National College of Ireland

BSc in Computing

Internet of Things

2016/2017

Peter Barry

X12112631

peterbarry1@student.ncirl.ie

Raspberry Pi Weather Analyser

Thesis

Table of Contents

Executive Summary

This report delivers an examination of the present and potential expansion of a system in development. The system in question is a raspberry pi weather analyser. The system is designed to gather weather data for analysis so that trends can be shown. These trends will be displayed on high charts and hosted on a website for easy to read analysis. The system has already been tested in a prototype and it shows accurate readings.

Included in the appendices of the report are the project proposal and GUI. The project proposal explains what the project is about and who the target audience is. The GUI shows wireframes and explanations of the webpages that will be used as part of the system. Also included in the report are the technologies used in the project which include the hardware and software needed to have a fully functioning system.

As mentioned in section 4 of this report, further development and research shows that this system has the possibility to expand if given the proper resources. I feel that the system could attract the fitness and health market as a mobile wearable device. The system at the minute is designed for indoor use with the exception of the sensors that are capable of reading the elements outside, but I feel that the analyser could fit into the marketplace as a mobile device.

1 Introduction

This report shows how a raspberry pi, weather sensors and a computer will gather the information from the weather outside. It shows the integration of all these components needed to build a weather analyzer. When the system is fully implemented it will carry out regular readings of the elements so that I can conduct analysis on the data and compare it to recent readings. The report is put in place to show the accuracy of the system and as a reference to other similar systems that exist.

As this is a work in progress some information might not be included or accurate at this stage. Over the course of the project I will be new to some of the technologies used and familiar with other technologies. In this report, I will briefly explain the technologies both new and old. A list of the technologies used is available in the appendix.

1.1 Background

This project started out of interest in raspberry pi plus I get the chance of working with the latest technologies in the Internet of Things (IoT). I also get the opportunity to work with new programming languages which will further enhance my skillset. With the development of this project I can get a better understanding of weather systems and how this project can fit into the everyday lives of the users of the system.

Along with this it gives me a better understanding of raspberry pi and to see how the technology works. I'm really excited about the prospect of getting the system up and running, and to view the sensor readings of the elements that affect our daily lives. I am also looking forward to future projects with the raspberry pi after I have built up the skills and confidence on the completion of this project.

1.2 Aims

The reason behind this project is to build a weather analyser using a raspberry pi and outdoor sensors that reads the temperature & humidity, wind speed, rainfall and UV rays that occur on a daily basis within the city. As well as this I hope that the system can change the daily lives of the users of the system as a user can get up to date weather forecasts. Having this available to a user gives them the ability to plan ahead with any work or activities they might be carrying out outdoors.

As for myself, I aim to have the skills and knowledge needed for future employment working in the Internet of Things sector. Having knowledge of raspberry pi and python are two new skills I can sell to a future employer so it's a great interest to me to have a good working product that can give a user what they want.

1.3 Technologies

It will take several months to get the system fully functioning as a lot of coding will be required. Coding will be used to read the sensors and get them sending back the data to the user and database. For this the python language will be used to read and collect the sensor data which will be sent to the SQL database.

In regards to the displaying of the information I will have a website that displays the weather data on high charts. For this I will use HTML5, CSS3, JavaScript and the relevant API's. These software tools are vital for the successful deployment of the website. The HTML will display the data to a webpage. CSS3 will bring the appropriate style to the pages. The website will be hosted online using XAMP webserver and FileZilla to support and encrypt the data online.

JavaScript will make the webpages dynamic thus giving me the ability to display the high charts that shows the regular readings of the weather. The API's will be used to pull in the Open Weather library I will use for the site. I will most likely use the Google map API also. The below mentioned hardware and software will be needed to build the weather analyser system when combined.

Hardware:

- Raspberry Pi 3
- Raspberry Pi Sensor Kit
- LCD Display
- Raspberry Pi Enclosure
- Mouse & Keyboard combo

Software:

- Raspbian OS
- Python
- SQL
- HTML5
- CSS3
- JavaScript
- XAMP
- FileZilla
- API's

GitHub

This is a valuable tool for myself and the supervisor as I will host my system code on GitHub on a regular basis so that the supervisor and I have complete access to the system in case we need to modify any of the code.

Raspberry Pi 3

Raspberry Pi 3 is a small credit card sized single board computer. It is a Linux based system that lets a user write python command code. This code can be used to build various types of projects.

