Sheffield Hallam University

College of Business, Technology and Engineering

Department of Computing
Project (Technical Computing)
[55-604708]
2020/21

Author:	**Ryan Diduca**
Student ID:	**27015161**
Year Submitted:	**2021**
Supervisor:	**Babak Khazaei**
Second Marker:	**Peter O'Neill**
Degree Course:	**BSc Computer Science**
Title of Project:	**Modular Weather Monitoring and Prediction Application Using IoT**

Confidentiality Required?

NO

I give permission to make my project report, video and deliverable accessible to staff and students on the Project (Technical Computing) module at Sheffield Hallam University.

YES

Acknowledgements
I would like to thank my supervisor Dr Babak Khazaei for his invaluable assistance throughout the planning and development of this project. I would also like to thank my family and friends for supporting me throughout my studies, as well as those who provided feedback in the survey.

Abstract
The ability to track weather and predict what it will do next is a hugely important one for many people, be it for gardeners needing to know whether there will be a morning frost that will kill off their plants or just for individuals wanting to know how warmly they should dress for the day. Nowadays, there is an abundance of sources for getting weather forecasts, but these are often provided over relatively large catchment regions, for example a whole town or a spread-out rural area. This can cause problems when the region is split by geographic features such as large hills that can create microclimates, resulting in quite different conditions on either side of the hill as a prediction will be made for the whole area and therefore can be inaccurate.

As a solution to this, taking your own readings and using them to predict conditions in small, localised areas would allow for greater accuracy. Taking a modular approach to this means that data can be collected from multiple locations that the user would choose and thus it can either be compared or averaged out over a customised area to enable the user to review data and make predictions about the precise area they would like. Additionally, utilizing IoT means that the collected data can be transmitted to a central location where it can be processed without the need for all the devices to be physically connected, increasing portability.

Contents

Abbreviations

ARIMA	AutoRegressive Integrated Moving Average
ARIMAX	AutoRegressive Integrated Moving Average with eXogenous factors
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IoT	Internet of Things
IP	Internet Protocol
MQTT	Message Queuing Telemetry Transport
OS	Operating System
PHP	PHP: Hypertext Preprocessor
Pi HATs	Pi Hardware Attached on Top
RDBMS	Relational Database Management System
SARIMA	Seasonal AutoRegressive Integrated Moving Average
SBD	Single Board Computer
SPI	Serial Peripheral Interface
SQL	Structured Query Language
SSH	Secure Shell Protocol
TSA	Time Series Analysis
UI	User Interface
Wi-Fi	Wireless Fidelity

Figures

Introduction

Aims & Objectives

The primary objective of the system is to make monitoring weather conditions in specific, user-defined locations as easy as possible by creating a system that allows the users to place data collection devices wherever they deem appropriate and using this information to show the previous conditions in each location and predict future conditions as well. To enable the users to view all of the data themselves, a web page will be created to display it in a useful, practical manner where the user can interact with the system. Finally, to evaluate the success of the project, a survey will be created to help assess how well a group of people think the project meets the above goals.

Approach

With the goal of creating an entirely new system from scratch in mind, the fact that several iterations of system development may well be required mandated that an agile methodology be employed. While a waterfall approach could have been taken, the fact that it has a rigid structure meant that it would not allow for any changes in design no matter how essential, therefore an agile approach was necessitated.

Agile Methodology

Cprime Inc (2021) describes agile as a group of software development methodologies based on iterative development. It is an approach that allows for a review after each round of the development process so that new plans can be made, designed and developed in the next sprint phase, allowing for much greater flexibility when creating a system.

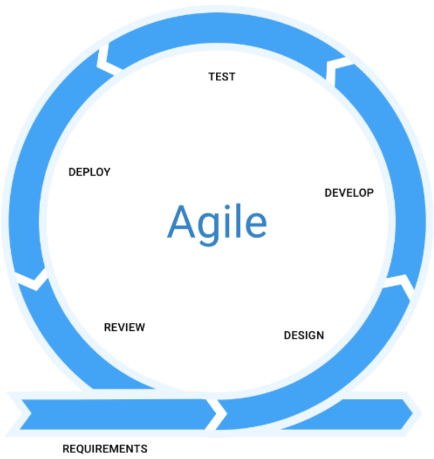


Figure 1: The Agile development methodology

This approach is ideal for the project as it will require a substantial development phase where new issues are likely to appear as progress is made, which agile allows for. For the purposes of this system, the software developer will act as the primary stakeholder to review each iteration and determine the changes and improvements that needed to be made in the following sprint phase.

Investigation

IoT

The internet of things (IoT) is defined by Rouse (2020) as "a system of interrelated computing devices, mechanical and digital machines that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction". Using this definition, it is clear to see that a major advantage of utilising IoT is that it enables processes to be automated so that no human input is necessary to the successful running of an application. It also shows that it is well suited for use in a project such as this, whereby there are multiple devices running in different locations as it allows for communication between these devices as long as they are connected to a network.

Weather Patterns

For the weather prediction aspect of the project, several different approaches were investigated. A standard weather forecast by the Met Office uses around half a million data readings of temperature, pressure, wind speed and direction, humidity and many other factors to form the basis of the forecast model (Met Office, 2009). From there, numerical weather prediction algorithms are run using a supercomputer and satellite images & rainfall radar observations are analysed to produce an accurate forecast. Since this project will be on a much smaller scale, considering fewer variables from data collected across less locations, a different approach will be required.

Another possible approach would be to utilise machine learning to predict weather. This would involve obtaining a mathematical model that fits the data inputted and then, once this model represents accurately known weather, it would perform the prediction again using new data (Cifuentes et al, 2020). In theory by doing this, the predictions get more accurate over time as more simulations are run. This is a better option as it can work with less variables, however it still requires a good deal of processing power to run successfully, which is not ideal for the hardware used in this project.

Ultimately, it was decided that a machine learning approach would be taken, using a time series analysis (TSA) model to predict the future conditions. Three of the most popular of these models are ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal AutoRegressive Integrated Moving Average) and ARIMAX (AutoRegressive Integrated Moving Average with eXogenous factors). Each of these models produces a set of new values based on the weighted sum of the values input into the model while also considering past residual values to account for time series decomposition. In the case of SARIMA, the ability to weigh the values differently based on seasons is introduced. While this in theory would be more accurate than a standard ARIMA in the long term, it would be far more inaccurate until the first time period had passed (one year in this case) as not enough data would have been collected for the model to be correctly influenced by the seasons. Lastly, ARIMAX is similar to SARIMA in that it can account for extra variables to skew the produced values, however in this case the variable can be any other variable that might have an effect on the weather. This would not necessarily be needed as the system only takes in a set number of variables but could be useful if the project were to be extended in scope. Therefore, the ARIMA model will be used as it will provide the most accurate results over the timescale of this project, however if it were to be extended then the SARIMA model could be swapped in for it at a later point.

Hardware

In order to meet the constraints of the IoT definition mentioned previously, a couple of single-board computers or microcontrollers will be required to enable the data collection, transfer and storage. For the purposes of this project there be two different roles for the computers to fulfil, firstly the data collection devices of the system which will collect weather data at a location and then send it off to the second type, a central “server” device. Also required for the project would be sensor components capable of taking weather data readings in conjunction with the data collection devices.

Raspberry Pi 4 Model B

There are several requirements for the main “server” device in this project, the first of which is the ability to run multiple tasks at once. This is because the “server” will be running several programs simultaneously, including one that listens for incoming data from the collection devices as well as hosting the web page for displaying the data. Another requirement is that it will need to be able to connect to the internet as this is how the data transfer aspect of the project will be handled. Additionally, a good amount of storage capacity will be required to store the weather data as this will only grow over time, therefore the ability to upgrade the storage capacity in future would be optimal.

Microcontrollers have many advantages, for example low cost, low performance overhead due to a lack of an operating system and the fact that they can run tasks immediately upon booting. Despite this, when discussing one of the best-known microcontrollers (the Arduino UNO), Pounder (2020) states that they are only capable of running one program at a time. Also, an Arduino cannot natively connect to the internet on its own and instead requires the use of an external “Shield” to do so (Pedamkar, 2021). Therefore, according to the above requirements, microcontrollers are not suitable for this project and an alternative will be required.

Looking further into the topic revealed single-board computers to be an obvious choice to use instead of microcontrollers as they can perform almost all of the tasks of a regular desktop computer in a small, self-contained space. According to All3DP (2021), the Raspberry Pi is the “Undisputed heavyweight champion of the world” when it comes to single board computers, however it does go on to list a range of other devices that would both match the above criteria and be comparable to the Pi, each with their own pros and cons. Some of the others listed include the LattePanda V1, the Rock64 Media Board and the Pine A64-LTS, however after considering the features of each, it was decided that the Raspberry Pi in its latest release (Pi 4 Model B) would be used for this project.

Figure 2: Raspberry Pi 4 Model B

One of the main reasons the Pi was chosen over the others in this situation was that the Pi offers everything that is required for a reasonable price and since it is very widely used, with sales passing 27 million units (Heath, 2019), there is a large amount of support available online, both official and community made. This means that any issues with development can likely be easily remedied by a quick internet search. In addition to this, the Pi has a few different options when it comes to the amount of RAM it has; 2gb, 4gb and 8gb being the versions currently available. For the purposes of this project, the 4gb version will be used. While this is more than the project will likely require and a less powerful version could be used, it will enable the project to run smoothly and allows for future scalability of the system if necessary.

Raspberry Pi Zero WH

The data collection devices mentioned earlier have slightly different requirements from the "server" as they will need to be able to interact with a sensor component that will take weather data readings, but they will not need to have as much processing power as they will have less tasks to do within the system. They will however still need to be able to connect to the internet to transfer the data.

The requirements in this case mean that microcontrollers could be considered again as only one task would need to be running at a time, however ultimately it was decided that the convenience of a smaller SBD with built in Wi-Fi and the ability to code directly on the device itself would make it the better choice in this situation.

A device that kept cropping up in research that fit the description was the Raspberry Pi Zero WH, with Sims (2017) describing it as "versatile and flexible". This SBD has the advantage of being able to connect to Wi-Fi on its own and it has access to the full range of Raspberry Pi supported GPIO headers, known as Pi HATs ("Hardware Attached On Top") (Maker.IO, 2021). There are several Pi HATs that can take weather readings, however the Zero is also capable of connecting to other sensors via its GPIO pins, meaning that there are plenty of options for weather components available. Additionally, since it is simply a less powerful, smaller form-factor version of the Raspberry Pi 4 Model B, any software used on one device will work on the other, making setup and development easier due to the consistency between the devices.

