Project Name: WeatherPal

Milestone 3:
Project Implementation Report

Group: CodingSquad

Ronald Jaglal	816003245
Jose Bravo Mata	816011156
Vinod Lochan Dassrath	816117511

Description of Technology

Software:

- Operating System on Raspberry Pi Weather Station:
  - Raspbian

- Markup/Style Sheet/Programming Languages:
  - HTML5
  - CSS
  - JavaScript
  - Python v3

- Frameworks/Libraries:
  - Material Design Bootstrap

- Modules/Packages:
  - Python Modules/Packages
    - bme280.py:

      ```
      sudo pip3 install RPi.bme280
      ```

    - adafruit_veml6070.py:

      ```
      sudo pip3 install adafruit-circuitpython-veml6070
      ```

- Database:
  - Google Cloud Firestore

Hardware:

- Raspberry Pi 3 B+

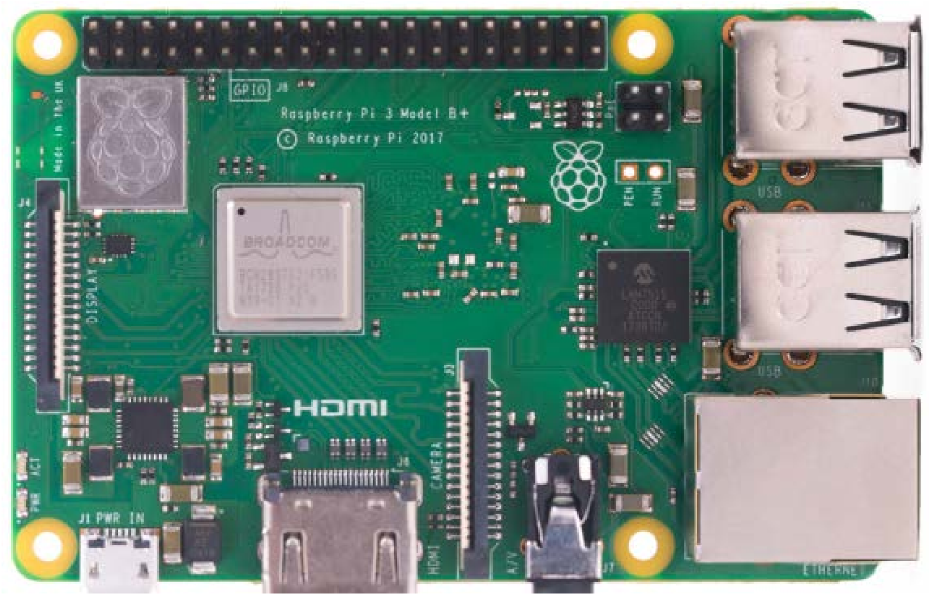


Figure 1.0: Showing Raspberry Pi 3 B+ computer

“The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

What’s more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work”.

Source: The Raspberry Pi Foundation

Specifications: Raspberry Pi 3 Model B+

- Adafruit BME280 Sensor

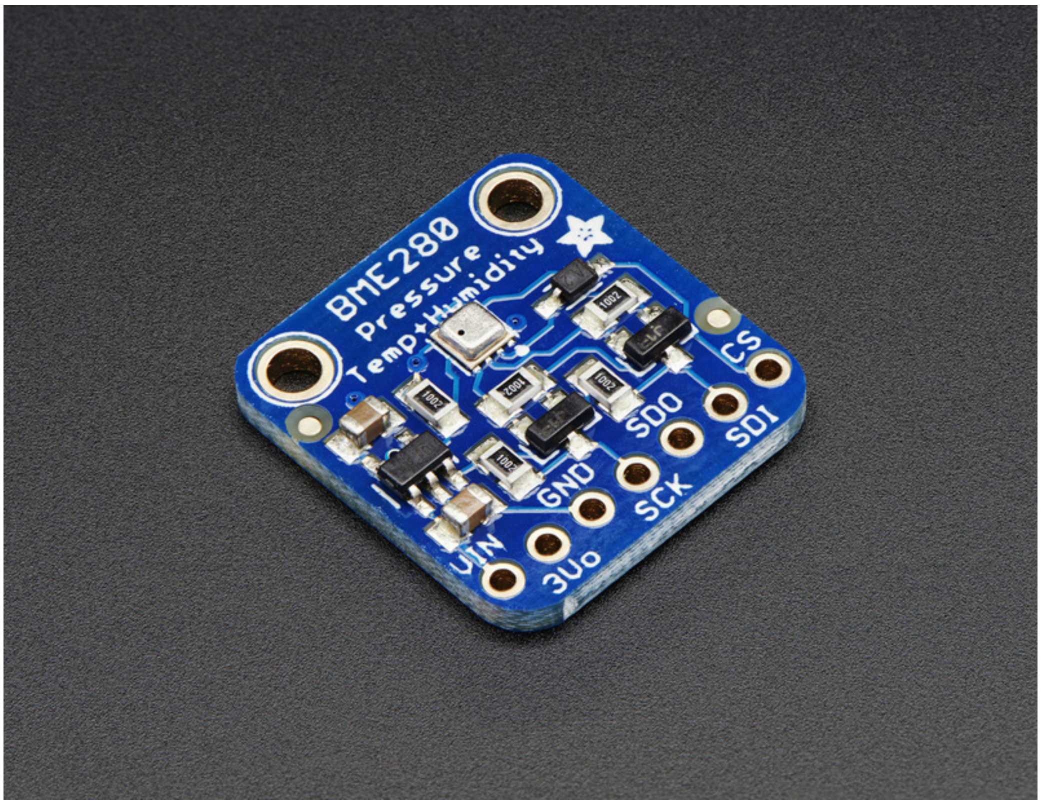


Figure 1.1: Showing BME280 sensor from Adafruit

The BME280 is a digital environmental sensor for humidity, barometric pressure, and ambient temperature developed by Bosch Sensortec. The variant used in this project was purchased from Adafruit. It can be connected via the Inter-Integrated Circuit (I2C) bus to the Raspberry Pi.

“This precision sensor from Bosch is the best low-cost sensing solution for measuring humidity with ±3% accuracy, barometric pressure with ±1 hPa absolute accuracy, and temperature with ±1.0°C accuracy.” More info (includes pinout, wiring and sample python code for I2C connection).

Specifications: BME280 Datasheet

- Adafruit VEML6070 Sensor

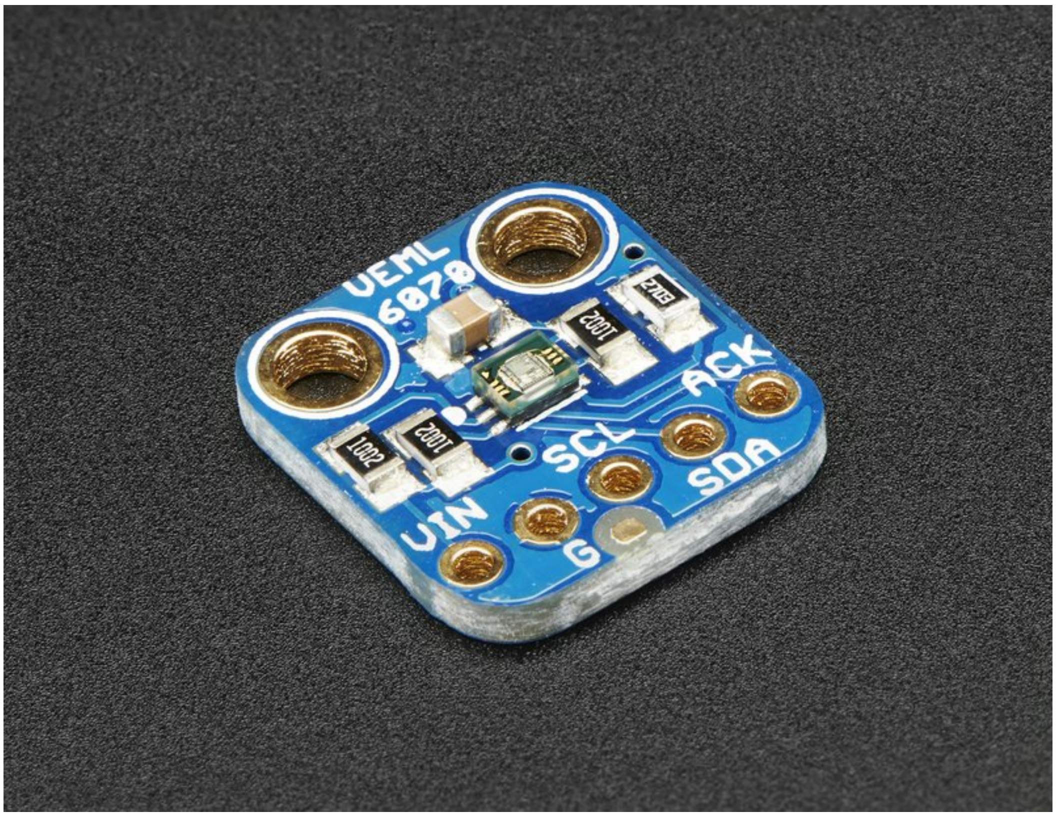


Figure 1.2: Showing VEML6070 sensor from Adafruit

The VEML6070 is digital UV sensor developed by Vishay. The variant used in this project was purchased from Adafruit. Like the BME280, it can be connected via the Inter-Integrated Circuit (I2C) bus to the Raspberry Pi. More info (includes pinout, wiring and sample python code for I2C connection).

