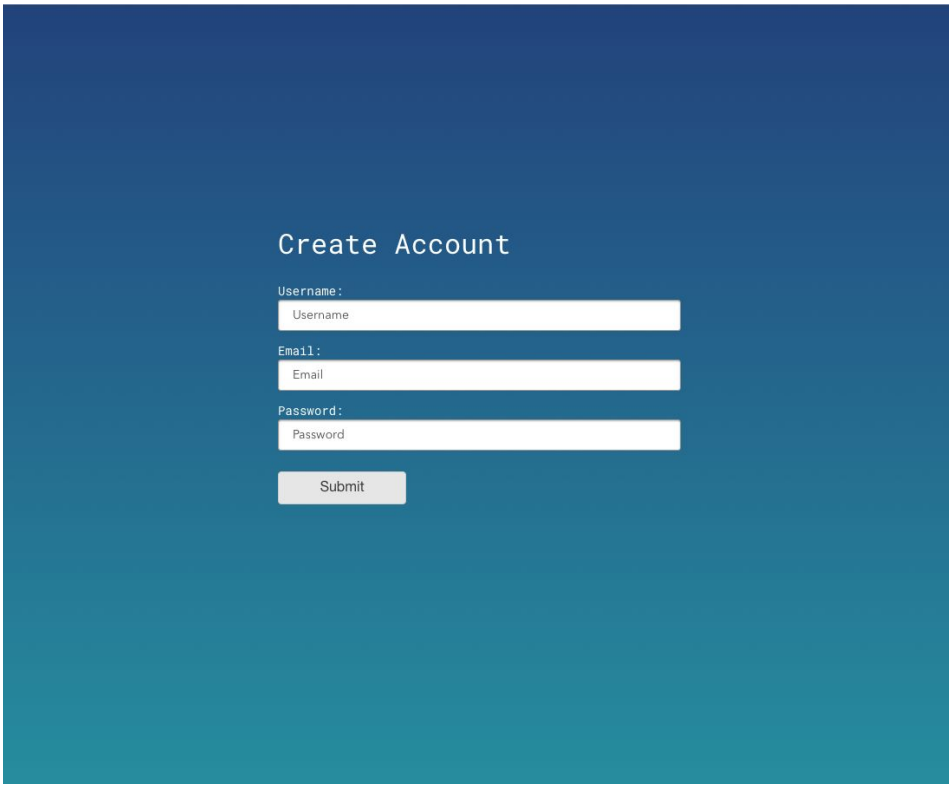


Figure 3.1.1-2: Create Account View

If a user has forgotten their password, they can click a reset password link on the login page to direct them to enter their email. If they do have an account, they will be sent an email with a link to the reset password page where they can enter their new credentials.