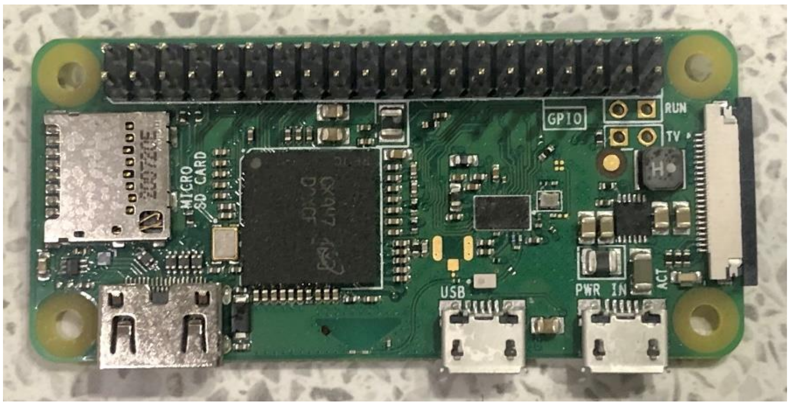


Figure 3: Raspberry Pi Zero WH

A combination of the above reasons as well as the low price point, £9.30 at the time of writing on Pimoroni (2021), is why the Pi Zero WH will be used for this project. Two of these devices will be required in order to demonstrate the modular aspect of the system and show working in multiple locations.

Enviro for Raspberry Pi

The weather data readings that will be taken in this system will be the temperature, humidity and pressure, therefore either a single sensor that can measure all three of these at once or a series of multiple sensors working in tandem will be required. Convenience dictates that the latter of these would be ideal as it would mean not having to deal with the inconsistencies in software between all three sensors.

Since Raspberry Pi Zeros are being used as the data collection devices in this system, Pi HATs can be utilized as previously mentioned, with there being a handful of options that fit the requirements such as the “Sense HAT”, the “Enviro for Raspberry Pi” and the “Adafruit SCD-30”. Of these options the Enviro appeared to be the best option as it can take all three of the measurements whilst also coming with a pre-existing python library that enables the easy use of the device (King, 2019), therefore, for the purposes of this project the Enviro for Raspberry Pi will be used.

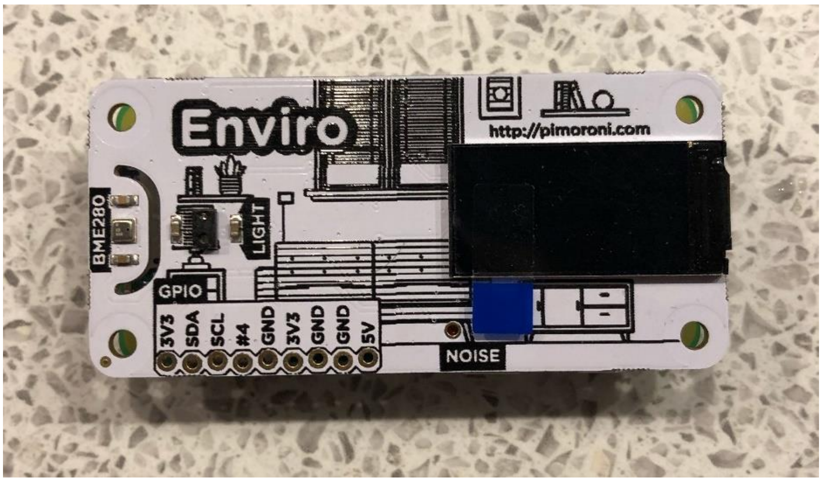


Figure 4: Enviro for Raspberry Pi

Back-end

The web page user interface aspect of the project will follow the client-server model whereby the client will make a request to the server, while the server manages the requests and provides the information to the client.

Python

According to the Python Software Foundation (2021) “Python is an interpreted, object-oriented, high-level programming language with dynamic semantics”. In this project, python will be used on the devices for collecting and transmitting data, this is because the library for the Enviro weather sensor is written in Python. In addition to this, Python is the language of choice for development on a Raspberry Pi due to its versatility and simple syntax making it easier to edit in text editors.

PHP

PHP is an open-source general-purpose scripting language that is especially suited for web development and can be embedded into HTML (The PHP Group, 2021). For the purposes of this project, PHP will be used to retrieve information from the database so that it can be displayed on the web portal.

Cron

Cron is a time-based job-scheduler in Unix based operating systems (ArchLinux, 2021). It is a utility that allows a task or script to be run automatically at pre-set times and dates, making it very useful for this project in which the data gathering scripts on the collection devices need to be run at regular, repetitive intervals. This means that once a “cron job” has been set up to handle running the data collection scripts, no human input is required to make the system run, thus making it fully automated.

```
# ┌───────────── min (0 - 59)
# │ ┌────────────── hour (0 - 23)
# │ │ ┌─────────────── day of month (1 - 31)
# │ │ │ ┌──────────────── month (1 - 12)
# │ │ │ │ ┌───────────────── day of week (0 - 6) (0 to 6 are Sunday to
# │ │ │ │ │                  Saturday, or use names; 7 is also Sunday)
# │ │ │ │ │
# │ │ │ │ │
# * * * * *  command to execute
```

Figure 5: The structure of the Cron job scheduler

MariaDB

MariaDB is an open-source SQL RDBMS that is very similar to MySQL, however MariaDB offers increased performance in most scenarios (Jankov, 2020). This will be the main data store for the project on the server where all weather information will be placed once it has been transmitted from the collection devices.

Front-end

The web page UI will be the one location in the project where a user can interact with the system, allowing them to monitor the data stored within the database. The page should be a simple design that does not contain information which is irrelevant or rarely needed as every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility according to the 8th of Nielsen's 10 usability heuristics for user interface design (Nielsen, 2020). In addition to simplicity, the page should be responsive. Responsive web design responds to the needs of the users and the devices that they are using, the layout changes based on the size and capabilities of the device (LePage & Andrew, 2020). This definition shows that the portal should be resizable to account for users accessing it on different devices in order to improve cross-platform support.

The main functionality required of the web portal is to graphically display the historical and future weather data stored in the system as well as to allow the user to select the time period over which they want to view this data.

HTML, CSS & JavaScript

HTML, CSS and JavaScript are the three languages that form the building blocks required to make a simple, functional web page, each of the three have their own unique purpose in the structure, design and functionality of the web page. Using pure CSS as opposed to a framework such as Bootstrap allows for a more lightweight web page that is tailored to the needs of the project, rather than using a framework that causes bloat and can result in slower loading times.

Chart.js

In order to fulfil the requirement of the web page that shows the stored temperature, pressure and humidity data graphically, a tool will be needed that can convert the data points stored in the database into graphs. According to Singhal (2020), Chart.js is perfect for small systems, providing flat, clean, elegant JavaScript charts, fast. Since it appears to have all the functionality required by the project whilst also being very minimal at just 60kb (Puszynski, 2019), this will be an ideal library to use in the web page.

Design

This section will discuss the design of the system, taking into consideration the objectives stated for this project and the requirements of potential users.

Software

For this project, there are essentially two halves to consider; the data collection and processing back-end as well as the front-end web page where the user interacts with the system. For the former, the devices chosen in the research section of the document would be used to collect the data via sensor components and then send it back to the central server device. To control the logic flow of this a few Python scripts would be needed, these would be developed using the “IDLE” Python development environment and then copied across to the devices where the “Nano” text editor provided by the operating system would be used to make any small changes during testing.

The front-end would require a similar approach, with the main section of development being done in the “Sublime Text 3” development environment before transferring the files across to the central server device used to host the web page where any changes would again be undertaken in the “Nano” text editor.

In addition to the software used for the development of the code, GitHub was used as a version control system and a way of easily moving files between devices. This was exceptionally helpful to the project as the need to develop on multiple devices and move them across to where they needed to be was made a lot simpler.

Lastly, Trello was employed for the system as a Kanban board to track progress and set goals in the development of the system. Thanks to this, it was straightforward to update the to-do list as development progressed and work out what needed to be done next.