Specifications: VEML6070 Datasheet

- SDS011 Nova PM Sensor

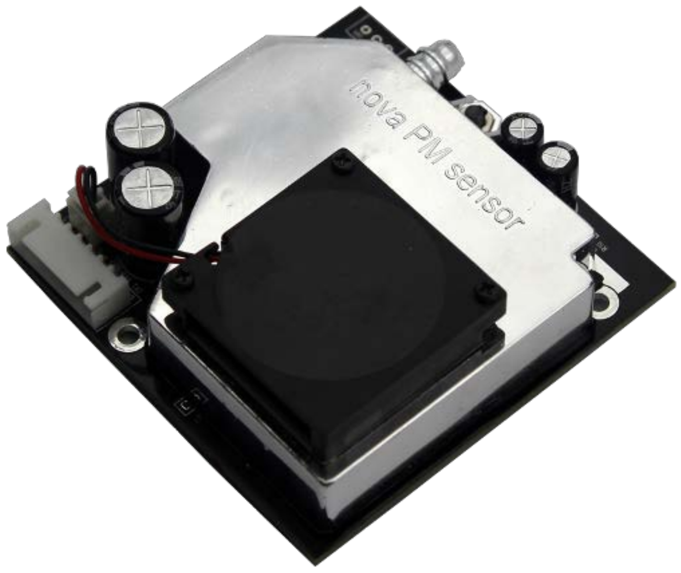


Figure 1.3: Showing SDS011 Nova PM Sensor

The SDS011 Nova PM sensor is a Particulate Matter or Particle Pollution sensor that has a measuring range 0.0 to 999.9 $\mu g/m^3$, response time of one second and a relative 15% margin of error ($\pm$ 10$\mu g/m^3$). It can connect the Raspberry Pi over USB.

It gives output using the standard PM2.5 and PM10 levels. PM10 levels include particles generally with diameters 10 micrometers and smaller in size ($\leq$ 10 μm) like pollen, dust, mold etcetera while PM2.5 levels include only finer particles with diameters that are generally 2.5 micrometers and smaller ($\leq$ 2.5 μm), i.e. about 3% the diameter of the human hair, such as smoke from fires or particles from a car exhaust. The following table shows a simple guide on interpreting the effect of daily average PM10 and PM2.5 levels our health.

PM10 (µg/m3) *(Low - High)*	PM2.5 (µg/m3) *(Low - High)*	Air Quality	
		Index *(Low - High)*	Category
0 - 54	0.0 - 12.0	0 - 50	Good
55 - 154	12.1 - 35.4	51 - 100	Moderate
155 - 254	35.5 - 55.4	101 - 150	Unhealthy for Sensitive Groups
255 - 354	55.5 - 150.4	151 - 200	Unhealthy
355 - 424	150.5 - 250.4	201 - 300	Very Unhealthy
425 - 504	250.5 - 350.4	301 - 400	Hazardous
505 - 604	350.5 - 12.0	401 - 500	

Table 1.0: Showing Air Quality Index (AQI) for ranges of measured level of concentrations of particulate matter

Specifications: Laser PM2.5 Sensor SDS011

- Adafruit MCP3008 ADC

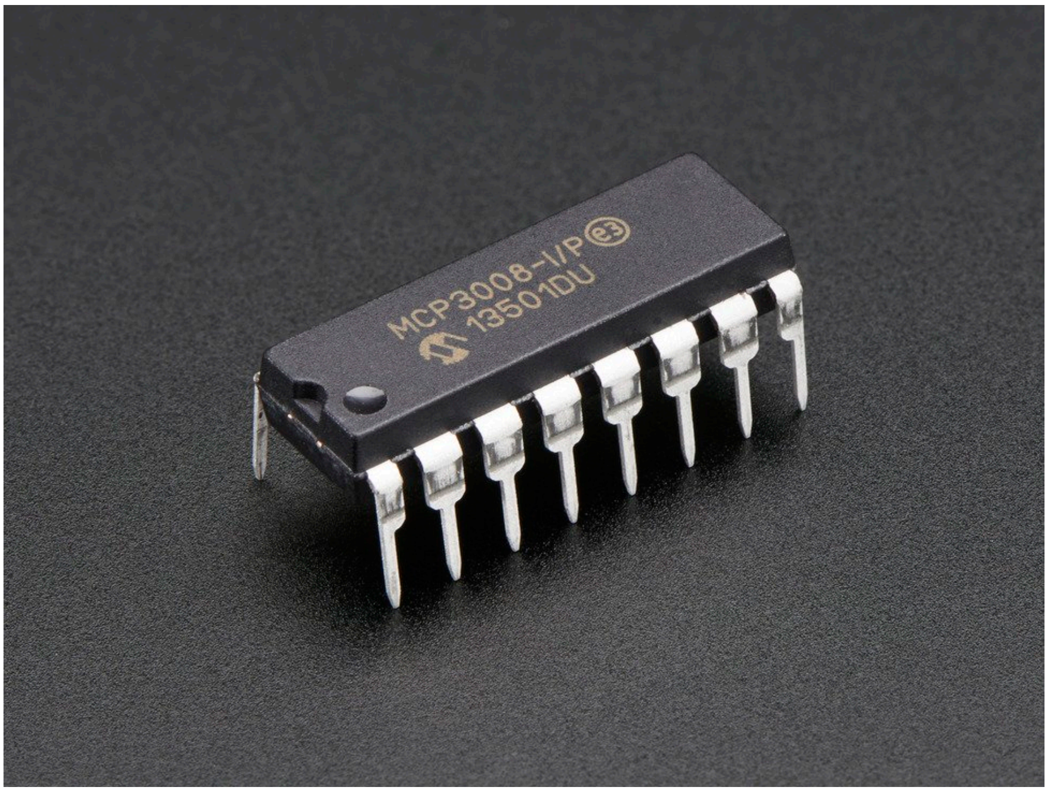


Figure 1.4: Showing MCP3008 ADC IC from Adafruit

The MCP3008 purchased from Adafruit is an Analogue to Digital Converter (ADC) delivered in the form of a 16-pin integrated circuit (IC). It uses Serial Peripheral Interface (SPI) to connect to the Raspberry Pi.

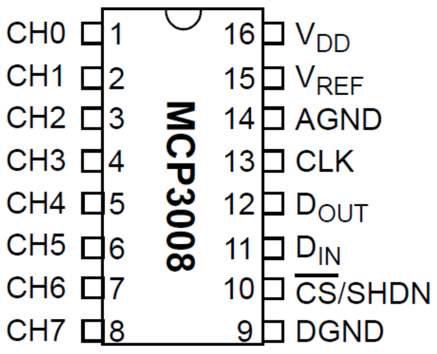


Figure 1.5: Showing pinout for MCP3008 ADC IC

It has 8 channels and each can convert any analogue input into a 10-bit digital output and therefore can detect very precise changes in voltages i.e. $\frac{x}{2^{10}}V \approx x\,mV$ so roughly down to a few millivolts as the maximum input/output voltages of the Raspberry Pi GPIO pins are 3.3V.