Raspberry Pi Starter Kit

This kit has some of the sensors needed to work with the weather system. It will have the temperature and humidity sensor needed to read the outside weather for the system. It will also have the rain sensor needed to measure the rainfall which is also needed for the system.

Wind Speed Sensor

This is another sensor that will be used in the system. It will be responsible for gathering the daily wind speeds needed in the weather system.

UV Sensor

This sensor will gather the UV rays during the day and relay the data back to the system for analysis.

Website

The website will involve coding the commands of the weather analyzer using Python and the website will be designed using HTML5, XML, CSS3, JavaScript and API's. The website will be used to display the weather charts which show the trends in the weather.

API's

There will be two API's used in the system. The first is the Open Weather API which is used to pull in the library used for the local and international weather. The second will be Google API which is used to pull in the high charts used for displaying the weather data.

Webserver

XAMPP is a free webserver that will be used to host my website online. This will give the freedom to display my data online thus avoiding costs that come with hosting a website.

FTP Server

As I have used it before and I'm somewhat familiar with it I will use FileZilla for transferring my web data. This will be responsible for hosting the web files used in the system.

1.4 System Cost

Figure 1

Components	Cost
Raspberry Pi 3 (RPI 3)	€35
Raspberry Pi Starter Kit	€45
7” LCD Display	€40
Raspberry Pi Enclosure	€11
Mouse & Keyboard Combo	€12
Micro SD Card 16GB	€6
Total Cost	€149

2 System

2.1 Requirements

Users should not have any problems running the system as everything will be menu driven in the GUI. Instructions on how to operate the system will be included within the website so that users can easily run the system.