Architectural Model

System Context

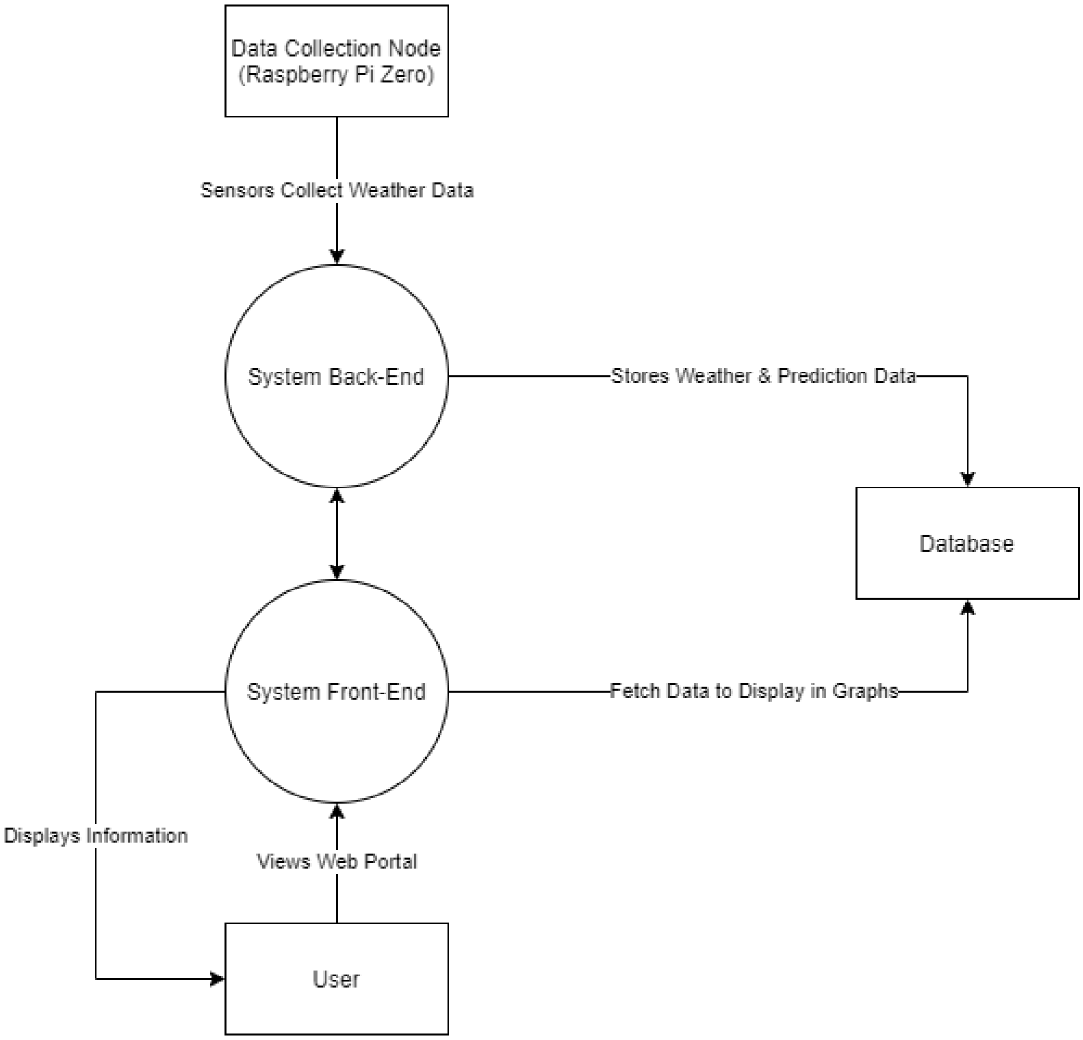


Figure 6: The system boundary diagram

Architecture Overview

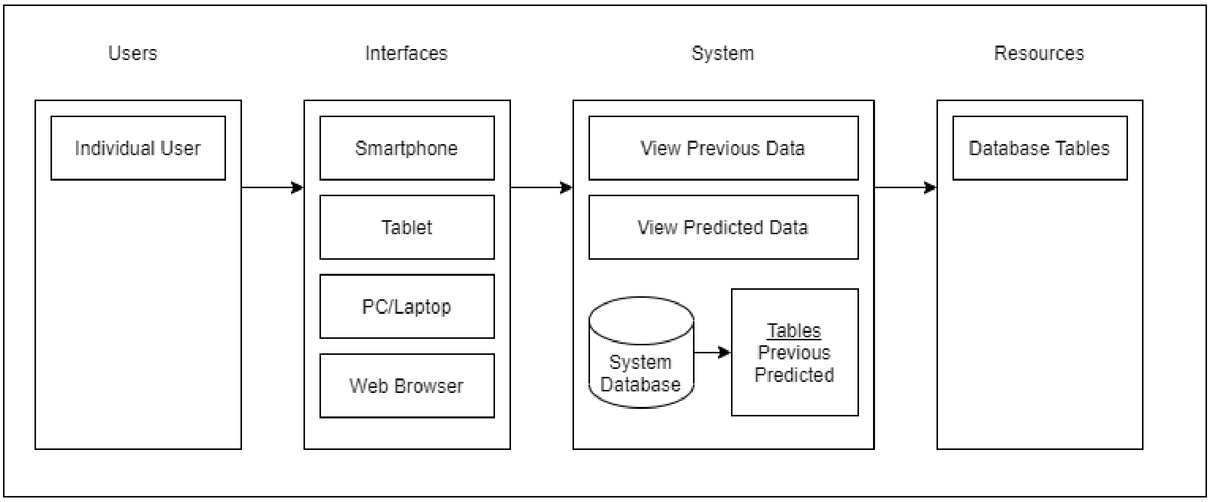


Figure 7: The architectural overview showing the major aspects of the system and how they interact

Data Collection

As discussed in the research section of this report, the Raspberry Pi Zero WH was chosen as the device to collect weather data and transmit it across to the central server. It would do this in combination with the Enviro for Raspberry Pi sensor component, which would take the data readings itself when commanded to by the Pi Zero. Connecting the component to the device would not be an issue as it was designed to sit on the header pins of the Pi and from there, as long as I2C and SPI communication was enabled on the Pi, it would be ready for use.

Figure 8: The Pi Zero and Enviro connection

In order to get the most accurate data possible, the data would need to be collected regularly at each location. To achieve this, a Python script would need to be created to handle the collection process which would then be set to regularly repeat by using the Cron job scheduler.

Data Transmission & Storage

Data transmission was one of the key aspects of this system, the collection devices needed to be able to transmit data to a central server from wherever they were located, therefore communication would need to be done wirelessly. There were a couple of options to handle this, for example using HTTP (Hypertext Transfer Protocol) or MQTT (Message Queuing Telemetry Transport) to send data across the internet via a message broker, ultimately MQTT was chosen due to it being more suited to data transfer as opposed to HTTP, which is more appropriate for document transfer. Additionally, the throughput of MQTT is 93 times faster than HTTP (Serozhenko, 2018), making it much more responsive. Python also has a good deal of libraries available to handle communication via MQTT, such as “paho-mqtt”, “HBMQTT” and “gmqtt” so using one of these will enable easy communication between the collection devices and the server device.

In terms of data storage, using a database was determined to be the best approach as opposed to saving values to a CSV file as it was far more scalable, allowing for different tables to be added to store separate information, such as previously recorded weather data, predicted weather data and data averaged over certain time periods. The chosen method for implementing this, as discussed in the research section, was to install MariaDB on the central server device and create tables as necessary.

User Interface

To meet one of the objectives of the project, the user needed to be able to interact with the system in a meaningful way that allowed them to view the stored and predicted data. Therefore, a web page was decided upon as the best way of displaying this information as it meant that the user could view it from any internet connected device at any location, making it more portable than a desktop application for example. In order to host this web page, the same central server used to save and process collected weather data would be needed to store the web pages and then host them via PHP's built in web server by running the following command.

```
php -S 192.168.137.23:8000
```

Figure 9: PHP command to start a web server

The decision was made to host the web page on a local network only instead of on the internet, as purchasing a domain was unnecessary to show the functionality of the system. This could be integrated at a later point if necessary, but it would work in a very similar way with the central server device hosting the web page.

```
19:06:00 2021] 192.168.137.1:59695 [200]: /
19:06:00 2021] 192.168.137.1:59698 [200]: /css/WeatherStyles.css
19:06:00 2021] 192.168.137.1:59696 [200]: /js/graphs.php
```

Figure 10: The server log when the web page is accessed

Security Consideration

As explained within the data transmission section, MQTT is the chosen method of sending information over the internet for this system. Any data that is transmitted across the internet is in danger of being intercepted by others, however the risk in this case is fairly minimal as the data being transmitted contains no personal or sensitive information. Having opted to use a public MQTT broker, "test.mosquitto.org", on which anyone could listen to and post on any topic, it was decided to use a fairly obscure topic that would have low to zero external traffic, in this case "aloadofrandomweatherdata" was chosen. Additionally, when reading data from the topic it would only be saved if it fitted the correct format of message used in the system, so as to reduce the risk of anyone inserting random values externally.

Development

In this section, the development process of the system will be described, including the technologies, libraries and tools utilised in the implementation. Python was the primary language used for the development of the back-end part of the system, whilst a combination of HTML, JavaScript, CSS and PHP were used to create the web portal section. Some SQL was also required to interact with the MariaDB database.

Setting Up Development Environment

The first decision that needed to be made in the development phase was to choose which operating system the Raspberry Pis would run on. While Raspberry Pi OS (formerly known as Raspbian) is the official supported operating system provided by the Raspberry Pi Foundation, it can require a substantial amount of processing power to run, especially in comparison with the capabilities of the Pi Zero's processor. Also, it comprises of multiple features that were not needed for this project, such as a GUI. Therefore "DietPi", an extremely lightweight Debian OS, was chosen instead as it provided all the features necessary for the system whilst minimising bloatware and its effect on resource usage.

With the operating system having been chosen, the next stage was to choose how to control each of the Pis as there were two options available: connect peripherals (monitor and keyboard) to each of the devices or use remote access software. Using the latter meant all development could be done centrally from one computer and it also avoided the need to spend money on unnecessary peripherals, so it was the far more convenient choice. To achieve this, "PuTTY" was installed on a central computer which was used to connect to each of the Pis over a network via the Secure Shell Protocol (SSH).

Unfortunately, the Wi-Fi connection provided to the Pis was on a shared network which does not allow the use of SSH, so a personal hotspot had to be set up on the central computer which the Pis could connect to. This did allow the Pis to get access to a network, however it caused some slight inconvenience as static IP addresses could not be set up, so the IP address for each Pi changed quite regularly.

Since "DietPi" was being used as an operating system for each of the Raspberry Pis, it was decided that its inbuilt text editor "Nano" would be used to write and edit the code being run on the devices as it was simple to use and was compatible with each of the languages used. In addition to this, some coding was done on the central computer with "Sublime Text 3" being used for part of the web-based section of the project and "IDLE" being used for some Python development before the code was then copied across to the individual devices as these more professional development environments were more convenient for debugging purposes than the "Nano" text editor.

3rd Party Libraries

Over the course of the project several external libraries were used to aid development, the majority of which were used within Python. Python has a huge array of libraries available to it for all manner of functions, so finding ones that fit the purpose of the project was easy. Firstly, to handle communication via MQTT, it was decided that “Paho MQTT” would be ideal as it was capable of reading and writing to MQTT topics and it had a good amount of documentation available on how it worked, meaning that its implementation was simple. The other aspect of communication to be handled was how the Python script would read and write to the MariaDB MySQL database, for which the “MariaDB” library was used as it was created by the MariaDB Corporation themselves for this exact purpose, so it was therefore the best suited to the task.

For the implementation of the prediction algorithm, a combination the “NumPy” and “statsmodels.tsa.arima.model” libraries were utilised. These were used to correctly format the starting data set and then provide the ARIMA time series analysis mathematical model that was chosen in the research section of this report.

To connect the Raspberry Pi Zeros to the Enviro sensor and allow measurements to be taken, a combination of the “SMBus” and “BME280” libraries were required. “SMBus” allows the Pi to connect to Enviro via I2C (“Inter-Integrated-Circuit Bus”) and send and receive data, whereas “BME280” is a library supplied by “Pimoroni”, the makers of the Enviro sensor, which provides the functions necessary to request data from the sensor and return it correctly formatted. Additionally, “datetime” and “logging” were two standard modules used in development for taking timestamps and printing out values for debugging purposes, respectively.