Its use in the project will be discussed in later on in the wind vane section (see page 9)

Specifications: MCP3008 Datasheet

- Argent Data Systems Wind/Rain Sensor Assembly

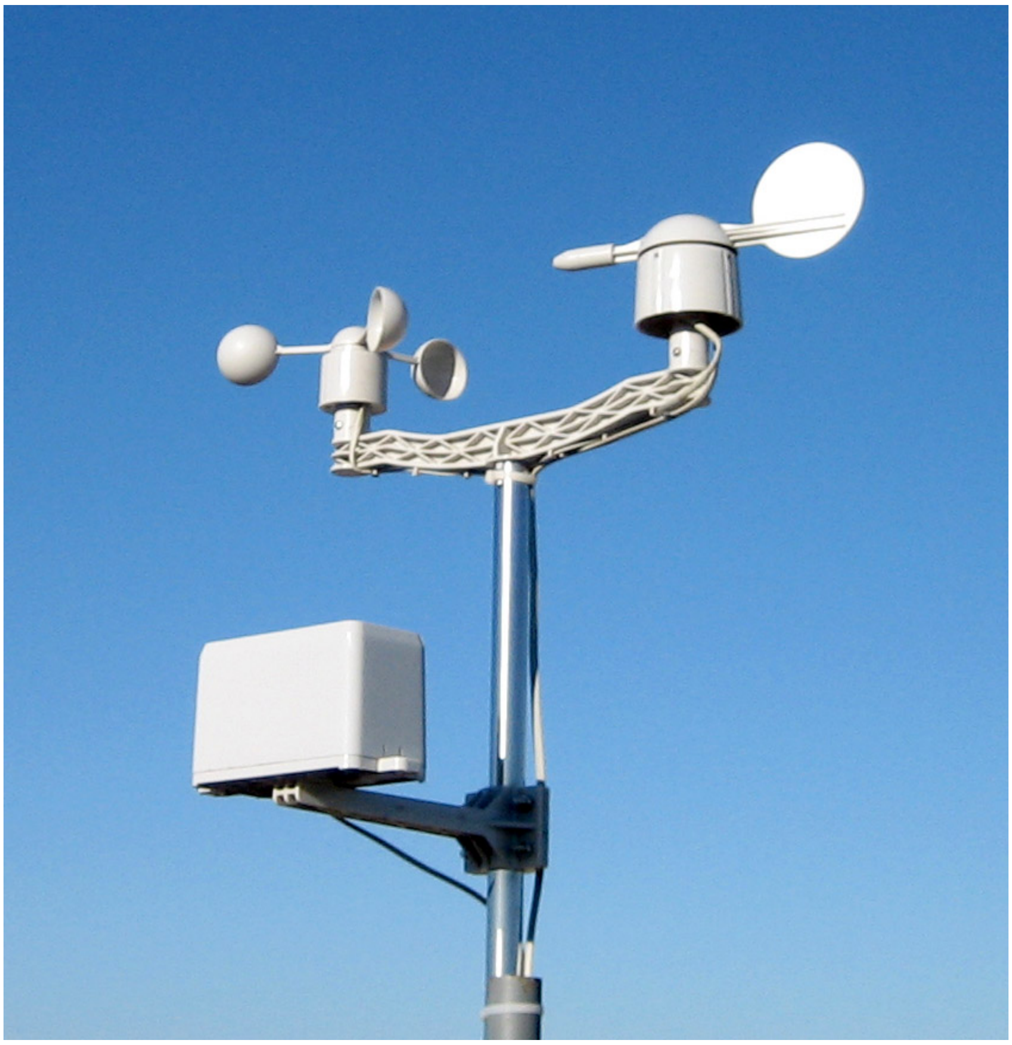


Figure 1.6: Showing anemometer, wind vane and rain gauge instruments from Argent Data Systems

This package of weather instruments from Argent Data Systems includes three mechanical sensors i.e. an anemometer, a wind vane and rain gauge. They come with RJ11 connectors making connections simple and secure.

- Anemometer

The anemometer contains a reed switch inside that is activated once every half rotation and twice every full rotation. By connecting one end of the circuit to a GPIO pin and the other to the Ground pin on the Raspberry Pi, we can detect a signal every instance the circuit is closed. The radius of the circular path made by a signal rotation of the anemometer cups is 9cm or $9 * 10^{-5}$km.

The wind speed is calculated from the distance a point on the circular path would travel in a fixed time as the effect of the wind causes the cups to rotate. This distance with one full rotation would be the circumference of the circular path.

So wind speed (S_w) in km/h is calculated as follows:

$$S_w = \frac{Distance\ Travelled}{Time\ Elapsed\ in\ Hours}$$

$$= \frac{Full\ Rotations\ *\ Circumference}{Time\ Elapsed\ in\ Hours}$$

$$= \frac{\left(\frac{Signals\ Detected}{2}\right) * \left[\left(2\pi * \left(9 * 10^{-5}\right)\right)\right]}{Time\ Elapsed\ in\ Hours}$$

- Wind Vane

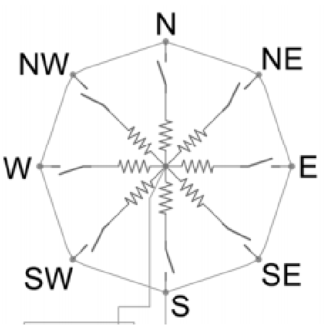


Figure 1.7: Showing circuit of reed switches and resistors for the wind vane

The wind vane has eight reed switches as depicted in figure x above. Each reed switch is arranged with a resistor such that when the wind vane rotates to point in the direction the wind is heading, either one or 2 of the switches can be closed at a particular instance. Therefore it allows for a maximum of 16 different resistance values that each correspond to a particular direction.

Direction (Degrees)	Resistance (Ohms)	Voltage (V=5v, R=10k)
0	33k	3.84v
22.5	6.57k	1.98v
45	8.2k	2.25v
67.5	891	0.41v
90	1k	0.45v
112.5	688	0.32v
135	2.2k	0.90v
157.5	1.41k	0.62v
180	3.9k	1.40v
202.5	3.14k	1.19v
225	16k	3.08v
247.5	14.12k	2.93v
270	120k	4.62v
292.5	42.12k	4.04v
315	64.9k	4.78v
337.5	21.88k	3.43v

Table 1.1: Showing reference values for reading wind direction from the signals outputted by the wind vane.

The table above shows what direction in degrees the wind is heading, the corresponding output resistance and output voltage in column one, column two and column three respectively if a circuit is set up with a reference input of 5V at 10kΩ of resistance.

However since we would be working with 3.3V from the GPIO pin on the Raspberry Pi, we would have to setup a voltage divider circuit with our input voltage (V_s) as 3.3V, one resistor (R_2) connected as seen the diagram below and then we can determine the output voltage (V_{out}) the wind vane should produce for each particular output resistance combination (R_1) that corresponds to a particular wind direction from the table.

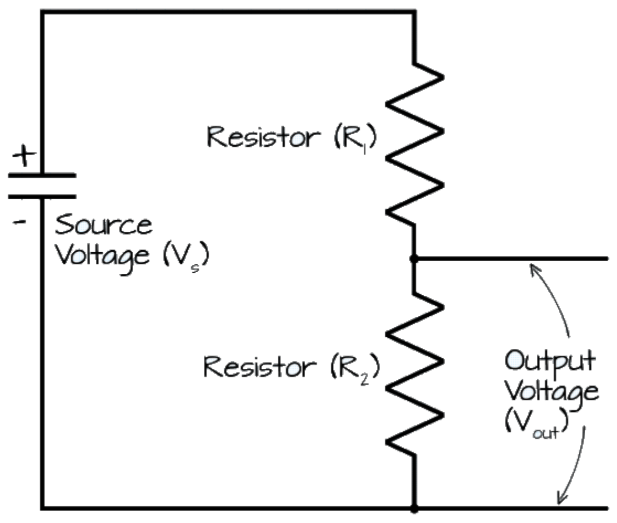


Figure 1.8: Showing circuit diagram for a voltage divider.

The formula for calculating output voltage from a voltage divider circuit (V_{out}) is:

$V_{out} = \frac{V_S + R_2}{(R_1 + R_2)}$

So for reference if we would like to know the output voltage when the wind direction is towards 0 degrees (cardinal direction, North) we can calculate as follows:

Note for R_2 we are using a standard $4.7k\Omega$ resistor as it gave sufficient distance between output voltages we would be working with to accurately determine wind direction.

$$V_{out} = \frac{V_S + R_2}{(R_1 + R_2)}$$

$$V_{out} = \frac{3.3V + 4.7k\Omega}{(33k\Omega + 4.7k\Omega)} = \frac{3.3 + 4700}{(33000 + 4700)} = 0.4\text{V}$$

The following table shows all the calculated output voltages and their corresponding wind directions for our wind vane circuit.

V_{out} (Volts)	Wind Direction	
	Degrees	Cardinal
0.4	0.0	N
1.4	22.5	NNE
1.2	45.0	NE
2.8	67.5	ENE
2.7	90.0	E
2.9	112.5	ESE
2.2	135.0	SE
2.5	157.5	SSE
1.8	180.0	S
2.0	202.5	SSW
0.7	225.0	SW
0.8	247.5	WSW
0.1	270.0	W
0.3	292.5	WNW
0.2	315.0	NW
0.6	337.5	NNW

Table 1.2: Showing output voltages from our wind vane and voltage divider circuit that correspond to particular wind directions.

For these analogue voltages to be read into the Raspberry Pi, the MCP3008 ADC is used to convert them to digital signals. The voltages outputted by the wind vane may not always correspond exactly to those seen above since there might be small amounts of added resistance when the wind vane jiggles. However we can round the voltages with good precision from the 10-bit values from the ADC so that they correspond to a particular voltage and hence wind direction from the table above.