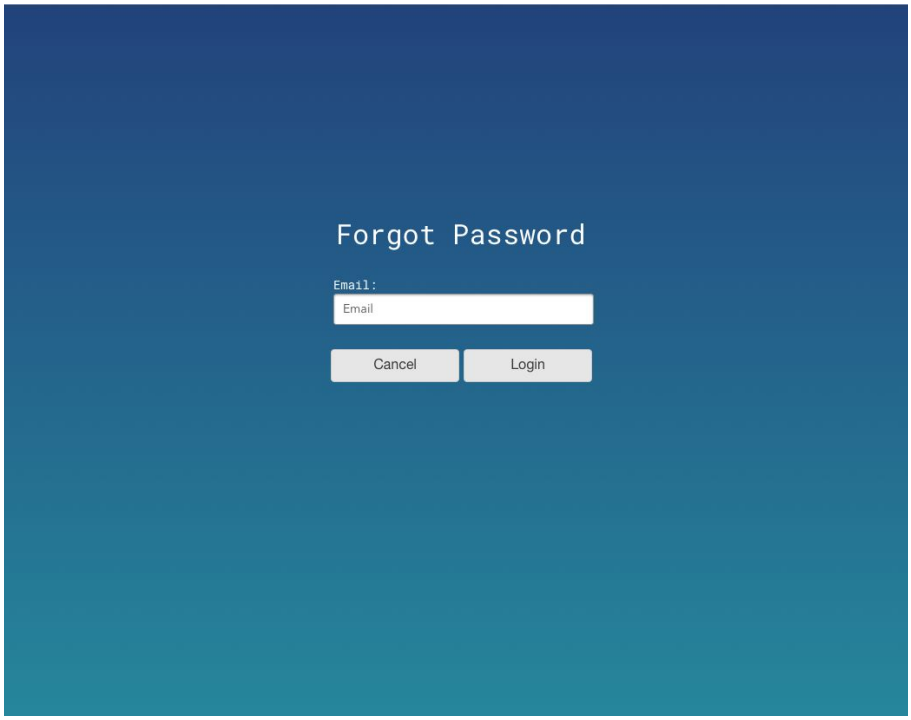


Figure 3.1.1-3: Forgot Password View

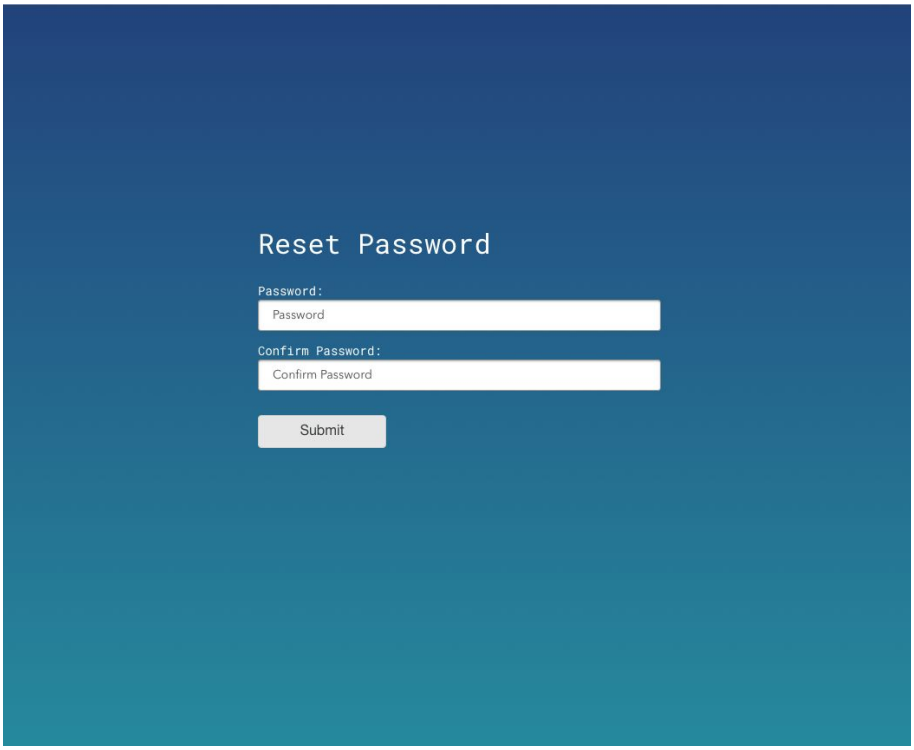


Figure 3.1.1-4: Reset Password View

Once a user has logged into the website, they will be greeted with the connected stations view. This page will display all currently connected weather stations as a scrollable list. This list will also be filterable by station name with a search box at the top of the list. This view will also change styling based on the size of the screen the user is on.

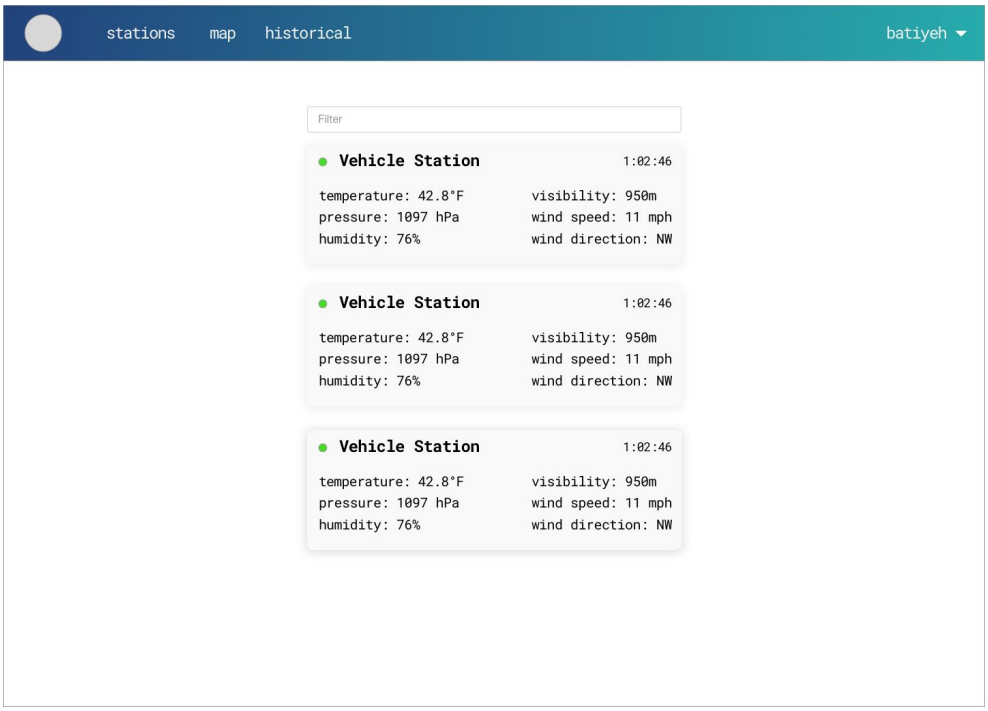


Figure 3.1.1-5: Connected Stations View

While on the connected stations page, the user can click on each weather station card. This will open up a window showing additional information about that station such as MAC address, latitude, longitude, and an input box for the station name. Clicking on the latitude and longitude coordinates will redirect the user to the map page. If the user has admin privileges, they will be able to edit the station's name within the input box at the top of the window.