Lastly, the JavaScript code for the web page made use of one external library called “Chart.js” which enables the easy creation of charts and graphs using the HTML Canvas element. This was very useful for the system as it meant that the creation, updating and formatting of graphs was far simpler than the manual alternative.

Setting Up Data Collection

The first major aspect of the system to be developed was the collection and storage of data using Python on the Raspberry Pi Zero devices. This was because doing this part first meant the system would have more time to collect data from the nodes which would allow the prediction algorithm, which was to be implemented later, to have a larger dataset to work with and hopefully cause it to provide more accurate results.

Setting Up Collection Devices

Setting up the data collection devices, two Raspberry Pi Zeros with Enviro sensors, was a straightforward process. Once the previously chosen “DietPi” OS had been installed on both devices, the next step was to enable I2C & SPI and connect the Enviro sensor to the headers of the Pis so that the Pis could communicate with them. From there, it was a case of installing external libraries onto each device, in this case the “Paho MQTT” and “BME280” libraries, via the “pip install” command.

Adding Code to Take Reading from Collection Devices

On each of the nodes, a Python script called “weather.py” was created for the purpose of taking a measurement for the current temperature, humidity and pressure. The “BME280” library mentioned previously provided several functions used in this script, namely the “get_temperature()”, “get_pressure()” and “get_humidity()” functions. To get an accurate measurement off the sensor, it appeared that a couple of readings needed to be taken sequentially because the sudden change in voltage provided to the sensor after having been dormant for a while seemed to skew the results. Therefore, a first reading was taken and discarded before 3 further sets of readings of each of the variables were taken a second apart and then averaged to provide a more accurate reading as shown below.

```
for i in range(3):
    temperature += bme280.get_temperature()
    pressure += bme280.get_pressure()
    humidity += bme280.get_humidity()
    time.sleep(1)

temperature = temperature / 3
pressure = pressure / 3
humidity = humidity / 3
```

Figure 11: Python code to retrieve average readings

Adding Cron Job to Automatically Take Regular Readings

Once the "weather.py" script had been created, the next stage was to use Cron to schedule the script to run at a regular time interval so that continuous readings could be taken. Ultimately, every five minutes was chosen as the interval for the readings to be taken and so "crontab -e" was called in the terminal of each collection device and the following was added.

```
 GNU nano 3.2

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/5 * * * * python3 /root/docs/weather.py
```

Figure 12: Crontab showing the weather.py script running every 5 minutes

Data Transmission & Storage

The next logical step in development was to add a way for the data being gathered by the collection devices to be sent to the central server device and stored in a database. For this purpose, a Python script called "listener.py" was created on the server device and then a Cron job was added to make the script run automatically whenever the device started up. The script would then always be running in the background so it could always be ready to receive data from the nodes.

This script would take an incoming message from the collection devices and separate it appropriately into variables matching its constituent parts, making use of the Python ".split()" function to do so.

```python
def on_message_received(client, userdata, message):
    breakdown = str(message.payload).split(",")
    loc = int(breakdown[1])
    temp = float(breakdown[2])
    hum = float(breakdown[3])
    pres = float(breakdown[4])
```

Figure 13: Part of the Python function to breakdown a message received via MQTT

Implementing MQTT with Paho-MQTT

For the “listener.py” script to receive data from the collection nodes via the MQTT protocol, the “Paho MQTT” library was added to both the “listener.py” script as well as the “weather.py” scripts on the collection devices so that they could publish the data to a chosen MQTT topic. The topic ultimately chosen for this was called “aloadofrandomweatherdata” as it was deemed obscure enough that there would not be much, if any, external traffic using it. The MQTT server used was “test.mosquitto.org” as it was a free server that allowed communication over the internet rather than just a local network, which made it ideal for this project where the collection devices would be located in different places across different networks.

The “listener.py” script was also setup so that whenever a message was received, a function would be called to do the breakdown process mentioned previously. It would also continually listen for these messages on an infinite loop.

Creating Database

As stated in the research section of this report, MariaDB was the chosen to be the main data store for the project. It was therefore installed and setup on the central server device with one initial table created named “weather”, the SQL code for which is as follows:

“CREATE TABLE weather (ID INT PRIMARY KEY AUTO_INCREMENT, location INT, temperature FLOAT, humidity FLOAT, pressure FLOAT, date DATE, time TIME);”

Once the database had been created, the next stage was to connect it to the “listener.py” script so that data could be permanently stored. This was accomplished using the “MariaDB” Python library which allowed the script to form a connection to the database and then use SQL commands to read and write to the database so that the newest data could be inserted and saved.

Creating User Interface

Once the data had been collected, formatted and saved, the next stage in the development process was to create the web page where the user could interact with the system and view all the stored data. The page was created using a combination of HTML, CSS and JavaScript to handle the structure, appearance and functionality of the page, respectively. In addition, PHP was required to facilitate the connection between the database and the web page. A couple of dropdown buttons were added to allow the user to choose what they wanted to be displayed on the screen, this could be either predicted weather over the course of the next week or previously collected data shown over a period of 24 hours, 7 days or 12 months. To complement this, a table was also added that displays the most recent few data readings as they come in.

```
<?php
$hostname = [redacted]
$username = [redacted]
$password = [redacted]
$db = "weather";
$dbconnect = new mysqli($hostname,$username,$password,$db);

if ($dbconnect->connect_error) {
        die("Database connection failed: " . $dbconnect->connect_error);
}
```

Figure 14: The PHP connection to the database

PHP allowed for a connection to be made to the MariaDB server and for data to be read back via the use of SQL queries. The results of these queries were formatted and then stored in JavaScript variables so that they could be used as data points in the graphs which would be created using the “Chart.js” library mentioned previously.

Implementing Weather Analysis & Prediction

The last major aspect of the development process to be tackled was firstly processing the stored data before formatting it into useful time periods and saving it back to the database and finally analysing the stored data and running it through the chosen mathematical model to predict future weather patterns.

Analysing & Formatting Collected Data

As the web page gives the user the option to view previously stored data as an average value for each hour, day or month, a method of creating those average values needed to be established. This is where the “sortPastData.py” Python script was added to the central server which reads all values stored in the “weather” database table, averages them over each of the three time periods and then saves them into three new database tables called “hourly”, “daily” and “monthly”. The purpose of this was to make retrieving data on the web page easier as the averages would already be worked out and could simply be saved into variables and displayed as graphs on that end rather than trying to average the values within PHP itself. The script was set to run once every hour via Cron so that it could update and account for the latest incoming data.

Predicting Future Weather

To implement the prediction of future weather, another Python script called “predictor.py” was created on the central server. To begin with, the script would loop through the stored temperature data in the “weather” database table for each location and use this to find the average temperature for each location. It would then make use of the previously mentioned “NumPy” and “statsmodels.tsa.arima.model” libraries to feed this data into an ARIMA mathematical model, which would use it to produce a series of 7 values for each location as predictions for the temperature at each location over the course of the following week.

```
model = ARIMA(seriesToPredict, order=(7,0,1))
modelFit = model.fit()
startIndex = len(seriesToPredict)
endIndex = startIndex + 6
prediction = modelFit.predict(start=startIndex, end=endIndex)
```

Figure 15: Part of the Python code showing the mathematical model

As discussed previously, using a SARIMA model would likely have been better in the long run to produce more accurate data, however due to the time constraints of this project ARIMA was the most likely to produce correct results with the smaller dataset. If the project were to be extended over a longer period, the SARIMA model could be swapped in without too much trouble.

Lastly, weekly predictions were stored in a new database table called “predictions”, so that they could be retrieved on the web page in the same manner as the other stored data. This prediction script was set to run once a day via Cron and update the forecast for the week ahead as it would then have an extra day to work with each time to improve the accuracy of the prediction.

Testing

The testing process was again split into two halves to analyse both of the main aspects of the system, the data collection and processing back-end and the user interface web page. For the former, the Python scripts were tested by printing out the final values produced by the collection script in the command line and checking that the corresponding values in the listening program were the same to ensure the data was being transmitted correctly. This approach was used to print out the final values for each Python script in the command line to check they were producing the correct, valid results. Additionally, checking that the data was being correctly stored in the database was a case of entering the MariaDB program and running SQL commands to find all of the values in the table and making sure they were up to date with the latest values produced by the system.

For the user interface, a usability survey was created to be filled in by volunteer participants so that they could provide feedback on the system and how well it worked. The analysis of the responses to this survey can be found in the evaluation section of this report.

User Survey

A usability survey was created using Google Forms so that participants could provide their feedback on the system. A copy of this survey can be found in the appendix of this document, along with the information sheet and consent forms provided to participants. To fill in the report, a few people were approached via social media to request that they take part.

Once participants had been selected and filled in their consent forms, they were asked to perform a series of tasks on the user interface to help acclimatise them with the system and see how easy they found it. The tasks were as follows:

- Change timescale to hourly
- Change timescale to daily
- Change timescale to monthly
- Change to viewing future data
- Refresh the page to receive new values

The participants were then asked to fill in the survey based on their experiences with the system to provide feedback. These sessions took place online via the Discord application so that the participants could share their computer screens in order for the process to be monitored for note taking purposes.

Evaluation

Critical Reflection

As stated within the objectives laid out for the project, the main goal was to produce a system that allowed a user to place data collection devices wherever they would like and then use the readings from these devices to present the user with previous weather patterns in each location and predict future conditions as well. Additionally, a web page was to be created to act as a dashboard where the user could interact with the system. When measured against these metrics, the project can be deemed to have been a success as the final deliverables achieved these goals.

The process of building the project from start to finish through all of the different stages of development has been a largely enjoyable experience. It provided an invaluable insight into the software creation process as it required a hands-on approach to every stage over the course of a long period of time. While the system was at times a cause of frustration when it came to fixing errors and working out how certain aspects would be carried out, it was ultimately a positive undertaking as working through these issues boosted resilience and provided an enriching technical experience.