- Rain Gauge

  The rain gauge contains 2 self-emptying, tipping buckets mounted on a pivot like a see-saw that requires 0.2794 mm of collected rainfall to tip to either side. When tipped to any side, a reed switch activates and the bucket empties. Then as another 0.2794 mm of rainfall collects on the bucket on the opposite side, the bucket tips to that end and activates another reed switch on that end. From this continuous process, we can have one end of the rain gauge circuit connect to the Ground and the other on a free GPIO pin on the Raspberry Pi and every time a signal is detected, we would know 0.2794 mm of rainfall has been collected. To calculate daily rainfall, $Rainfall_{daily}$ the following formula is used:

$$Rainfall_{daily} = Signals\ Detected\ in\ a\ day *\ 0.2794\text{mm}$$

Specifications: Wind/Rain Sensor Assembly Datasheet

- MicroSD Card

  - For the Raspberry Pi a readily available 16GB microSD card of no particular preference was used however maybe with a better card there can be some performance boost particularly in boot time when the weather station loses and regains power. However performance is limited by the microSD bus on the Raspberry Pi which is capped at 20 MB/s write speeds at normal and 40MB/s when overclocked.

- USB over RJ45 Extender

  - Used with a 5V 1A power adapter and RJ45 cable to extend the distance between the SDS011 Nova PM sensor (see page 6).

- RJ11 Breakout Boards

  - Used with RJ11 cables to connect the Wind/Rain sensors as well as the I2C sensors.

- 4.7KΩ resistor

  - Used in our voltage divider circuit for the wind vane (see page 9).

- Adafruit Perma-Proto HAT – with EEPROM

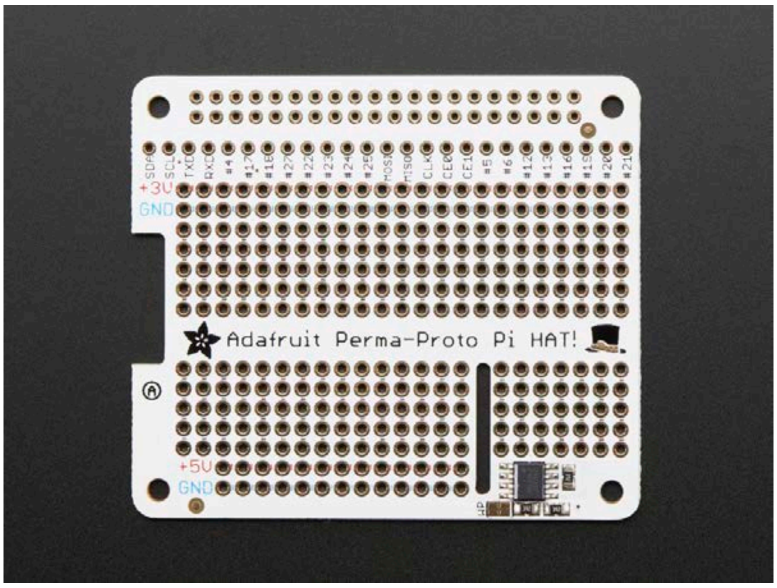
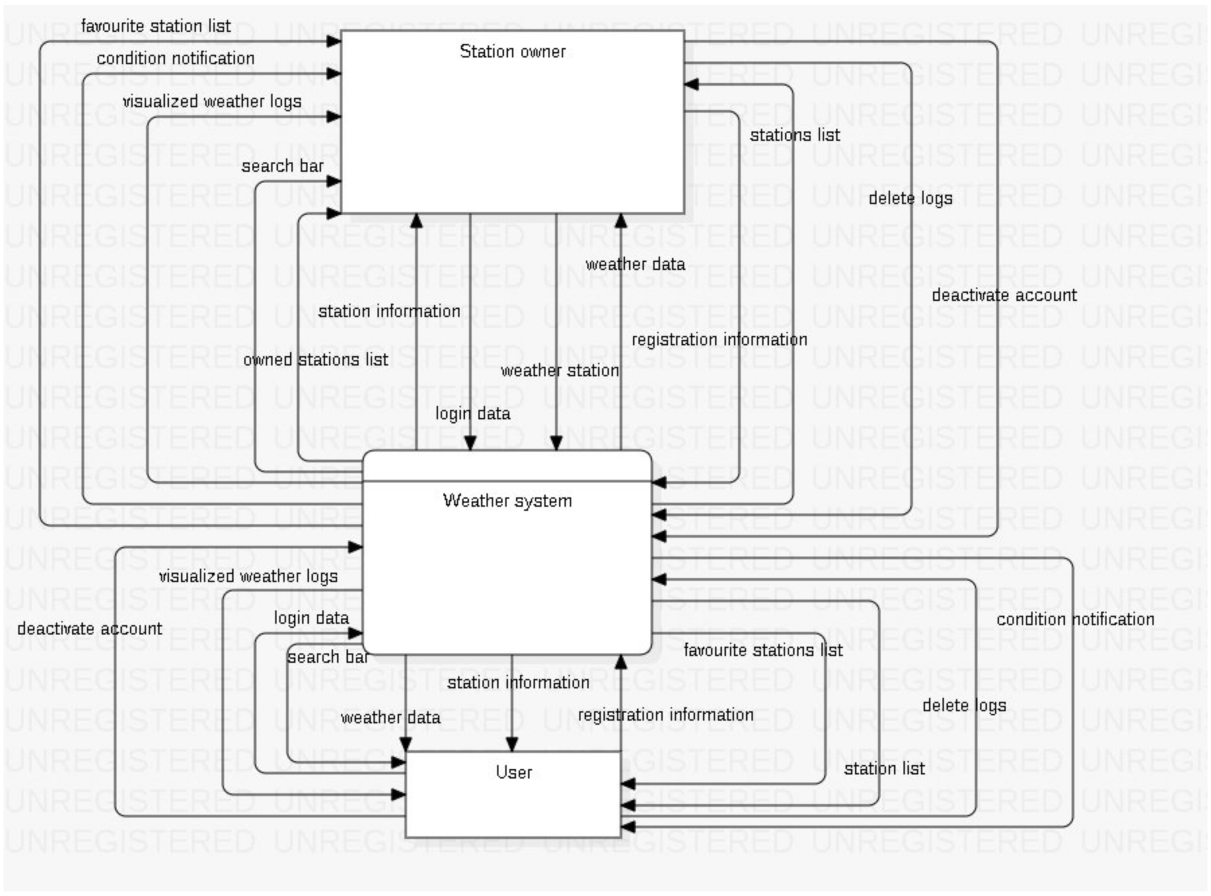


Figure 1.9: Showing Adafruit Perma-Proto HAT

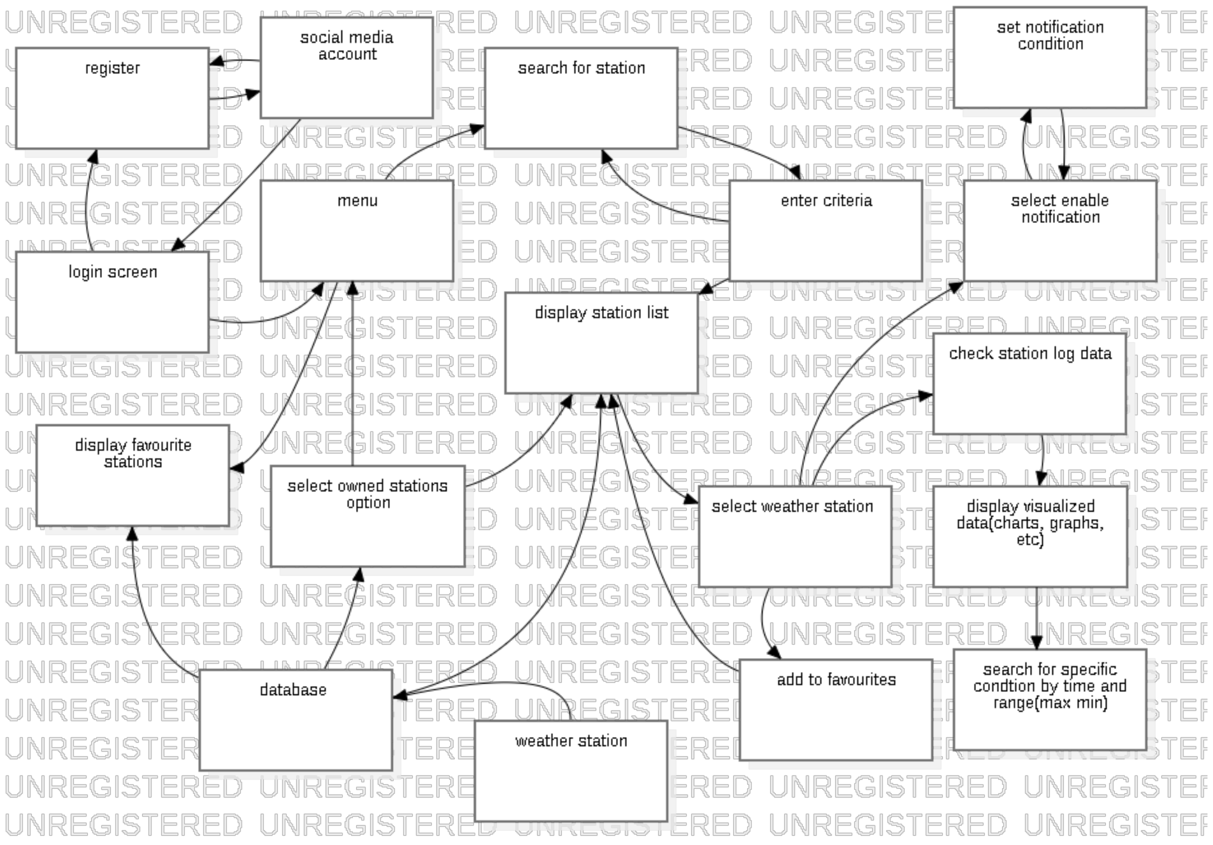
➢ Used with GPIO Female Header to make sensor connections to the Pi simple and easy. It attaches to the GPIO pins and can be removed when needed.