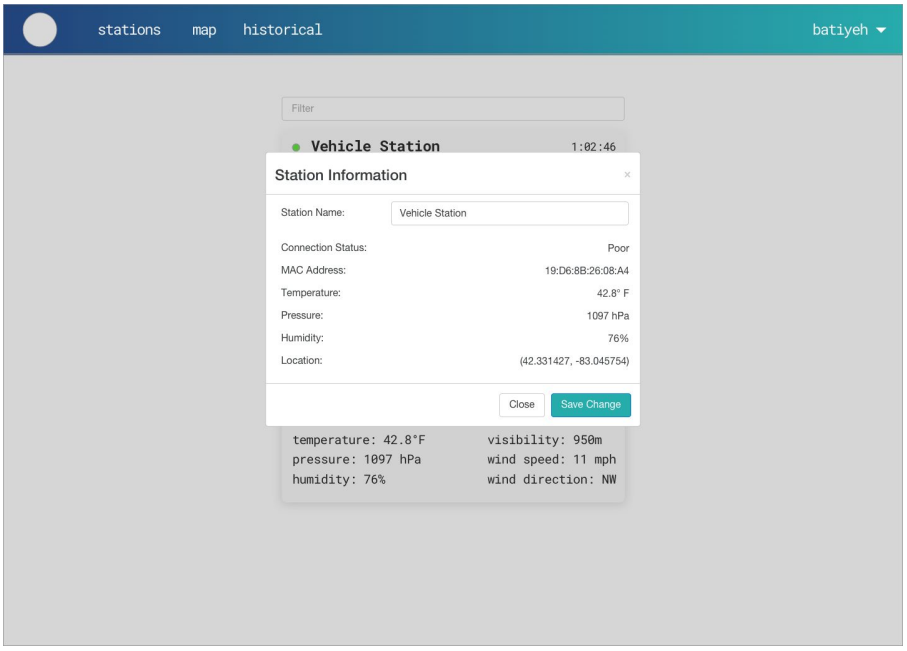


Figure 3.1.1-6: Connected Stations Additional Information Modal

The map page will show users exactly where stations are located using Google Maps. The user also has the option to filter stations by station name. After filtering, the user can view the locations of each station by checking the box next to each station name.

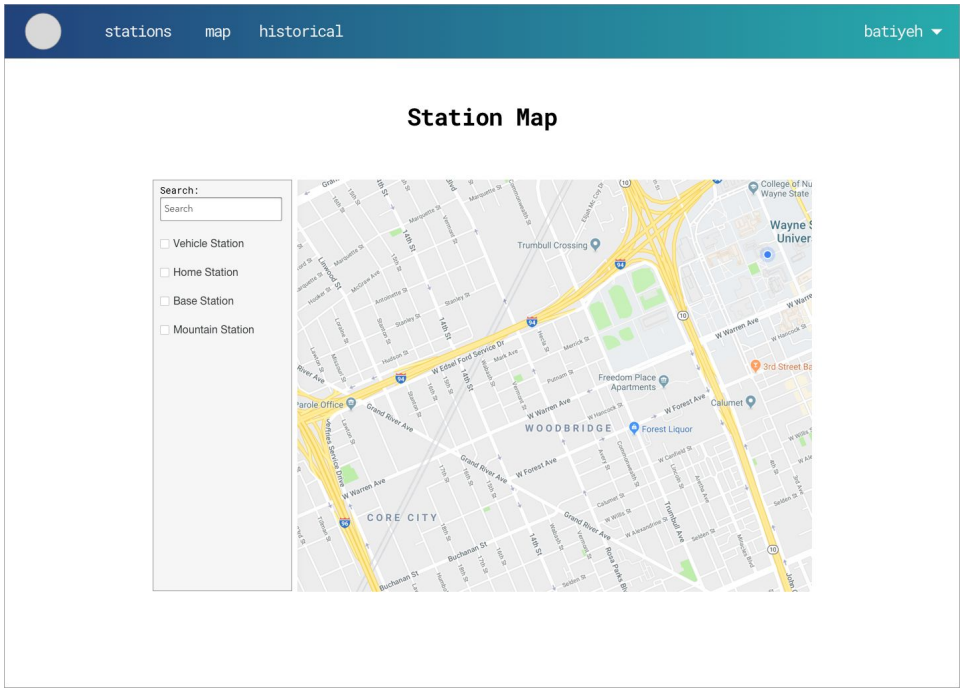


Figure 3.1.1-7: Station Map View

The user can navigate to the historical page by clicking the link on the right of the navigation bar. This page will display historical weather data from both the Open Weather API and from the weather stations. By clicking the filter button, the user can choose to display weather between two dates, the data type (API or station), and select individual or multiple stations.