After the goals for the system had been laid out, research was the first step undertaken in the project to determine the feasibility of the system as well as what technologies, pieces of software and hardware would be required to make it run optimally. This phase took quite a while as many avenues needed to be explored but the decision to start here was ultimately a very good one as it provided a solid foundation on which to base the rest of the system. The reason this phase took so much time was due to the need to investigate new concepts, such as the methods used for weather prediction along with how the mathematical models that enabled this prediction worked as this was something had not been previously looked into. Additionally, having a lack of previous experience working with sensor components and their communication with computing devices meant that additional research was required to fully understand the process so that It might be implemented in the system properly.

Following on from the research phase was the design stage, in hindsight more time could well have been spent on this initially as some major decisions about the system architecture ended up happening in the development phase instead, such as what data the database needed to store and the layout of which it was stored in. This resulted in a few delays as changes had to be made in the code elsewhere to accommodate the new structure, however the agile approach taken to development proved to be very useful in this regard as it meant that it was flexible when changes and improvements needed to be made at the end of each development cycle.

The longest stage of the whole process, development, came next. Starting this part felt a bit overwhelming as there were many aspects to cover, spread out into many different areas so the decision to implement Trello as a way of keeping track of what needed to be done next was an extremely helpful one. The most efficient way to undertake the development section turned out to be focussing on each aspect of the system sequentially as described in the development section of the report, because this meant that each part could be fully implemented before worrying about the next section as opposed to trying to develop several aspects in different languages simultaneously. Having to develop in several languages on the same system and getting them to work together successfully was a great way to improve general coding skills as well as a good opportunity to see how this is handled in larger projects.

Lastly, the testing phase was unfortunately somewhat restricted in its scope. In an ideal world, the usability survey that was sent out would also have included allowing the participants to install the devices in a location of their choosing for about a week before they undertook the interview and survey, so as to gain further feedback into how users felt the system performed as a whole. Regrettably, due to restrictions brought about by the COVID-19 lockdown rules it was deemed to be too impractical to get the whole system to a group of different people and retrieve it again each time, therefore an alternative was reached where the participants would just review the user interface online via a video call where they would share their screens. Therefore, the feedback received was slightly impeded, however it did ultimately judge how a user would interact with the system and so it did provide some useful responses.

Overall, the technologies used proved to be very well suited for their purpose as they allowed for the creation of a fully functioning system that fulfilled the requirements of the project while also leaving scope to add new features and improvements further down the line. Additionally, the hardware obtained for the system largely did a good job of handling what was required of it, however if the project was to be done again then a different data sensor component than the Enviro for Raspberry Pi used in the system might need to be found. The reason for this is that the readings taken by the sensors proved to be a bit warmer than expected, this could well be due to the design of the sensor which sits directly above the Pis CPU, possibly causing excess radiating heat to be measured.

Successes

In terms of the deliverables, the final system achieved what was required of it. First of all, Devices were needed so that the user could place them in a location of their choosing to collect data which would then wirelessly transmit that information to a central location. This was achieved with the combination of the Raspberry Pi Zeros and Enviro for Raspberry Pi sensors that were very portable and could be placed anywhere with a network connection to send the aforementioned data to the central server via MQTT. Given that these devices produced valid readings and fit their specifications, this aspect of the project can be deemed to have succeeded.

Next was the need to have a central device that could listen for the incoming data, store it and use it to predict the future weather. The Raspberry Pi 4 selected in the research section turned out to be ideal for the purpose as expected because it performed all of these tasks without any performance issues. Using a combination of MQTT and MariaDB to handle the communication and storage aspects was highly successful as not only did it handle the amount of data needed for the system, but they were also extremely scalable so any future improvements such as taking readings of additional data types or adding more collection devices would not be an issue.

Having a clear and functional user interface web page was a vital requirement as it was the main area in which users would interact with the system. The one that was created fulfilled this need by providing a responsive, clear and functional web page that could be accessed on any device. This page displayed the stored and predicted data clearly through the use of graphs as well as a run-down table of the latest data values received by the system. According to the usability survey, all 100% of respondents either agreed or strongly agreed that the user interface was easy to understand and navigate, with 75% saying that they would not struggle to use the system without prior guidance or training. This clearly shows that the users felt that the interface met the requirements mentioned above

Limitations

There were a few issues that cropped up during the development of the system, one of the primary ones being to do with the mathematical model used for the prediction of future temperatures. As mentioned previously, the ARIMA model chosen allowed for the most accurate temperature readings over the short term as opposed to the SARIMA model, which would have been better over a longer timeframe. However, the amount of data collected by the system in the time it was running was still not enough to produce a result that was 100% accurate, the only way to combat this would be to just allow the system to continue collecting data and let the accuracy improve steadily over time. In the responses to the usability survey, only 50% of people felt that the predicted temperatures for the following weeks appeared to be valid possibilities, with 25% slightly disagreeing and the remainder unsure, showing that further improvement was needed. In addition to this, the model used only predicted future temperatures rather than humidity and pressure as well. This is largely due to the fact that readings beyond the scope of the project would be needed to have any chance of producing accurate results for these, such as atmospheric pressure readings taken over a wide area or the ability to take and analyse satellite pictures.

The Enviro for Raspberry Pi sensor also appeared to produce some slightly inaccurate readings as mentioned previously, therefore the data collected by the system was occasionally slightly incorrect. Choosing another sensor or array of sensors could have been a better option as not only might they provide slightly more accurate readings, but also could have improved upon the range of data variables collected, by taking account of factors such as wind speed and direction or precipitation levels.

Lastly, the responses to the usability survey showed that while the user interface was largely fit for purpose, some improvements could have been made such as having clearer axis labels on the graphs and units for the data types just to make the interface easier to use. One respondent also wrote that they would have liked more options for timescales over which to view the data. This is a good idea for future development; however, it was slightly outside the scope of the original design.

Future Development

If further work were to be undertaken on the project in future, the first task would be to set about fixing the issues mentioned in the limitations and implementing some of the recommendations from the usability survey responses. After these fixes had been made, the first major upgrade to make would be adding more sensor components to the collection devices so that they could collect more data types such as wind conditions and precipitation levels. Having more data types to work with would allow for an increased level of accuracy in any predictions made, so this would only serve to improve the system further.

One more thing would be to add additional options to the user interface such as more timescales to view the data over, but also a feature to allow the user to download any data to their local machine via a CSV file might well provide increased value to the system and give users more freedom.

Conclusion

Overall, the project has proven to not only be feasible, but also to a large extent successful. Further development could lead to an improved system in the future, however the final deliverable can be said to have met its original goals.

References

All3DP. (2021, January 21). *2021 Best Single Board Computers / Raspberry Pi Alternatives*. All3DP. https://m.all3dp.com/1/single-board-computer-raspberry-pi-alternative/

ArchLinux. (2021, March 27). *Cron - ArchWiki*. Archlinux.Org. https://wiki.archlinux.org/index.php/cron

Cifuentes, J., Marulanda, G., Bello, A., & Reneses, J. (2020). Air Temperature Forecasting Using Machine Learning Techniques: A Review. *Energies*, *13*(16), 4215. https://doi.org/10.3390/en13164215

Cprime, Inc. (2021, April 20). *What is AGILE? - What is SCRUM?* Cprime. https://www.cprime.com/resources/what-is-agile-what-is-scrum/

Heath, N. (2019, July 9). *What are the best Raspberry Pi alternatives? Everything you need to know about Pi rivals*. ZDNet. https://www.zdnet.com/article/what-are-the-best-raspberry-pi-alternatives-everything-you-need-to-know-about-pi-rivals/

Jankov, T. (2020, December 30). *MariaDB vs MySQL, a Database Technologies Rundown*. Kinsta. https://kinsta.com/blog/mariadb-vs-mysql/

King, P. (2019, December 27). *Enviro Raspberry Pi review*. The MagPi Magazine. https://magpi.raspberrypi.org/articles/enviro-phat-raspberry-pi-review

LePage, P., & Andrew, R. (2020, May 14). *Responsive web design basics*. Web.Dev. https://web.dev/responsive-web-design-basics/

Maker.IO. (2021, March 17). *The Top 10 HATs and pHATs for Raspberry Pi*. Https://Www.Digikey.Com/En/Maker/Blogs/2018/the-Top-10-Hats-and-Phats-for-

Raspberry-Pi. https://www.digikey.com/en/maker/blogs/2018/the-top-10-hats-and-phats-for-raspberry-pi

Met Office. (2009). *Clarity - Helping you understand the facts about weather forecasting*. https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/library-and-archive/library/publications/weather--climate-guides/weather_facts.pdf

Nielsen, J. (2020, November 15). *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/

Pedamkar, P. (2021, March 20). *Raspberry Pi 3 vs Arduino*. EDUCBA. https://www.educba.com/raspberry-pi-3-vs-arduino/

Pimoroni. (2021). *Raspberry Pi Zero W*. https://shop.pimoroni.com/products/raspberry-pi-zero-w?src=raspberrypi

Pounder, L. (2020, July 10). *Raspberry Pi vs Arduino: Which Board is Best?* Tom's Hardware. https://www.tomshardware.com/uk/features/raspberry-pi-vs-arduino

Puszynski, A. (2019, July 1). *These are the best JavaScript chart libraries for 2019*. FreeCodeCamp.Org. https://www.freecodecamp.org/news/these-are-the-best-javascript-chart-libraries-for-2019-29782f5e1dc2/

Python. (2021). *What is Python? Executive Summary*. Python.Org. https://www.python.org/doc/essays/blurb/

Rouse, M. (2020, February 11). *Internet of Things (IoT)*. IoT Agenda. https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

Serozhenko, M. (2018, June 12). *MQTT vs. HTTP: Which one is the best for IoT? - MQTT Buddy*. Medium. https://medium.com/mqtt-buddy/mqtt-vs-http-which-one-is-the-best-for-iot-c868169b3105#