2.1.1 Functional requirements

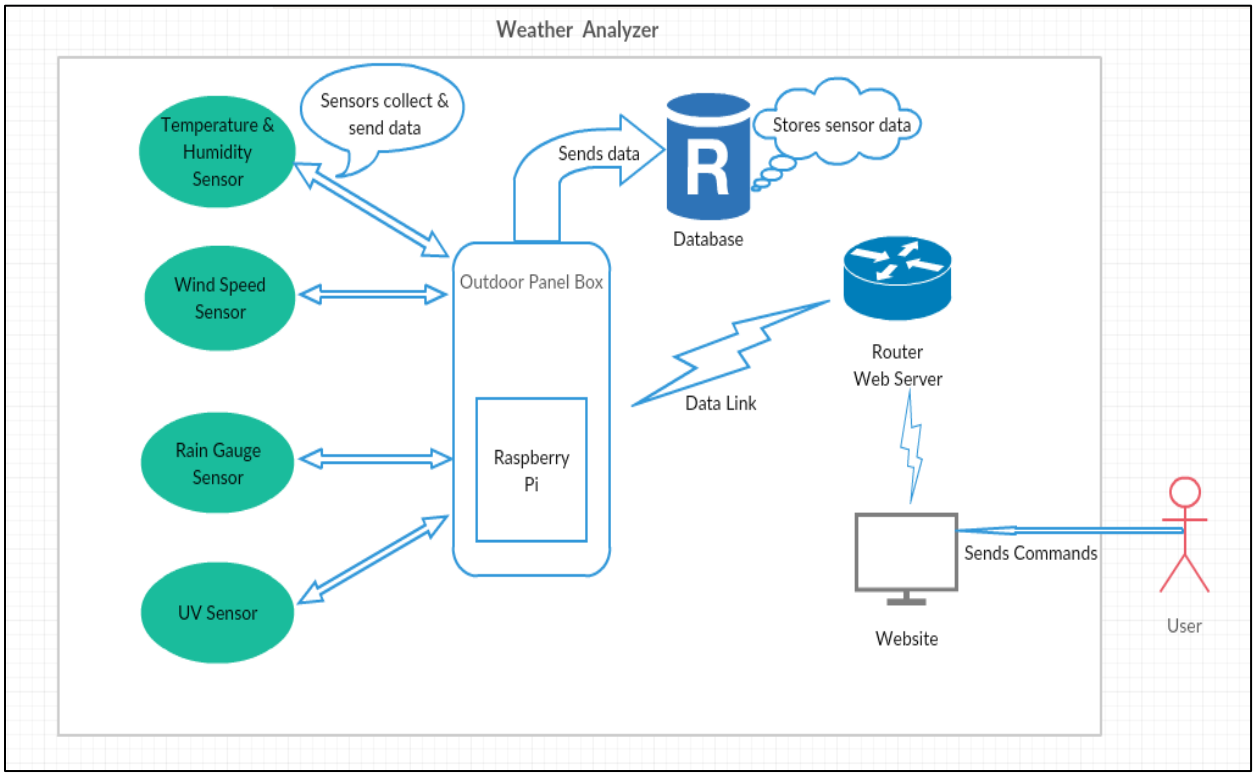
Functional

- Sensors
    - Data sensors should read the outside weather elements
    - Keep track of data readings on a regular basis
- Relational Database
    - Data should be exported
    - Correct code should be used
    - Check that readings are been stored regularly
- Website
    - Correct charts should be used for data readings
    - The website should be developed correctly to read data
    - Programming languages, database & API's should work together
- Responsive situations
    - Error handling
    - Recovery
- Relationship between I/O
    - I/O sequence

2.1.2 Use Case Diagram

Weather Analyzer

Figure: 2



2.1.3 Requirement 1 <Temperature & Humidity Sensor Reading>

2.1.3.1 Description & Priority

1. Sensor should collect data on a regular basis
2. Sensor should send back data
3. Sensor should display correct temperature & humidity
4. Sensors should be located in the correct system position
5. Sensors should extract correct data from the python code
6. Sensors should understand its correct function

2.1.3.2 Use Case

Scope

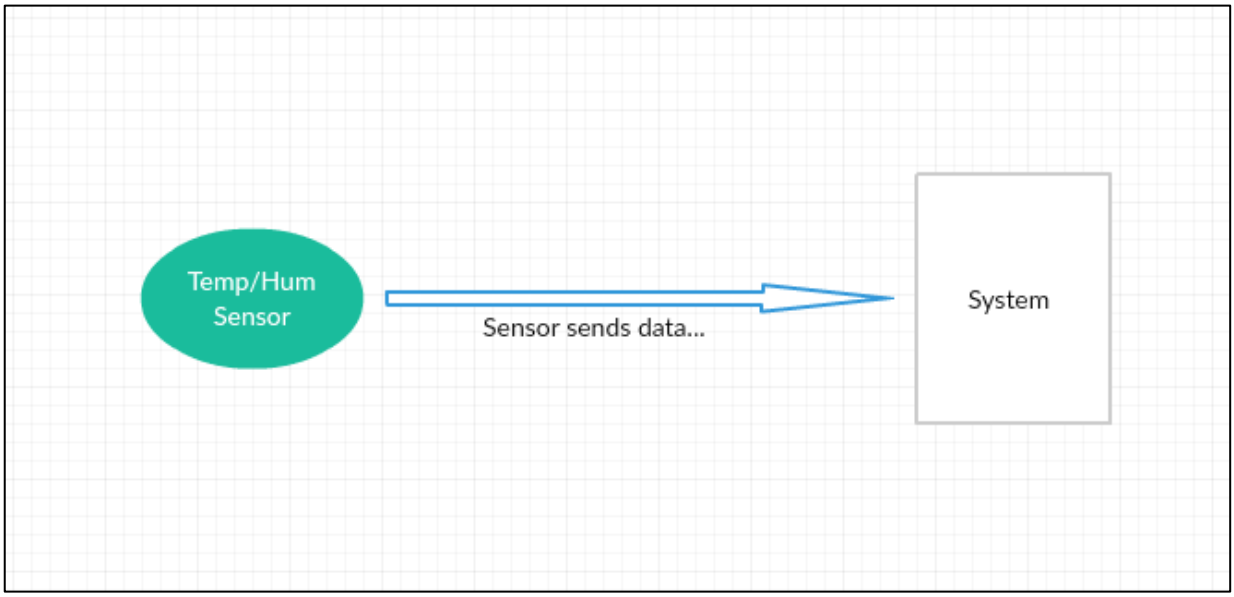
Getting the correct temperature & humidity readings is essential to the proper functioning of the system. The sensor should be responsible for sending, and receiving data so that the data can be sent back to the system, then stored to a relational database and is made available for reading on a website using data charts.

Description

The system shows the interaction between itself and the temperature & humidity sensor when sending data.

Use Case Diagram (Temperature & Humidity Sensor)

Figure: 3



2.1.3.2.1 Flow Description

Precondition

As we can see with the temp/hum sensor use case diagram in (figure 2) we know that the system sends the commands to the sensors which in turn collects the data from the weather elements outside and relays that data back for storage and analysis.