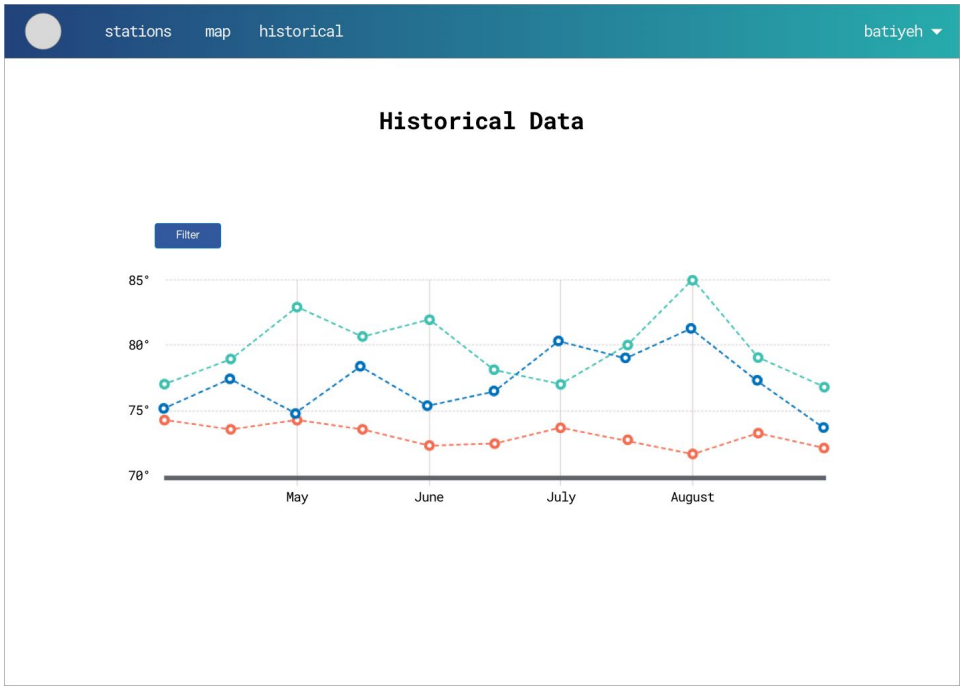


Figure 3.1.1-9: Historical Data View

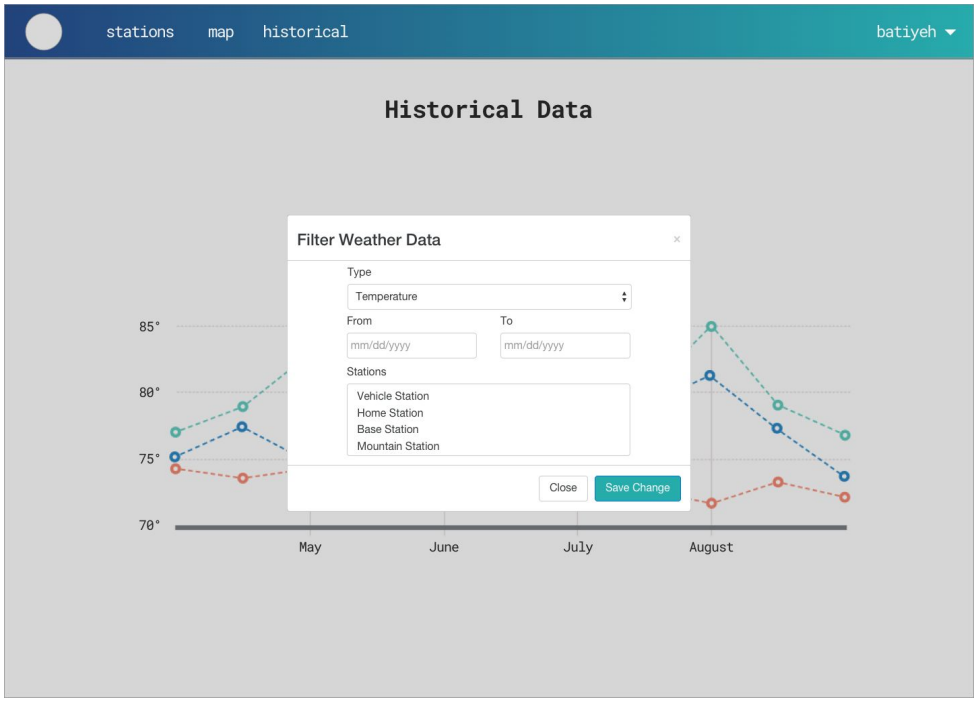


Figure 3.1.1-10: Historical Data Filter Modal

The user will be able to click their username in the top right hand corner of the navigation bar. This will open a dropdown which gives them the option to either view their user profile information or logout from the site. If they select “profile” they will be taken to a page which will display their currently stored information and allow them to edit any fields they wish.