- GPIO Female Header

System Context Diagram

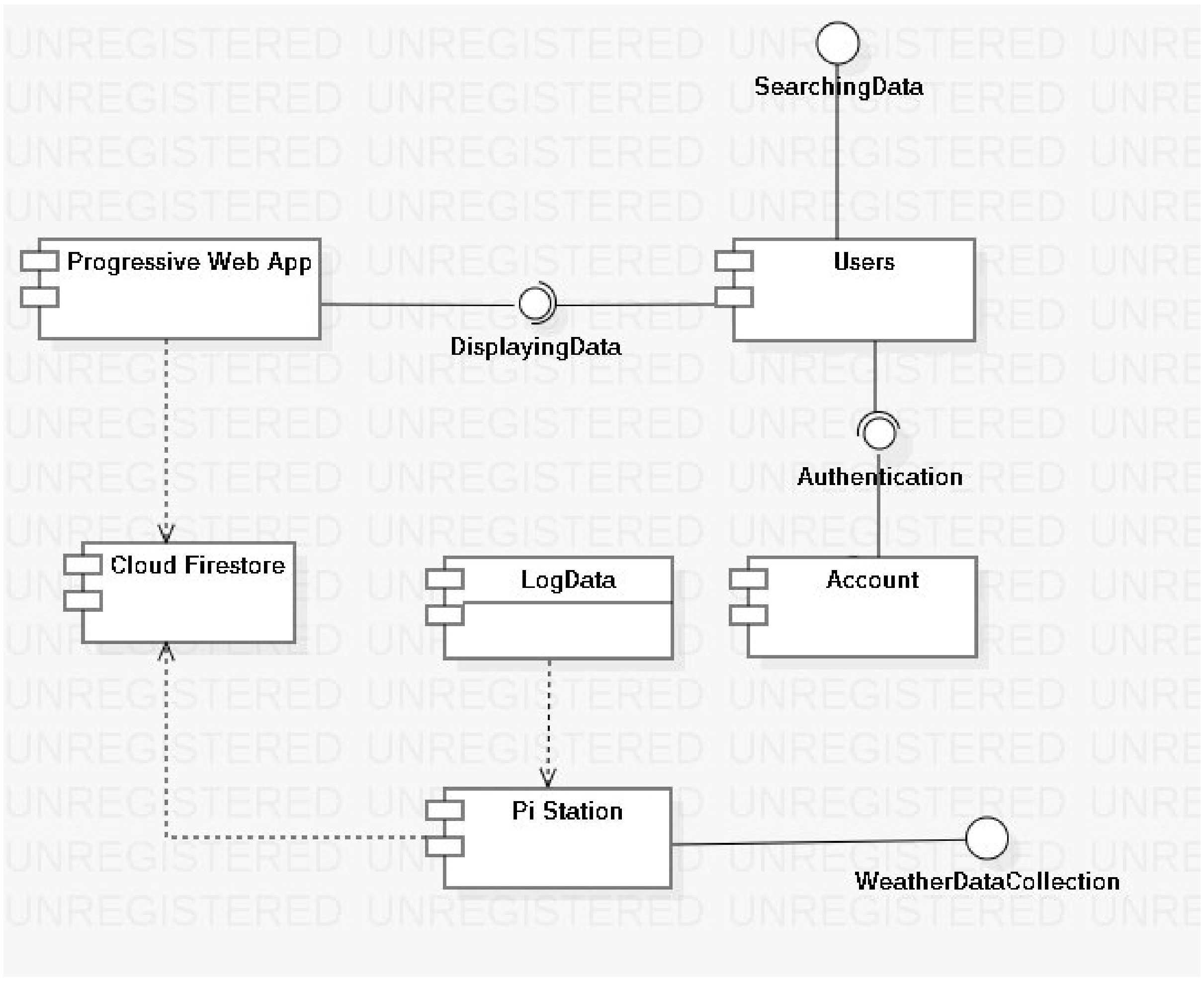


System Diagram

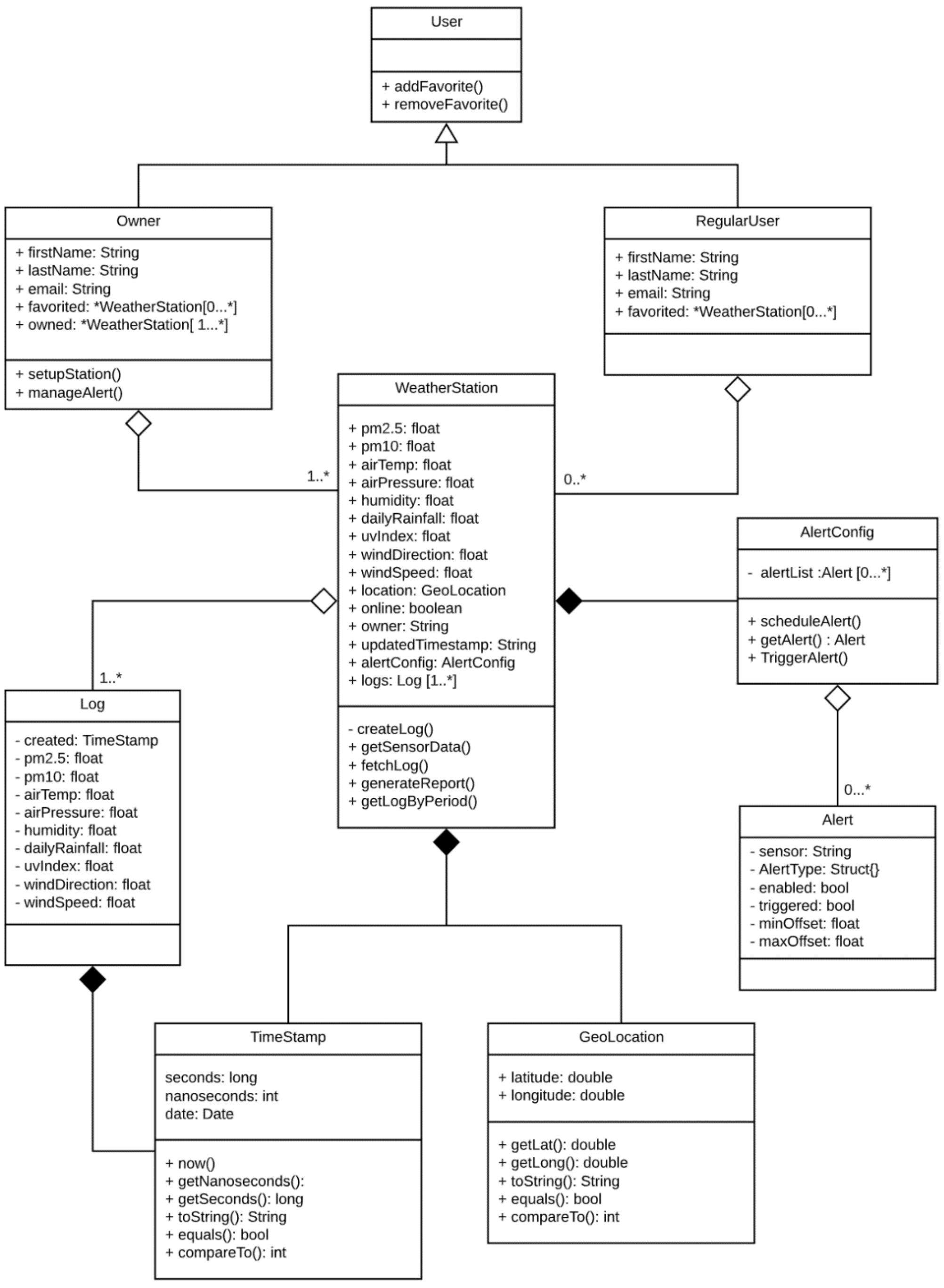


register
social media account
search for station
set notification condition
menu
enter criteria
select enable notification
login screen
display station list
check station log data
display favourite stations
select owned stations option
select weather station
display visualized data(charts, graphs, etc)
database
add to favourites
search for specific condtion by time and range(max min)
weather station