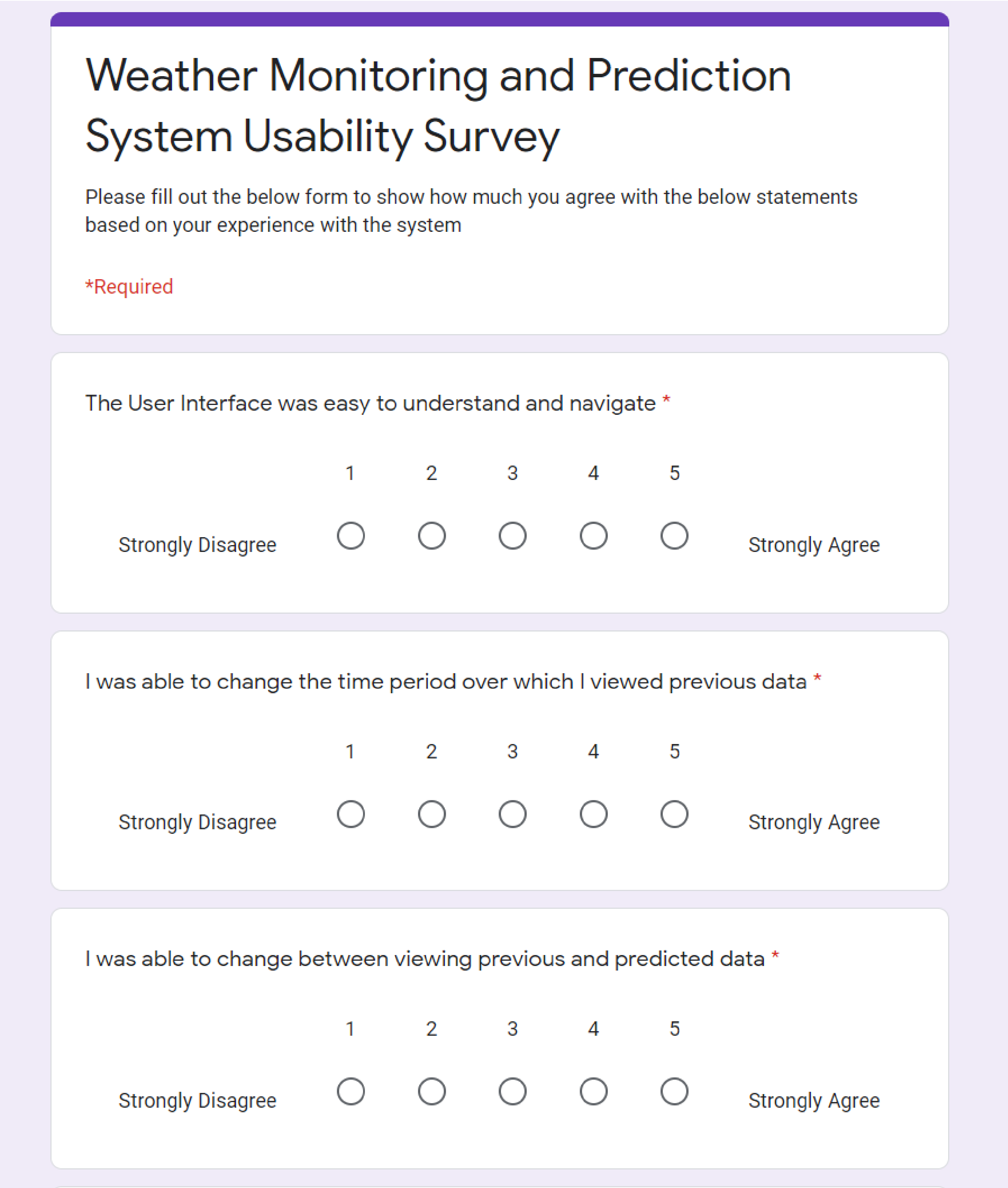
Sims, G. (2017, April 1). *Raspberry Pi Zero W review*. Android Authority. https://www.androidauthority.com/raspberry-pi-zero-w-review-756498/

Singhal, V. (2020, September 11). *20 best JavaScript charting libraries*. The Next Web. https://thenextweb.com/dd/2015/06/12/20-best-javascript-chart-libraries/

The PHP Group. (2021). *PHP: What is PHP? - Manual*. Php.Net. https://www.php.net/manual/en/intro-whatis.php

Appendix

Usability Survey



Weather Monitoring and Prediction System Usability Survey

Please fill out the below form to show how much you agree with the below statements based on your experience with the system

*Required

The User Interface was easy to understand and navigate *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I was able to change the time period over which I viewed previous data *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I was able to change between viewing previous and predicted data *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The use of graphs to display data made it clear to see weather patterns *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The predicted values for the next week appeared to be valid possibilities *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The system provides a useful service that I would use again *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The interface didn't crash and there were no obvious issues that affected my experience *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I always knew what data I was being presented with *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I think I would struggle to use this system without guidance or training beforehand *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Do you have any other opinions about the system and your experience using it during the interview *

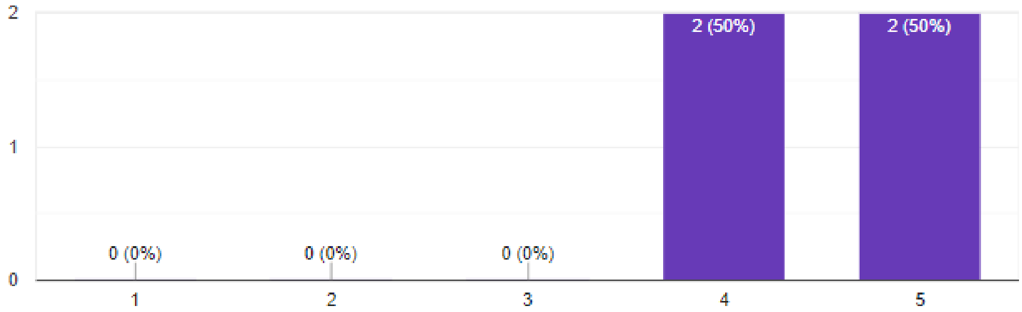
Your answer

Submit

Survey Responses

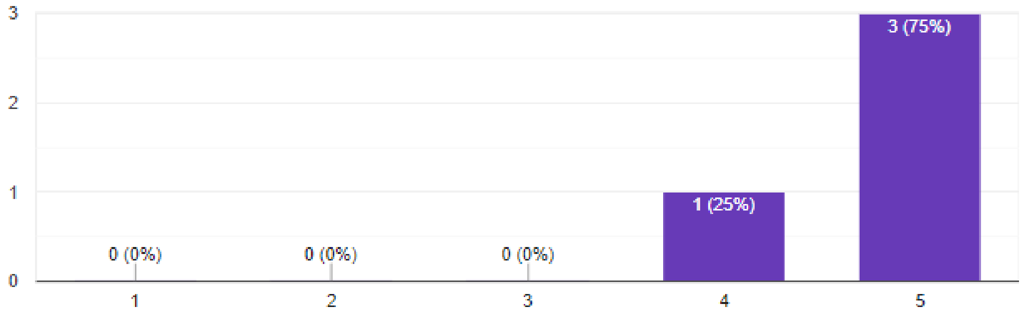
The User Interface was easy to understand and navigate

4 responses



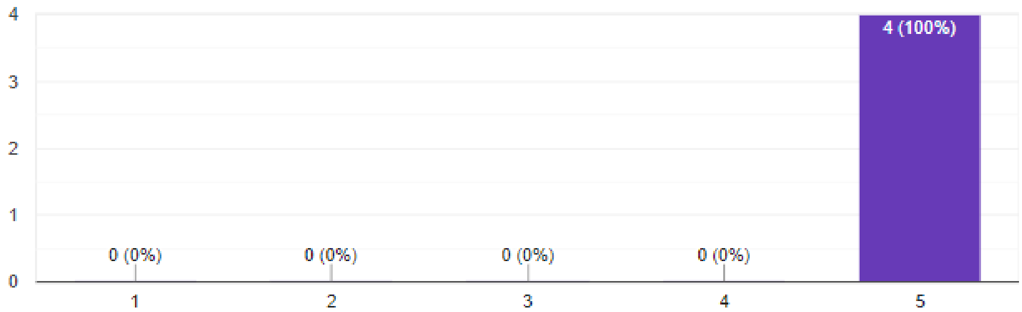
I was able to change the time period over which I viewed previous data

4 responses



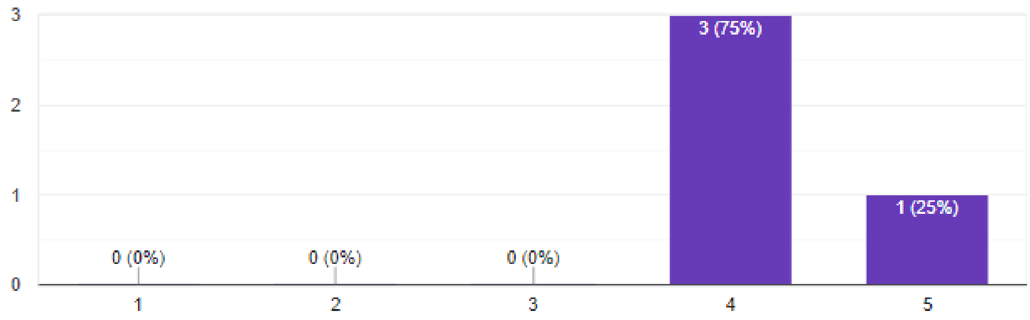
I was able to change between viewing previous and predicted data

4 responses



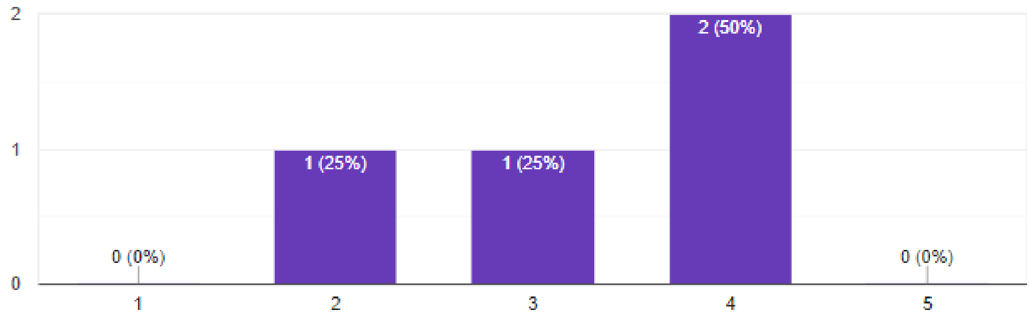
The use of graphs to display data made it clear to see weather patterns