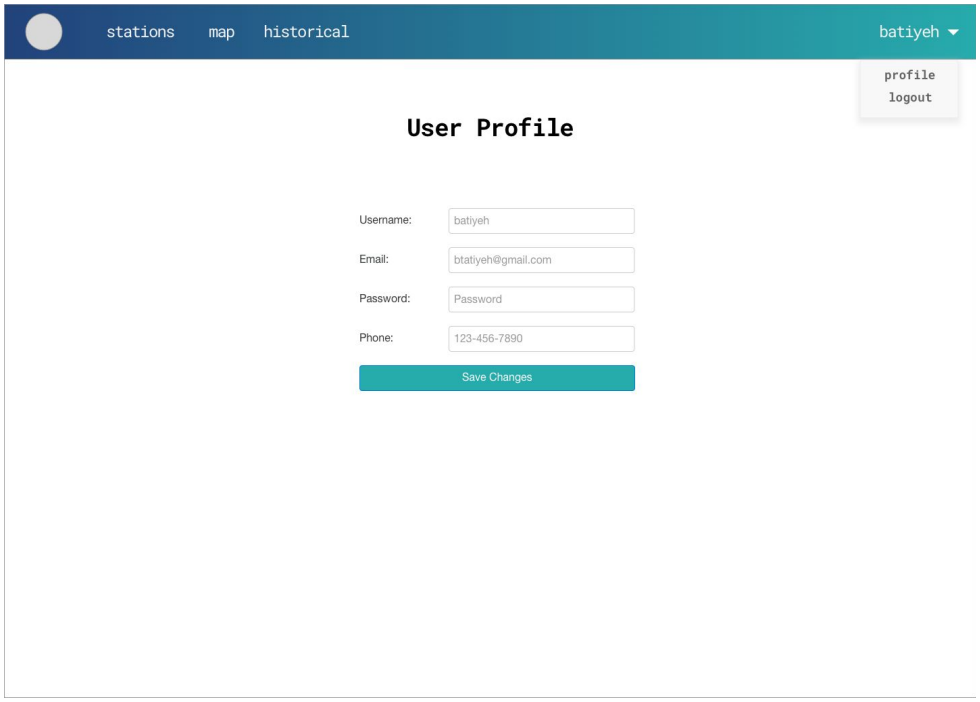


Figure 3.1.1-11: User Profile Page

3.1.2 Hardware Interfaces

The main component of the weather station is the Raspberry Pi with the Sense HAT that contains the sensors. The version of Raspberry Pi to be used is the three model B which includes a quad core 1.2GHz broadcom BCM2837 64 bit CPU, 1GB of RAM, BCM4348 on board wireless LAN, and 40-pin extended GPIO [1]. To store the operating system and data, a SanDisk Ultra 32GB microSDHC card is going to be utilized.

The Sense HAT containing all sensors to be used (thermometer, barometer, hygrometer, accelerometer, gyroscope) will connect to the Raspberry Pi directly on the 40-pin GPIO. The thermometers range is accurate to 2°C from 0-65°C (32°F-149°F). The accuracy and range of the hygrometer is accurate to 4.5% in the 20-80% rH. The barometer accuracy depends on the temperature, but under normal conditions its is 0.1 hPa with a range of 260-1260 hpa [2].

Attached via USB port will be a GPS dongle with a Chip u-blox 7020 chip to get the latitude and longitude of where the Raspberry Pi is. In the case that the station is unable to be connected to an electrical outlet, a battery pack will power it to make it portable.

The battery pack it will use is a 20000mAh Anker PowerCore II. With the sensors connected to the Raspberry Pi this should allow for approximately 20 hours of usage.

3.1.3 Software Interfaces

The weather station web application will be developed using Node.js, Express, React, and MySQL. Node is a backend framework built using Javascript which allows for nonblocking I/O. This means that when Node handles requests, the rest of the application can still run. Express is a Node.js library which handles HTTP requests and page routing. When a user navigates to the website, Express will find the correct route for the web page, Node will pull the necessary data from MySQL, and then pass it over to React. After receiving the data, React will render the interface of the page and display information to the user. The Raspberry Pi client code was developed using Python. Every three seconds, new data from the weather sensors is retrieved and sent to our web server to be stored in a MySQL database.

3.1.4 Communications Interfaces

The weather station application will be run on a web-based network server which relies on data coming in from external hardware and third party APIs. The web application is built around an API to handle HTTP requests. By accessing the website, users are initiating a request to pull data from the database and the Open Weather API before it is displayed. These HTTP requests are handled asynchronously with Node which means that the app can handle multiple requests at the same time. The Raspberry Pi will also communicate with the web server via HTTP requests. It will rely on pushing data to an asynchronous API endpoint on the web server. From this endpoint, it will be stored in a database for later retrieval.

3.2 Functional Requirements

FR-1	User Login
Description	Users will be able to log in from a login page using their own created account after it has been approved by an admin. A successful login will redirect the user to the connected stations page. If the account credentials are invalid, the user will see an error notification which prompts them to enter their credentials and try again.

FR-2	Create a User Account
Description	Users can click the *register* button on the login page to be redirected to the create account page. Once on the create account page, the user will see a form with a username, password, and email input. If the username or email already exists, the form submission will return an error prompting the user to try a different one. The form submission will also return an error if the password does not meet the password requirements (FR-3). If the user is successful in submitting the form, the user will be redirected to the login page and a request for access will be emailed to all admin users. Once the account access request is approved (FR-12), the user will be able to login (FR-1) with their credentials.

FR-3	Password Requirements
Description	Passwords must include at least one letter and one number. Passwords cannot include characters that are not letters or numbers. Passwords must be at least 8 characters in length.

FR-4	Reset a Password
Description	Users will be able to click the *Forgot Password?* link on the login page where they will be prompted to enter their email address. After entering their email address, they will be sent an email with a link to take them to a reset password page. If the email does not exist as an account, the user will be notified of the error on the email form. After clicking the reset link in their email, the user will be able to fill out the password form with their new password which they can use in combination with their username to login after successfully submitting the form.

FR-5	User Logout
Description	Logged in users should be able to click their username in the top right corner of the navigation bar. This will drop down a menu that displays the option to logout. After successfully logging out, the user will be redirected back to the login page.

FR-6	View Stations
Description	Logged in users will be able to view the latest weather from all stations. Each station will display last weather received time, temperature, pressure, and humidity. If the station's GPS coordinates allow for data from the Open Weather Map API to be obtained, it will also include wind speed, wind direction, and visibility. Each station should also display a connection quality indicator to the left of its name. This information will be updated every five seconds.

FR-7	View Station Details
Description	A user will be able to click on each station card on the connected stations page. This will bring up a window that displays the station's name, temperature, humidity, pressure, latitude, and longitude. This window also gives the user the ability to rename the station if they are an admin user.

FR-8	View Station Location
Description	Users will be able to view the location of a station located at the bottom of each connected station detail window. The location will displayed as a teal marker within the Google Maps instance.

FR-9	Connected Stations Map
Description	Upon navigation to the map page, the user will see a Google Maps instance and a column to the left of the map containing a list of station names. Each station name will have a checkbox next to it which the user can check to display each station's location on the map. Each station's location will be displayed via a teal pin on the map itself.