Component Diagram

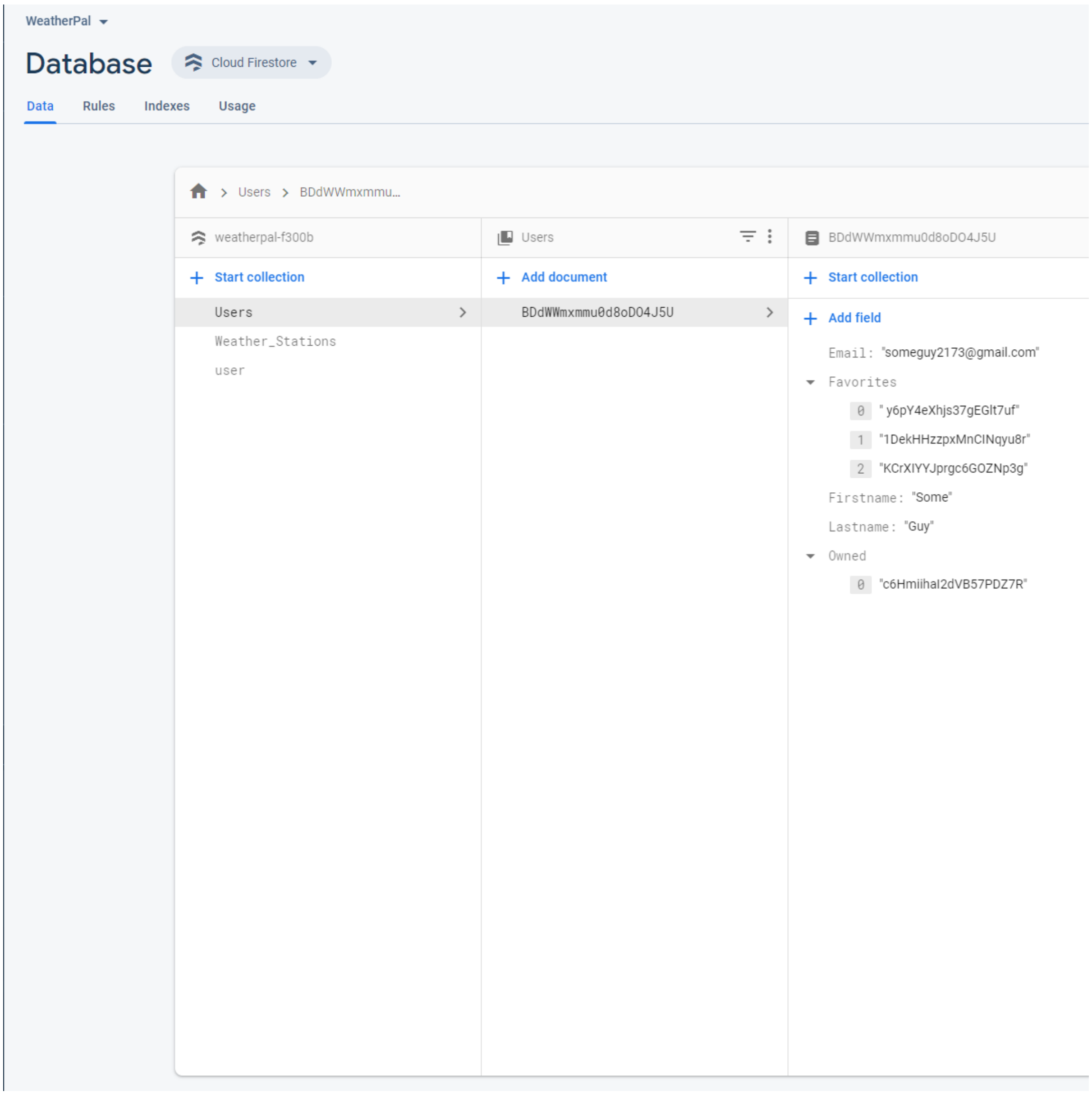


Class Diagram



Entity Relationship Diagram

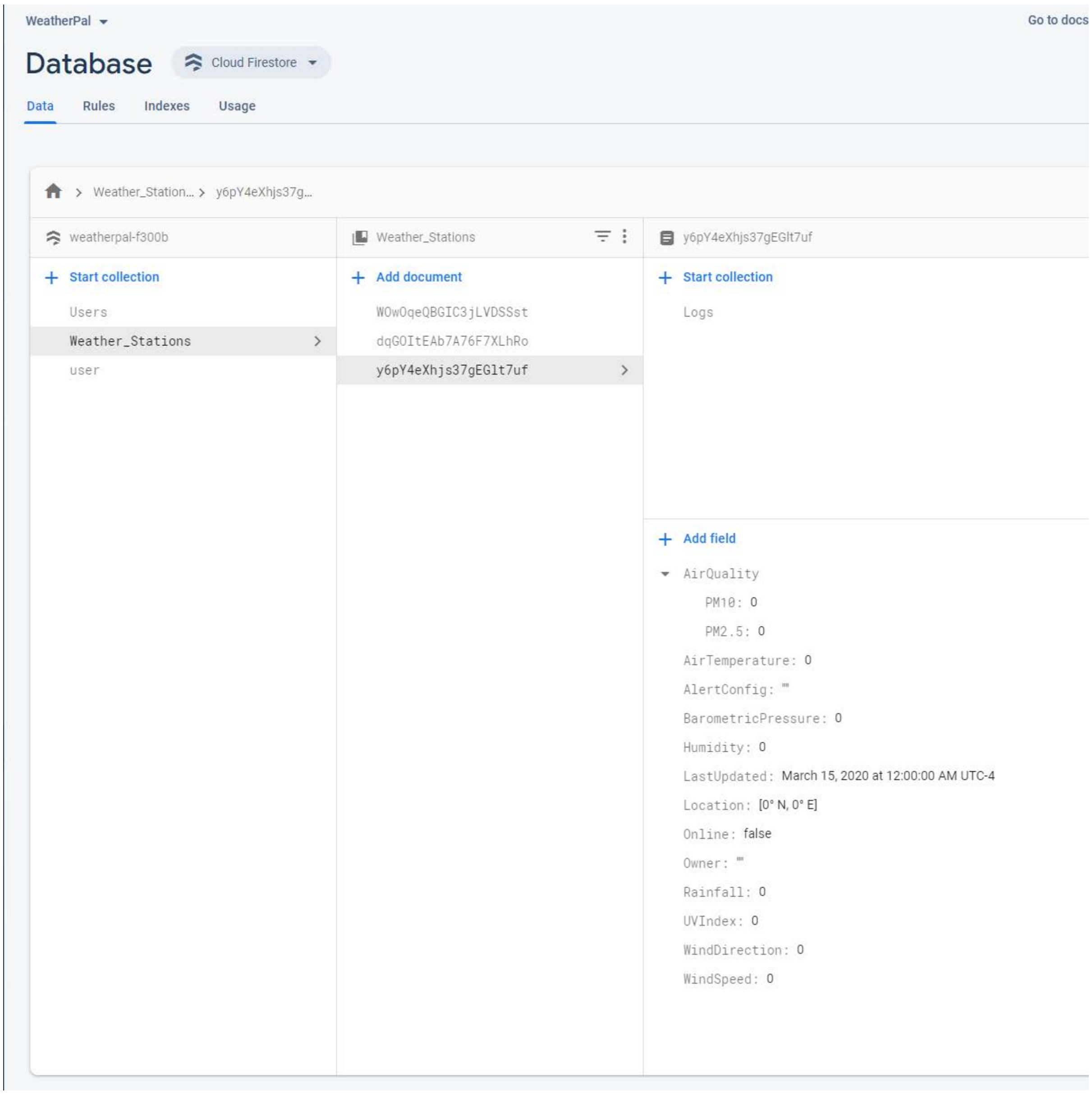
WeatherPal at the moment uses only Cloud Firestore database for storing its data. Since Cloud Firestore is a noSQL database with key value pairs, it cannot be represented with an ERD. However, screenshots of the current database structure is shown below.



```
Users                     BDdWWmxmmu0d8oDO4J5U
Weather_Stations
user
```

```
Email: "someguy2173@gmail.com"
Favorites
  0 " y6pY4eXhjs37gEGlt7uf"
  1 "1DekHHzzpxMnCINqyu8r"
  2 "KCrXIYYJprgc6GOZNp3g"
Firstname: "Some"
Lastname: "Guy"
Owned
  0 "c6Hmiihal2dVB57PDZ7R"
```

Figure 2.0: Showing fields currently used to store data on a WeatherPal user.