4 responses



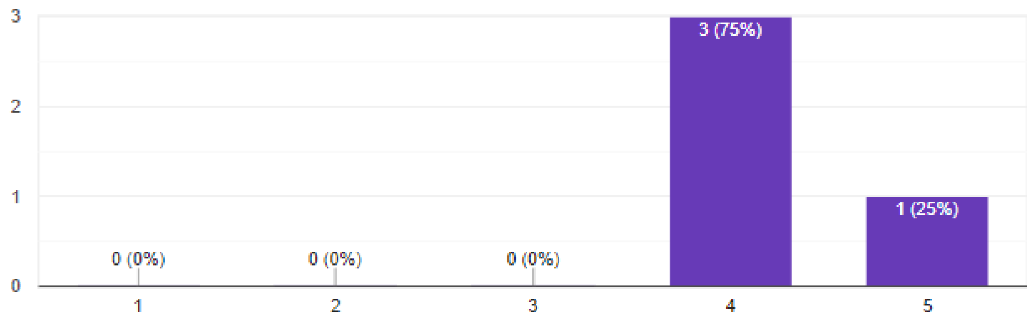
The predicted values for the next week appeared to be valid possibilities

4 responses



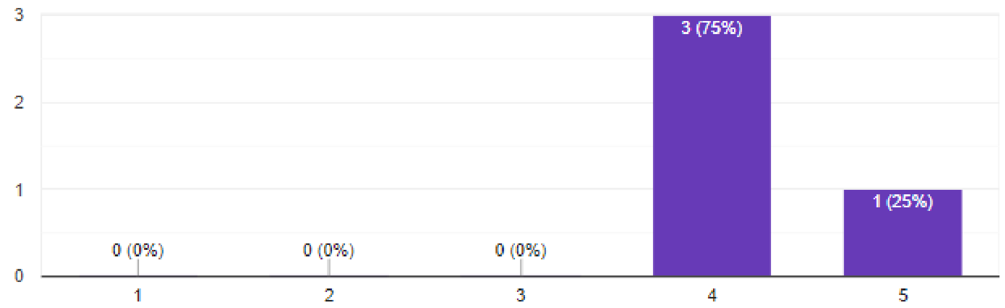
The system provides a useful service that I would use again

4 responses



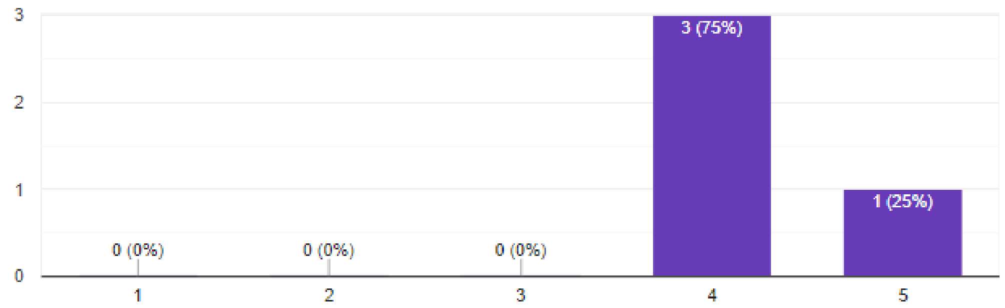
The interface didn't crash and there were no obvious issues that affected my experience

4 responses



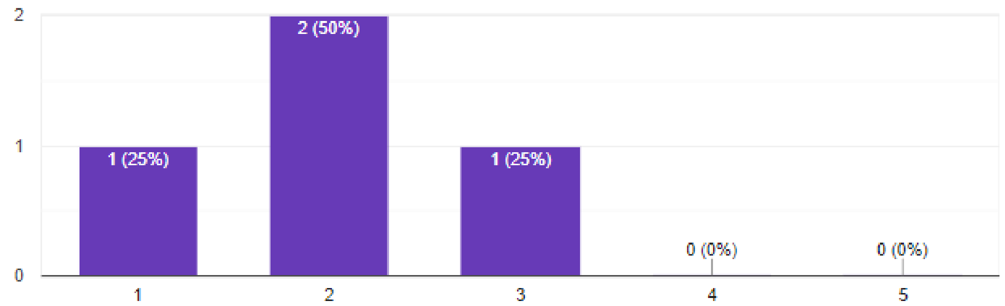
I always knew what data I was being presented with

4 responses



I think I would struggle to use this system without guidance or training beforehand

4 responses



Do you have any other opinions about the system and your experience using it during the interview

4 responses

Units for temperature, pressure and humidity could make the table and graphs clearer aswell as including axis labels.

The graphs seemed to bug out a little bit when hovered over on occasion

It was easy to use but more choice could be added for viewing the predicted information over a longer timescale

It appeared to be working fine but the predicted information didn't seem like it would be accurate

Consent Form

Participant Consent Form – Weather Monitoring and Prediction System Usability Survey

Please respond to the following statements by typing "Yes" or "No" in the response column:

Statement	Response
1. I have read the Information Sheet for this study and have had details of the study explained to me.	
2. Any questions I have had about the study have been answered to my satisfaction and I understand that I may ask further questions at any point.	
3. I understand that I am free to withdraw from the study within the time limits outlined in the Information Sheet, without giving a reason for my withdrawal or to decline to answer any questions in the study without any consequences to my future treatment by the researcher.	
4. I agree to provide information to the researchers under the conditions of confidentiality set out in the Information Sheet.	
5. I wish to participate in the study under the conditions set out in the Information Sheet.	
6. I consent to the information collected for the purposes of this research study, once anonymised (so that I cannot be identified), to be used for any other research purposes.	

Please fill in your details below:

Detail	Response
Participant's Name	
Participant's Signature	

Details for the researcher to fill in:

Detail	Response
Researcher’s Name	
Researcher’s Signature	

Information Sheet

Information Sheet – Weather Monitoring and Prediction System Usability Survey

Invitation
You are being invited to take part in a survey regarding a weather monitoring and prediction system using where you will be observed interacting with the user interface of the system and performing a few tasks to test it. All observations will be conducted remotely through a call where you will share your screen when using the system. This is an anonymous survey so no personally identifiable information will be recorded at any point. The Survey is being run by a student from Sheffield Hallam University. This survey aims to gather opinions on the usability of the weather system, specifically its user interface.

Before you decide to take part, please read this information.

Do I have to take part?
Participation is entirely voluntary. You are able to withdraw your contribution at any stage without giving a reason.

Activity consent
By taking part in the survey, you are giving the researcher permission to observe you using system remotely while you conduct a set task and for the opinions you give in the survey to be used for evaluation of the project and how the user interface can be improved.

All data given will be anonymous.

Project Specification

Student	Ryan DiDuca
Date	20/10/2020
Supervisor	Dr Babak Khazaei
Degree Course	BSc Computer Science
Title of Project	Modular weather monitoring and prediction application using IOT

Elaboration

Monitoring the weather and storing the resulting data not only allows you to keep an eye on current conditions, but more than this the data can be used to predict future weather patterns which can be useful for all sorts of reasons. Taking a modular approach to this means that data can be collected from multiple locations and thus it can either be compared or averaged out over a customised area to enable the user to review data and make predictions about the precise area they would like.

The aim of this project is to create a system that can monitor the weather in a modular fashion, this means that it will be able to take weather measurements such as temperature, humidity and barometric pressure in a variety of locations. One method to achieve this would be by having multiple monitoring nodes that each have sensors to record the measurements and utilise IOT to send them off to a central server. The nodes in this situation could be microcontrollers with the server being a single board computer in order to increase portability and reduce costs.

Users will be able to access the data via a web application connected to the server. The application will allow them to see the current conditions in each of the locations as well as historic data presented graphically. The user will be able to change the interval at which the system takes measurements as well as choose to take a reading manually. The system will then be able to take the previous data recorded and use it to make a prediction about future weather patterns in the area.

As part of my research, I will look into existing products and applications with similar aims to my project in order to gain more in depth knowledge of my projects domain. I will evaluate the success of my project by comparing the quantitative data produced by my application with existing weather prediction applications to see whether the generated data is accurate, both for reading the current conditions and predicting future ones.

Project Aims

- Investigate which hardware is most suited for the project.
- Investigate methods for predicting future weather patterns based on previous data.
- Research and source components such as sensors that will work well with the chosen hardware.
- Set up version control software to maintain and organise source code and documentation effectively.
- Create node(s) to monitor data by attaching necessary sensors to microcontrollers.
- Record weather data from node(s).
- Set up server to collect and store data from node(s).
- Create a web application and link to server to monitor data from node(s).
- Build page on web application to show current and historical data graphically.
- Implement login & registration system for web application.
- Implement weather prediction algorithm to predict the conditions in the area at a point in the future based on the data saved on the server.

Project Deliverables

Using a prototyping approach, I will create 3 nodes comprising of microcontrollers using sensors to take readings of the weather conditions, these will then be sent to a server which is to be created on a separate device. The data collected by the server will be able to be monitored online by means of a web application which presents the current data along with a graphical presentation of the historic data stored. The web application will also present a prediction of the future weather based on the past data stored. The project report will also be a deliverable.

Action Plan

Action	Deadline
Find a supervisor	9th October 2020
Upload project specification and Ethics form	30rd October 2020
Investigate which hardware is most suited for the project	6th November 2020
Investigate methods for predicting future weather patterns based on previous data and research similar existing products	13th November 2020
Investigate technologies that I can use in the development of my project	20th November 2020
Research and source components such as sensors that will work well with the chosen hardware	27th November 2020
The Information Review	4th December 2020
Set up version control software to maintain and organise source code and documentation effectively	4th December 2020
Create node(s) to monitor data by attaching necessary sensors to microcontrollers	18th December 2020
Record weather data from node(s)	1st January 2020
Set up server to collect and store data from node(s)	8th January 2020
Create a web application and link to server to monitor data from node(s)	22nd January 2020
Build page on web application to show current and historical data graphically	29th January 2020
Implement login & registration system for web application	12th February 2020
Implement weather prediction algorithm to predict the conditions in the area at a point in the future based on the data saved on the server	19th February 2020
The provisional contents page	19th February 2021
The draft critical evaluation	19th March 2021
Sections of a draft report	19th March 2021
Submit the project report via Turnitin and upload the report and deliverable via BB and/or the q drive	15th April 2021
Demonstration of the work	Before 29th April 2021

BCS Code of Conduct

I confirm that I have successfully completed the BCS code of conduct on-line test with a mark of 70% or above. This is a condition of completing the Project (Technical Computing) module. **Signature: Ryan DiDuca**

Publication of Work

I confirm that I understand the "Guidance on Publication Procedures" as described on the Bb site for the module. **Signature: Ryan DiDuca**

GDPR

I confirm that I will use the "Participant Information Sheet" as a basis for any survey, questionnaire or participant testing materials. This form is available on the Bb site for the module and as an appendix in the handbook. **Signature: Ryan DiDuca**

Ethics Form

UREC2 RESEARCH ETHICS PROFORMA FOR STUDENTS UNDERTAKING LOW RISK PROJECTS WITH HUMAN PARTICIPANTS

This form is designed to help students and their supervisors to complete an ethical scrutiny of proposed research. The University Research Ethics Policy should be consulted before completing the form. The initial questions are there to check that completion of the UREC 2 is appropriate for this study. The final responsibility for ensuring that ethical research practices are followed rests with the supervisor for student research.