FR-10	Administrator Permissions
Description	The application will start with a single superuser. This superuser can add or remove admins through a drop down on a list of each individual user, change the name of stations, and approve account creation of regular users via email links. Newly created admins function the same as the superuser with the exception that they cannot revoke admin privileges.

FR-11	New User Account Approval
Description	When a new account is created, an email will be sent to all admin users. This email will contain a link to an approval page. On the approval page, the admin will see the username and email of the new account. The admin can then press either the *approve* or *deny* buttons. If *approve* is clicked, the user who created the account will receive an email letting them know their account is active. If *reject* is clicked, the user will receive an email informing them of the rejection.

FR-12	Name a Station
Description	Admin users will be able to change the name of each individual station from an input box at the top of the station details window. This name will remain with the station every time it is connected. Stations cannot have a blank name.

FR-13	View Historical Weather Data
Description	Users will be able to view historical weather data by navigating to the historical page. They will be able to click the filter button at the top of the graph. This button will open a window where the user can filter the stored data by date range, station, and data type. This data will be dynamically displayed via a line graph in the center of the page.

FR-14	User Profile
Description	Upon navigating to the user profile page, users will be able to view their username, email address, and phone number (if they have one associated with their account) via a form at the center of the page. From this form, the user can update their username, email address, and phone number by editing the value in the input boxes and clicking *save*

	changes. Users will also have the option to sign up for weather alerts on this page (FR-12).

FR-15	**Weather Alerts**
Description	Users will be able to sign up to receive inclement weather alerts. The user will be able to go to their profile page and enable alerts by checking the box that says *enable alerts*. The user will have the option to add alerts in the form of emails (FR-17), SMS (FR-18), and the webpage (FR-19). The user will be able to enter their own alert requirements by setting rules for temperature, humidity, or pressure changes over any period of time. Admin users will be able to set alert requirements for each individual station in the same way.

FR-16	**Email Alerts**
Description	If the weather condition alert requirements are met (FR-16), every user signed up for email alerts will receive an email notifying them of the weather and describing why the alert was sent.

FR-17	**SMS Alerts**
Description	If the weather condition alert requirements are met (FR-16), every user signed up for SMS alerts will receive an SMS to their phone number notifying them of the weather and describing why the alert was sent.

FR-18	**Webpage Alerts**
Description	If the weather condition alert requirements are met (FR-16), users will receive a notification at the top of the webpage describing the weather and why the alert was sent.

FR-19	**Station Connection Quality Indicator**
Description	Each station will display a green plug icon or red circle icon on the stations page which shows connection status. A green plug icon will mean that the station is currently connected. A red circle icon means that the station has not sent data in thirty seconds.

FR-20	Save Station Data Locally
Description	In the event of disconnection from the server, the Raspberry Pi client will continue to retrieve data from sensors every five seconds. This data will be stored locally via a text file until connection with the server comes back online. A new file will be created for storing data each day that there is no connection. Once connection is made again, the locally stored data will be pushed to the server.

FR-21	Filter by Station Name
Description	Users will be given the option to filter both the stations page and the map page by station name. This filter will update the list in a live fashion as the user types out the name of the station they wish to view.

FR-22	Connect Stations Automatically
Description	Once the weather station is turned on, it should begin to send data to the server automatically. If there is locally stored data, it will be sent to the server prior to sending new data every three seconds. If the weather station is unable to connect to the server, it will continue trying to connect every five minutes.

FR-23	Add a Station
Description	To add a new station, an admin will click the 'Add Station' button on the Admin page. The add station form will be displayed on the page where the admin will have to input the station's name and an optional expiration date. The station's API key is displayed in a disabled input at the bottom of the form. Once "Submit" is clicked, the station is added to the list.

FR-24	Install New Station
Description	The admin will download the weatherstation client on the Raspberry Pi by going to the admin page. Once downloaded, the admin will extract all the files in the downloaded zip file to a folder location of his choosing. Once extracted, the admin will follow the instructions listed in the install.txt file.

FR-25	Webpage Alerts Display

Description	When the user receives a webpage alert, the indicator in the top right will update to notify the user the number of unread webpage alerts they have. When the indicator is clicked, the indicator will return to is default icon and a dropdown will open and will be populated with a list of webpage alerts. At the top of the list is a button that will dismiss all alerts on this list. When an alert is clicked a modal with weather information at the time the alert was triggered will be populated. If the indicator is clicked when there are no alerts, the dropdown will open and the user will see a message stating that there are no webpage alerts.

FR-26	**Send Weather Data**
Description	The Raspberry Pi weather stations shall be sending weather data to the website asynchronously through an API endpoint every five seconds. The data being received by the website will be temperature, pressure, humidity, gps(if available) , and a timestamp of when the data was taken. If there is a loss of internet connection the station shall store the data onto a text file. Once the internet connection is restored it will send the stored weather data from the text file.

3.2.1 Stretch Goals

FR-27	**Average Weather Data**
Description	On the map page users will be able to get the average weather of stations in a selected area. Users will be able to click a button on the sidebar called average weather which will enable them to draw a circle around stations marked on the map. After drawing the circle the average temperature, pressure, and humidity will be displayed to the user in a pop up box. Users will also be able to draw multiple circles on the map at one time.

FR-28	**Historic Alerts**
Description	At the bottom of the alerts page a list of all triggered alerts will be populated. By default it will show only alerts triggered on the current day. Users can filter the alerts displayed on the list by alert and by date. When one of the historic alerts is clicked it will open a modal that will display the weather data of the station at the time it was triggered.