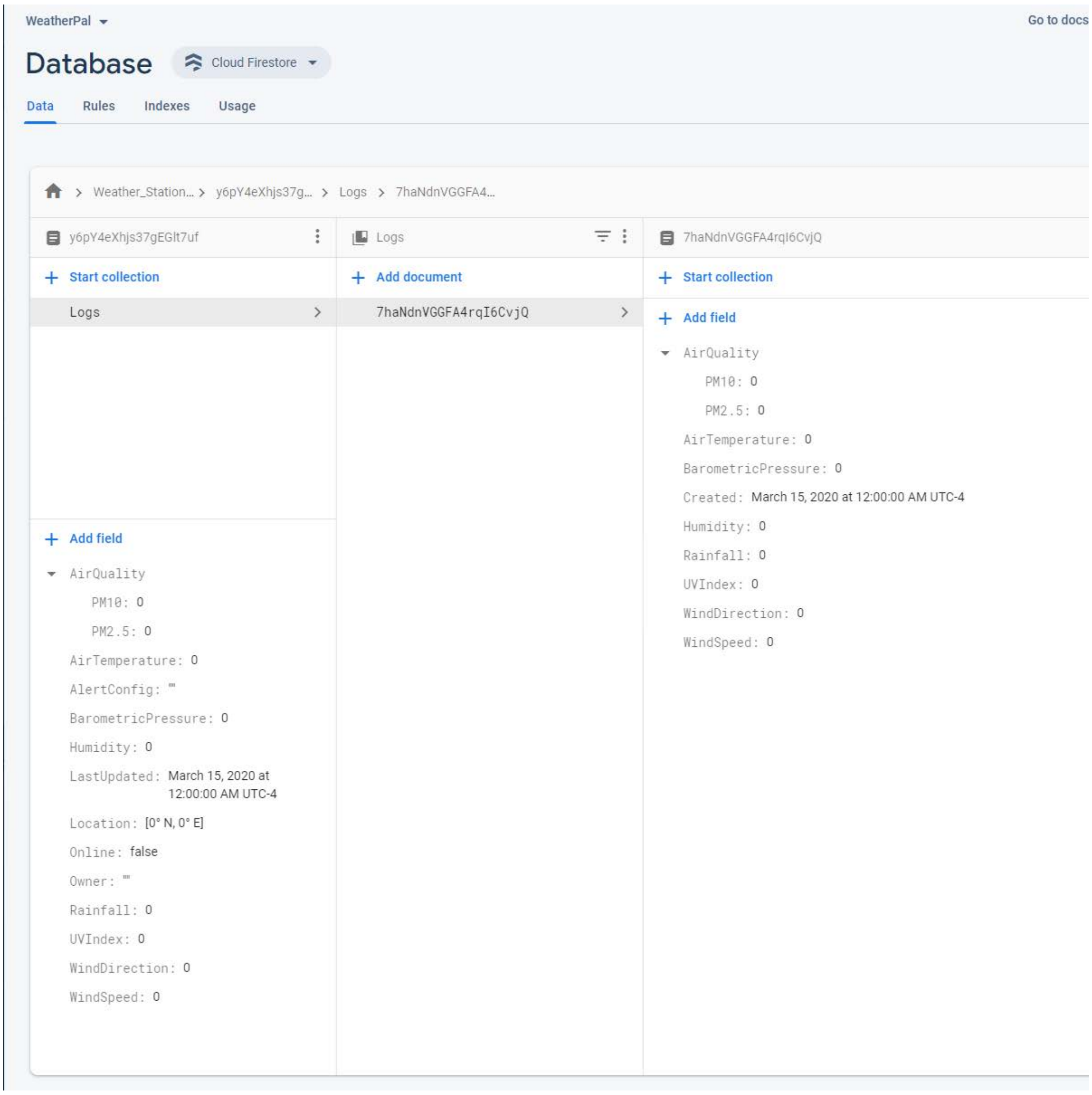
Users
Weather_Stations
user

WOwOqeQBGIC3jLVDSSst
dqGOItEAb7A76F7XLhRo
y6pY4eXhjs37gEGlt7uf

Logs

```
AirQuality
    PM10: 0
    PM2.5: 0
AirTemperature: 0
AlertConfig: ""
BarometricPressure: 0
Humidity: 0
LastUpdated: March 15, 2020 at 12:00:00 AM UTC-4
Location: [0° N, 0° E]
Online: false
Owner: ""
Rainfall: 0
UVIndex: 0
WindDirection: 0
WindSpeed: 0
```

Figure 2.1: Showing fields currently used to store data from a single weather station in a document within the “Weather _Stations” collection.



```
AirQuality
    PM10: 0
    PM2.5: 0
AirTemperature: 0
AlertConfig: ""
BarometricPressure: 0
Humidity: 0
LastUpdated: March 15, 2020 at 12:00:00 AM UTC-4
Location: [0° N, 0° E]
Online: false
Owner: ""
Rainfall: 0
UVIndex: 0
WindDirection: 0
WindSpeed: 0
```

```
AirQuality
    PM10: 0
    PM2.5: 0
AirTemperature: 0
BarometricPressure: 0
Created: March 15, 2020 at 12:00:00 AM UTC-4
Humidity: 0
Rainfall: 0
UVIndex: 0
WindDirection: 0
WindSpeed: 0
```

Figure 2.2: Showing the current contents of weather log documents within the “Logs” collection for weather stations.

Sequence Diagrams

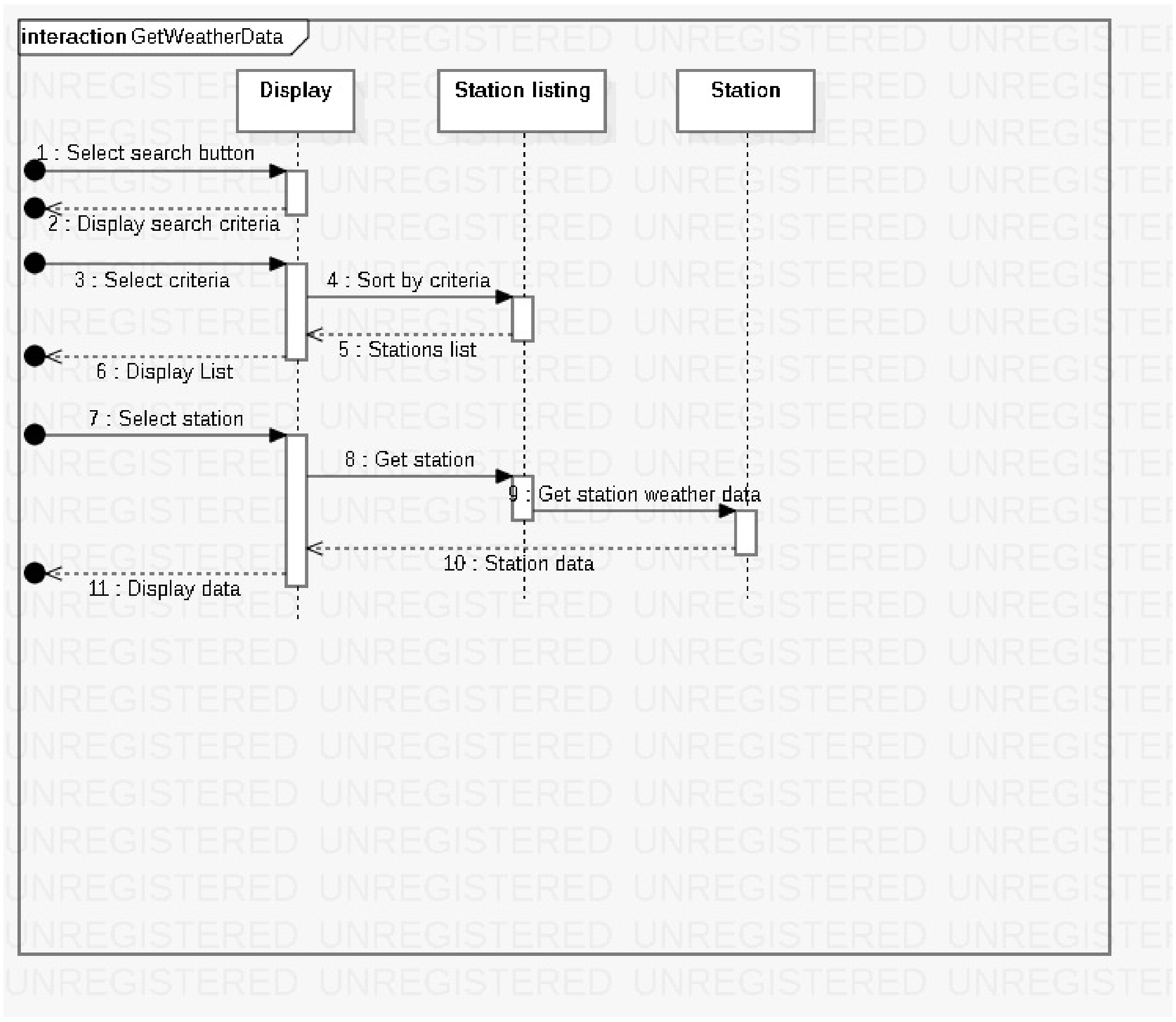


Figure 2.3: Showing the sequence involved in fetching weather data with WeatherPal