Note that students and staff are responsible for making suitable arrangements to ensure compliance with the General Data Protection Act (GDPR). This involves informing participants about the legal basis for the research, including a link to the University research data privacy statement and providing details of who to complain to if participants have issues about how their data was handled or how they were treated (full details in module handbooks). In addition the act requires data to be kept securely and the identity of participants to be anonymized. They are also responsible for following SHU guidelines about data encryption and research data management. Information on the Ethics Website

The form also enables the University and College to keep a record confirming that research conducted has been subjected to ethical scrutiny.

The form may be completed by the student and the supervisor and/or module leader (as applicable). In all cases, it should be counter-signed by the supervisor and/or module leader, and kept as a record showing that ethical scrutiny has occurred. Some courses may require additional scrutiny. Students should retain a copy for inclusion in their research projects, and a copy should be uploaded to the relevant module Blackboard site.

Please note that it may be necessary to conduct a health and safety risk assessment for the proposed research. Further information can be obtained from the College Health and Safety Service.

Checklist Questions to ensure that this is the correct form

1. Health Related Research with the NHS or Her Majesty's Prison and Probation Service (HMPPS)or with participants unable to provide informed consent

Question	Yes/No

1. Does the research involve? • Patients recruited because of their past or present use of the NHS	No
• Relatives/carers of patients recruited because of their past or present use of the NHS	No
• Access to data, organs or other bodily material of past or present NHS patients	No
• Foetal material and IVF involving NHS patients	No
• The recently dead in NHS premises	No
• Prisoners or others within the criminal justice system recruited for health-related research*	No
• Police, court officials, prisoners or others within the criminal justice system*	No
• Participants who are unable to provide informed consent due to their incapacity even if the project is not health related	No
2. Is this a research project as opposed to service evaluation or audit? *For NHS definitions of research etc. please see the following website* http://www.hra.nhs.uk/documents/2013/09/defining-research.pdf	No

If you have answered **YES** to questions **1 & 2** then you **MUST** seek the appropriate external approvals from the NHS, Her Majesty's Prison and Probation Service (HMPPS) under their independent Research Governance schemes. Further information is provided below. https://www.myresearchproject.org.uk

NB College Teaching Programme Research Ethics Committees (CTPRECS) provide Independent Scientific Review for NHS or HMPPS research and initial scrutiny for ethics applications as required for university sponsorship of the research. Applicants can use the IRAS proforma and submit this initially to their CTPREC.

1. Checks for Research with Human Participants

Question	**Yes/No**
1. Will any of the participants be vulnerable? *Note: Vulnerable' people include children and young people, people with learning disabilities, people who may be limited by age or sickness, people researched because of a condition they have, etc. See full definition on ethics website*	No
2. Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants or will the study involve invasive, intrusive or potentially harmful procedures of any kind?	No
3. Will tissue samples (including blood) be obtained from participants?	No
4. Is pain or more than mild discomfort likely to result from the study?	No
5. Will the study involve prolonged or repetitive testing?	No
6. Is there any reasonable and foreseeable risk of physical or emotional harm to any of the participants? *Note: Harm may be caused by distressing or intrusive interview questions, uncomfortable procedures involving the participant, invasion of privacy, topics relating to highly personal information, topics relating to illegal activity, or topics that are anxiety provoking, etc.*	No
7. Will anyone be taking part without giving their informed consent?	No

8. Is it covert research? *Note: 'Covert research' refers to research that is conducted without the knowledge of participants.*	No
9. Will the research output allow identification of any individual who has not given their express consent to be identified?	No

If you have answered **YES** to any of these questions you are **REQUIRED** to complete and submit a UREC 3 or UREC4). Your supervisor will advise. If you have answered **NO** to all these questions then proceed with this form (UREC 2).

General Details

Name of student	Ryan DiDuca
SHU email address	b7015161@my.shu.ac.uk
Course or qualification (student)	BSc Computer Science
Name of supervisor	Dr Babak Khazaei
email address	B.Khazaei@shu.ac.uk
Title of proposed research	Modular weather monitoring and prediction application using IOT
Proposed start date	30th October 2020
Proposed end date	29th April 2021
Background to the study and scientific rationale for undertaking it.	The aim of this project is to create a system that can monitor the weather in a modular fashion, this means that it will be able to take weather measurements such as temperature, humidity and barometric pressure in a variety of locations. Using a prototyping approach, I will create 2 nodes comprising of microcontrollers using sensors to take readings of the weather conditions, these will then be sent to a server which is to be created on a separate device. The data collected by the server will be able to be monitored online by means of a web application which presents the current data along with a graphical presentation of the historic data stored. The web application will also present a prediction of the future weather based on the past data stored. My research for this project will include looking into which hardware is best suited for the nodes, server and sensors in the project in order to determine which ones would be fitting for the project. Also, I intend to research methods for predicting future weather patterns based on previous data so that I can implement the most accurate and efficient technique possible into my project. I will look into any existing similar products so that I can gain a better understanding of how similar products have handled issues relating to my projects domain. In addition, I will explore existing technologies relating to my project to ascertain which are the most

	suitable and feasible to implement. Over the course of this project, I hope to gain an in-depth understanding of how devices can interact with each other using IOT and how predictions can be made based upon extrapolated data.
Aims & research question(s)	The survey provided to the participants will provide insight into the usability of the system. The participants will be asked to respond with how much they agree with the following statements: 1) The User Interface was easy to understand and navigate 2) I was able to change the time period over which I viewed previous data 3) I was able to change between viewing previous and predicted data 4) The use of graphs to display data made it clear to see weather patterns 5) The predicted values for the next week appeared to be valid possibilities 6) The system provides a useful service that I would use again 7) The interface didn't crash and there were no obvious issues that affected my experience 8) I always knew what data I was being presented with 9) I think I would struggle to use this system without guidance or training beforehand 10) Do you have any other opinions about the system and your experience using it during the interview
Methods to be used for: 1.recruitment of participants, 2.data collection, 3. data analysis.	The survey will be provided via Google Forms and any resulting data will be analyzed to check for patterns and the most common responses for each question. Participants will be approached from a variety of backgrounds to try and get a more accurate view of how the average person views the system.
Outline the nature of the data held, details of anonymisation, storage and disposal procedures as required.	The participant consent form will be held, but all data provided within the survey will be anonymised to produce a set of responses that aren’t attached to any personal details. Once the project has run it’s course, or if the user requests for their details to be removed later on, this will be done immediately to remove any trace of them.

3. Research in Organisations

Question	**Yes/No**

1. Will the research involve working with/within an organisation (e.g. school, business, charity, museum, government department, international agency, etc.)?	No
2. If you answered YES to question 1, do you have granted access to conduct the research? *If YES, students please show evidence to your supervisor. PI should retain safely.*	
3. If you answered NO to question 2, is it because: A. you have not yet asked B. you have asked and not yet received an answer C. you have asked and been refused access. *Note: You will only be able to start the research when you have been granted access.*	

4. Research with Products and Artefacts

Question	Yes/No
1. Will the research involve working with copyrighted documents, films, broadcasts, photographs, artworks, designs, products, programmes, databases, networks, processes, existing datasets or secure data?	No
2. If you answered YES to question 1, are the materials you intend to use in the public domain? *Notes: 'In the public domain' does not mean the same thing as 'publicly accessible'.* • *Information which is 'in the public domain' is no longer protected by copyright (i.e. copyright has either expired or been waived) and can be used without permission.* • *Information which is 'publicly accessible' (e.g. TV broadcasts, websites, artworks, newspapers) is available for anyone to consult/view. It is still protected by copyright even if there is no copyright notice. In UK law, copyright protection is automatic and does not require a copyright statement, although it is always good practice to provide one. It is necessary to check the terms and conditions of use to find out exactly how the material may be reused etc.* *If you answered YES to question 1, be aware that you may need to consider other ethics codes. For example, when conducting Internet research, consult the code of the Association of Internet Researchers; for educational research, consult the Code of Ethics of the British Educational Research Association.*	
3. If you answered NO to question 2, do you have explicit permission to use these materials as data? *If YES, please show evidence to your supervisor.*	
4. If you answered NO to question 3, is it because: A. you have not yet asked permission B. you have asked and not yet received and answer C. you have asked and been refused access. *Note You will only be able to start the research when you have been granted permission to use the specified material.*	

Adherence to SHU policy and procedures

Personal statement	
I can confirm that: • I have read the Sheffield Hallam University Research Ethics Policy and Procedures • I agree to abide by its principles.	
Student	
Name: Ryan DiDuca	Date: 28/10/2020
Signature: Ryan DiDuca	
Supervisor or other person giving ethical sign-off	
I can confirm that completion of this form has not identified the need for ethical approval by the FREC or an NHS, Social Care or other external REC. The research will not commence until any approvals required under Sections 3 & 4 have been received and any necessary health and safety measures are in place.	
Name: Dr. Babak Khazaei	Date:29/04/2021
Signature: B. Khazaei	
Additional Signature if required by course:	
Name:	Date:
Signature:	

Please ensure the following are included with this form if applicable, tick box to indicate:

	Yes	No	N/A
Research proposal if prepared previously	☐	☐	☐
Any recruitment materials (e.g. posters, letters, etc.)	☐	☐	☐
Participant information sheet	☐	☐	☐
Participant consent form	☐	☐	☐
Details of measures to be used (e.g. questionnaires, etc.)	☐	☐	☐
Outline interview schedule / focus group schedule	☐	☐	☐
Debriefing materials	☐	☐	☐
Health and Safety Project Safety Plan for Procedures	☐	☐	☐