3.3 Non-Functional Requirements

3.3.1 Performance

NFR-1	Number of Connected Stations
Description	The site will be able to run up to twenty Raspberry Pi weather stations connected. Once twenty connections have been made, the site will no longer be guaranteed to perform optimally.

NFR-2	Navigating the Site
Description	Navigating from one page to another on the site should have a load time of no more than one second.

NFR-3	Performance of Querying Data
Description	To ensure the website pages are loading in no longer than one second, the result queries for data on any page shall be done in under one second.

3.3.2 Reliability

NFR-4	Website Reliability
Description	The website shall have an uptime of at least 99%. This will be assured by preventing crashes through best practices such as error handling and gracefully closing connections.

3.3.3 Availability

NFR-5	Website Accessibility
Description	The website will be available strictly for the Department Of Defense. If a user wishes to create an account to access the site, they will need the approval of an administrator before the account can be used to login.

NFR-6	Battery Life of Raspberry Pi

Description	In the event that the Raspberry Pi Weather Station needs to be portable, it will support the use of a battery pack which will provide 16-20 hours of uptime.

3.3.4 Security

NFR-7	**Security of User Data**
Description	User passwords will be encrypted with bcrypt and stored in a database. Passwords will never be stored in plain text.

NFR-8	**Security for Cross-Site Forgery Request Attacks**
Description	To prevent cross-site forgery request attacks, we will use CSRF tokens when sending form requests in order to maintain security of the site.

NFR-9	**Security for SQL Injection Attacks**
Description	To prevent the database from being compromised we will be protecting against SQL injection attacks. This will be accomplished with Knex in Node.

3.3.5 Maintainability

NFR-10	**Website Maintainability**
Description	Before an updated version of the site is pushed, it must be tested for errors. If show stopping bugs are found post release, it must be reverted to the previous working version.

3.3.6 Portability

NFR-11	**Web Browser Accessibility**
Description	The site will support four modern browsers with their most up to date versions. For a specific list of browsers and versions, see section 2.5 of this document.

NFR-12	Mobile Web Accessibility
Description	The website has design support for mobile browsers. It will support any mobile devices that are running Android 8.0+ or iOS 10.3+. These devices must visit the website through the Google Chrome or Safari web browser apps.

3.4 Design Constraints

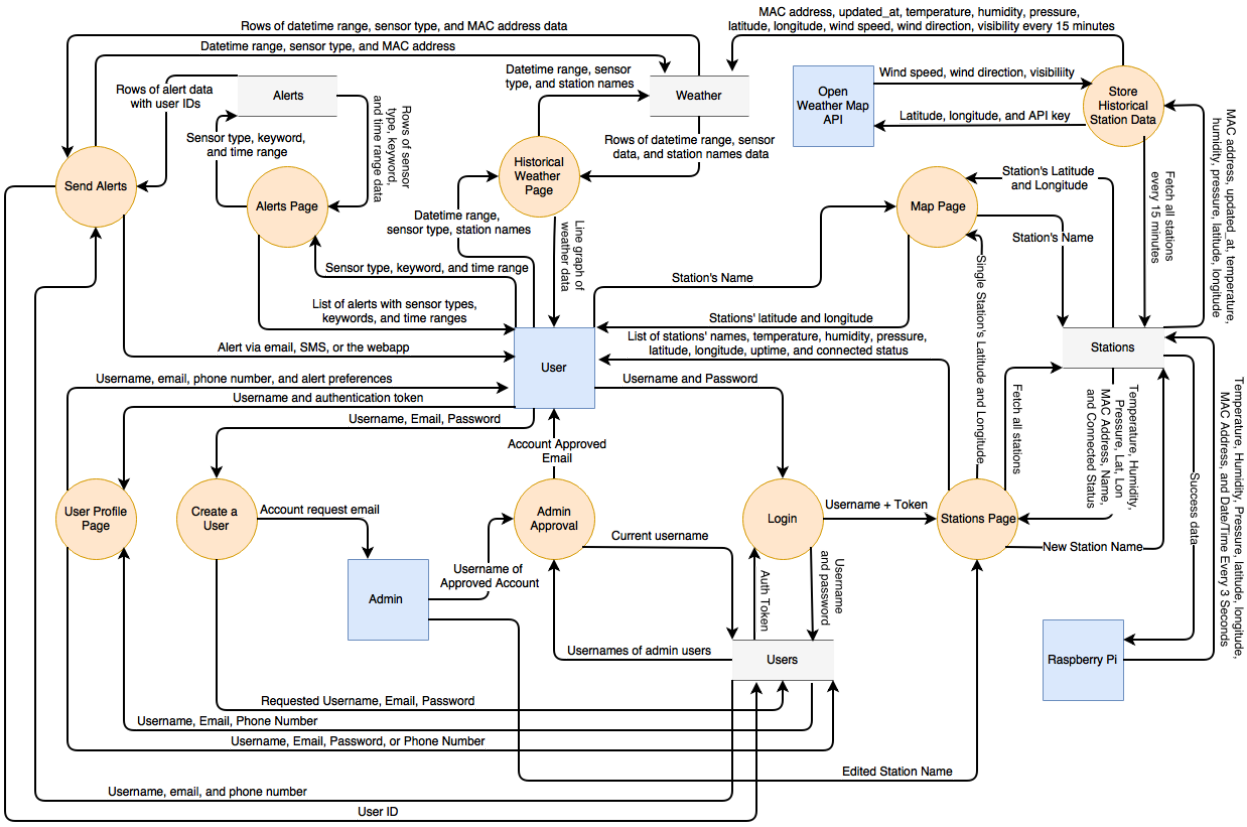
This project will mainly be constrained by the API call limits of third parties. The Google Maps API, which will be used to display the locations of all connected stations, allows for 25,000 Javascript API calls per day [3]. For the station map page, this means that each time a station's location is marked on the map it will use up one Javascript API call. For loading the map itself, Google Maps allows for 2,000,000 map loads per day [3]. The Open Weather Map API, which will be used to pull and combine additional weather data (wind speed, direction, visibility, and weather description) with the stations, allows for sixty API calls per minute [4]. The Gmail API which will be used to send email alerts to users allows for 1,000,000,000 quota units per day [5].