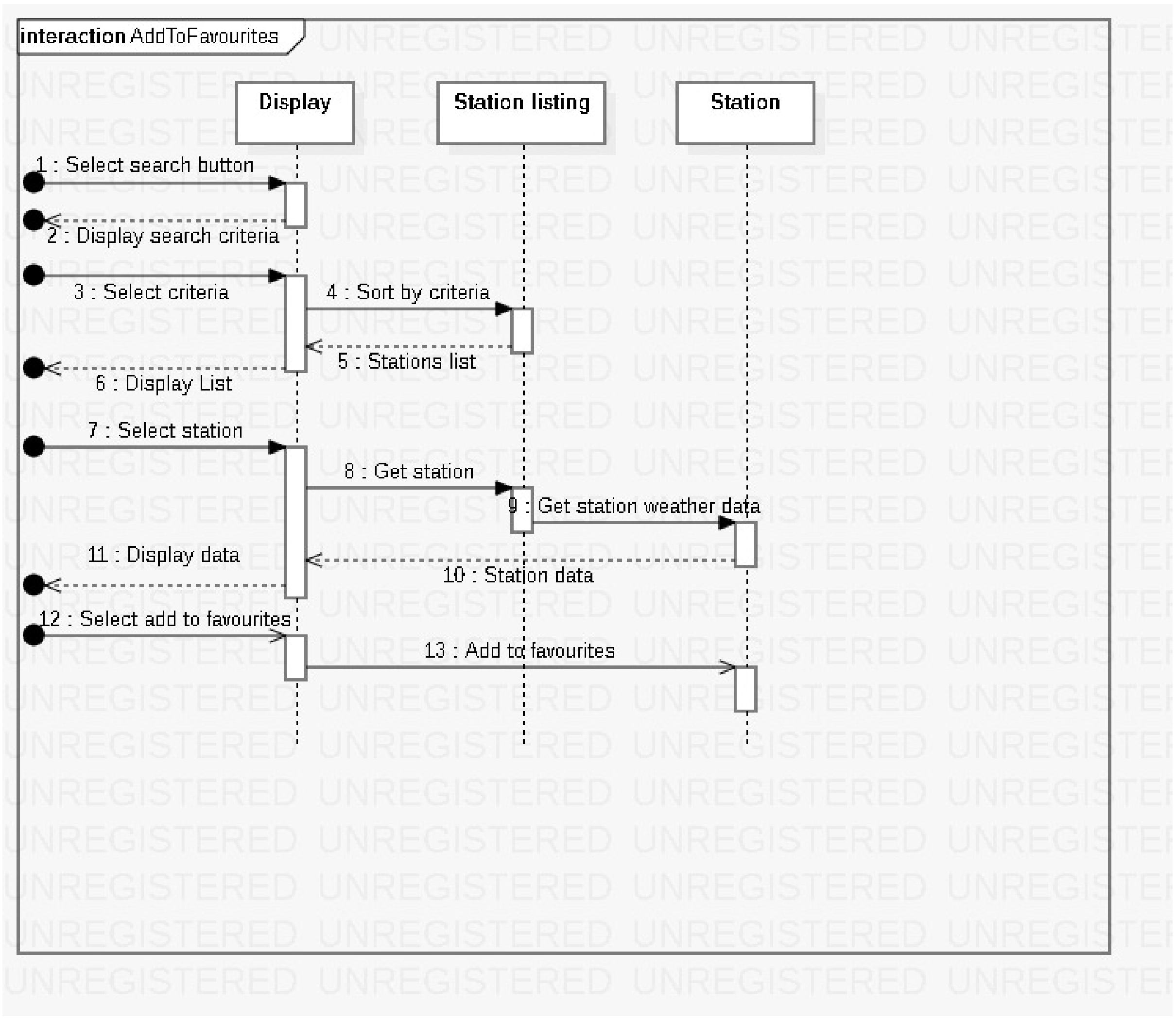


Figure 2.4: Showing the sequence involved in adding a weather station to the Favorites list with WeatherPal.

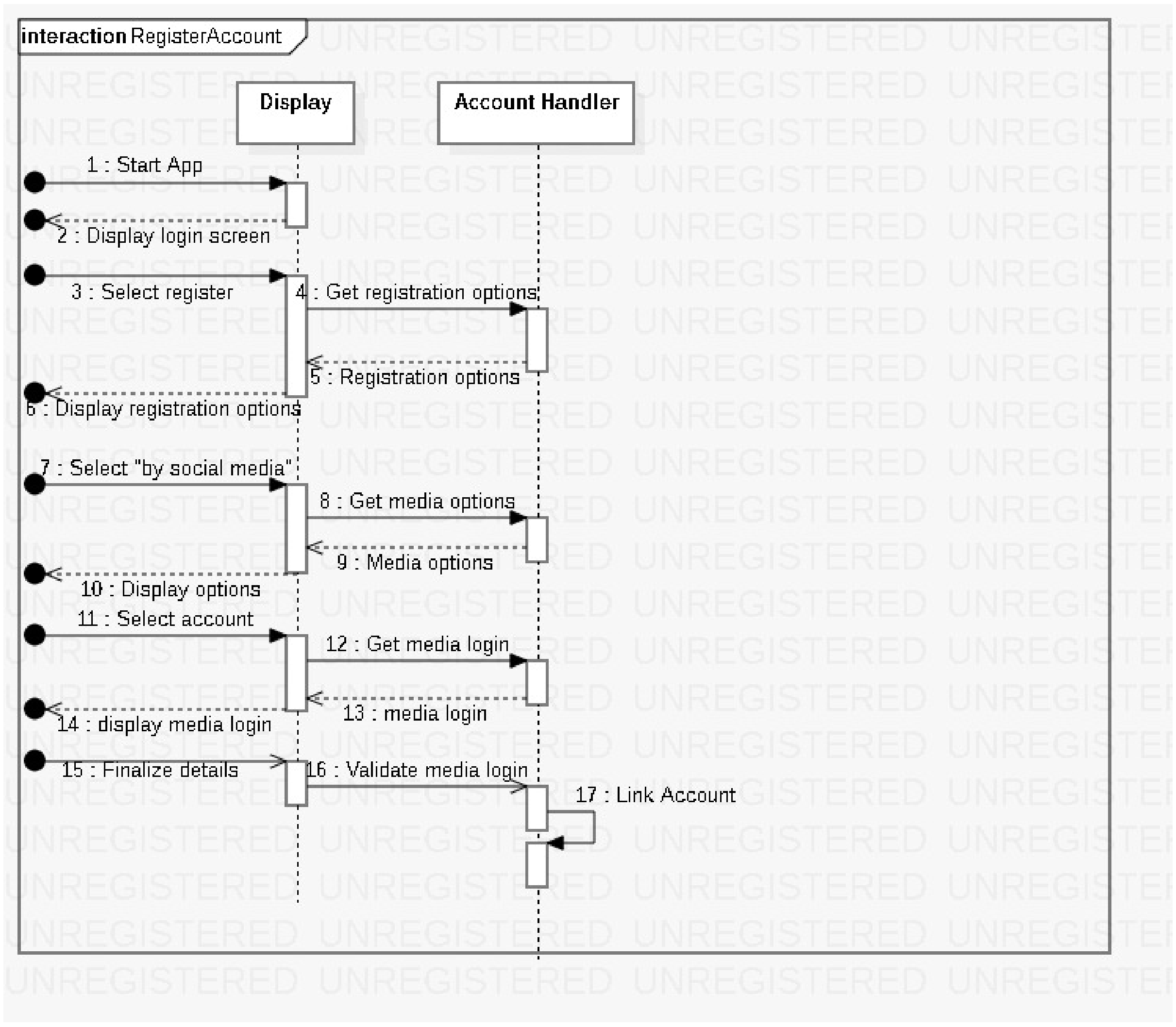


Figure 2.5: Showing the sequence involved in registering for an account with WeatherPal.