3.5 Logical Database Requirements

The weather station website will use MySQL as the primary database management tool. Relational databases will be used to store information such as username, password, station name, weather data at a point in time. This is a read/write heavy application as the weather stations are writing data to the database every 3 seconds and that data is being read and displayed to a page every 3 seconds. Every 15 minutes the data that is currently stored is saved permanently for historical purposes. To interface with SQL, we will also be using Bookshelf.js, which is a Javascript ORM for Node built on the Knex SQL query builder. This technology will help us to more easily interact with the database.

4 Analysis Models

4.1 Data Flow Diagram



A. Appendices

A.1 Appendix 1

Below is a traceability matrix to keep track of all requirements completeness.

	Requirement ID	Requirement Name	Use Case ID	Priority	Test Case ID
Requirements	FR-1	User Login	UC-2	High	TC-1
					TC-2
					TC-3
					TC-4
	FR-2	Create a User Account	UC-1	High	TC-5

					TC-6
	FR-3	Password Requirements	UC-4	Moderate	TC-7
					TC-8
					TC-9
					TC-10
					TC-11
					TC-12
					TC-13
					TC-14
					TC-15
	FR-4	Reset a Password	UC-4	Moderate	TC-16
					TC-17
					TC-18
					TC-19
					TC-20
					TC-21
					TC-22
					TC-23
					TC-24
	FR-5	User Logout	UC-3	High	TC-25
	FR-6	View Stations	UC-15	High	TC-26
					TC-27
					TC-28
					TC-29
					TC-30
					TC-31
			UC-25	High	TC-123
					TC-124

	FR-7	View Station Details	UC-17	Moderate	TC-32
					TC-33
					TC-34
	FR-8	View Individual Station Location	UC-23	Low	TC-35
					TC-36
	FR-9	View Stations Map	UC-21	High	TC-37
					TC-38
					TC-39
			UC-22	Moderate	TC-40
					TC-41
					TC-42
					TC-43
					TC-44
					TC-45
	FR-10	Administrator Permissions	UC-12	High	TC-46
					TC-47
			UC-13	High	TC-48
	FR-11	New User Account Approval	UC-5	High	TC-49
					TC-50
					TC-51
					TC-52
					TC-53
	FR-12	Edit Station Name	UC-18	Moderate	TC-54
					TC-55
					TC-56
	FR-13	View Historical Weather Data	UC-19	High	TC-57
					TC-58
	FR-13	Filter Historical Weather Data	UC-20	High	TC-59

					TC-60
					TC-61
					TC-62
					TC-63
					TC-64
					TC-65
					TC-66
					TC-67
					TC-68
			UC-6	Low	TC-69
					TC-70
					TC-71
					TC-72
					TC-73
					TC-74
					TC-75
					TC-76
	FR-14	User Profile	UC-7	Low	TC-77
					TC-78
					TC-79
					TC-80
					TC-81
					TC-82
					TC-83
			UC-8	High	TC-84
			UC-10	High	TC-85
	FR-15	Weather Alerts	UC-11	High	TC-86
	FR-16	Email Alerts	UC-9	High	TC-87

					TC-88
			UC-14	High	TC-89
					TC-90
			UC-9	High	TC-91
	FR-17	SMS Alerts	UC-14	High	TC-92
					TC-93
			UC-9	High	TC-94
	FR-18	Webpage Alerts	UC-14	High	TC-95
					TC-96
		Station Connection Quality			TC-97
	FR-19	Indicator	UC-15	Moderate	TC-98
					TC-99
	FR-20	Save Station Data Locally	UC-26	High	TC-100
					TC-101
	FR-21	Filter by Station Name	UC-16	Low	TC-102
					TC-103
		Connect Stations			TC-104
	FR-22	Automatically	UC-27	High	TC-105
					TC-106
	FR-23	Add a Station	UC-24	High	TC-107
					TC-108
					TC-109
					TC-127
					TC-128
	FR-24	Install new station	UC-28	High	TC-129
					TC-110
					TC-111
	FR-25	Webpage Alerts Display	UC-29	Moderate	TC-112

					TC-113
					TC-114
					TC-115
	FR-26	Send Weather Data	UC-30	High	TC-125
					TC-126
Stretch Goals	FR-27	Average Weather Data	UC-31	Low	TC-116
					TC-117
					TC-118
	FR-28	Historic Alerts	UC-32	Low	TC-119
					TC-120
					TC-121
					TC-122
		Multi weather type alerts		Low	