Activation

- The system is activated when the user sends the python commands to the system
- Activation starts when all requirements are satisfied
- System then activates sensors
- Sensors start collecting the data and sends it back to the system
- The data is stored
- Data is displayed on a website

Main flow

1. The system identifies the relationship between the sensor
2. The system sends instructions to the sensor
3. Commands are received by the sensor
4. The data is processed and sent back to the system
5. Data is received by sensors and stored on the system

Exceptional flow

E1: <Temp/Hum sensor error>

6. The system might run into difficulties due command errors
7. The user might not have connected the correct sensors to their pin slots
8. The user must use common knowledge when running the system

Termination

The system terminates when the user sends the python exit() command.

Post condition

The system has collected the temperature & humidity sensor data and is relaying this data correctly.

2.1.4 Requirement 2 <Wind Speed Sensor Reading>

2.1.4.1 Description & Priority

1. Sensor should collect data on a regular basis
2. Sensor should send back data
3. Sensor should display correct wind pressure
4. Sensors should be located in the correct system position
5. Sensors should extract correct data from the python code
6. Sensors should understand its correct function

2.1.4.2 Use Case

Scope

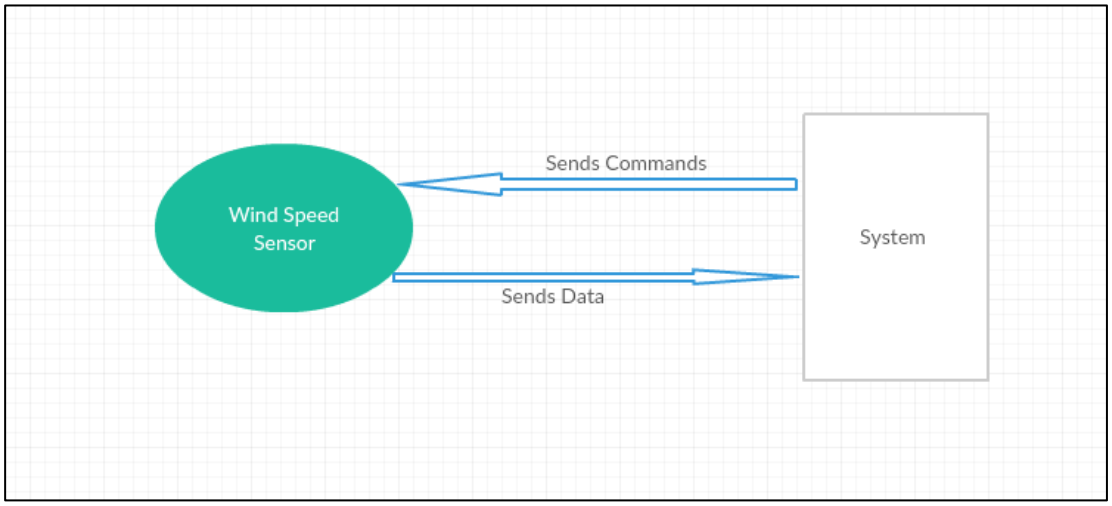
Getting the correct wind pressure readings should be relayed back to the system. The sensor should be responsible for sending, and receiving data so that the data can be sent back to the system, then stored to a relational database and is made available for reading on a website using data charts.

Description

The system shows the interaction between itself and the wind speed sensor when sending data.

Use Case Diagram

Figure 4:



2.1.4.2.1 Flow Description

Precondition

As seen with the wind speed use case diagram in (figure 3) we know that the system sends the commands to the sensor which in turn collects the data from the weather elements outside and relays that data back for storage and analysis.

Activation

- The system is activated when the user sends the python commands to the system
- Activation starts when all requirements are satisfied
- System then activates sensors
- Sensors start collecting the data and sends it back to the system
- The data is stored
- Data is displayed on a website

Main flow

1. The system identifies the relationship between the sensors
2. The system sends instructions to the sensors
3. Commands are received by the sensors
4. The data is processed and sent back to the system
5. Data is received by sensors and stored on the system

Exceptional flow

E1: <wind pressure sensor error>

6. The system might run into difficulties due command errors
7. The user might not have connected the correct sensors to their pin slots
8. The user must use common knowledge when running the system

Termination

The system terminates when the user sends the python exit() command.

Post condition

The system has collected the wind pressure sensor data and is relaying this data correctly

2.1.5 Requirement 3: <Rain Gauge Sensor Reading>

2.1.5.1 Description & Priority

1. Sensor should collect data on a regular basis
2. Sensor should send back data
3. Sensor should display correct rainfall
4. Sensors should be located in the correct system position
5. Sensors should extract correct data from the python code
6. Sensors should understand its correct function

2.1.5.2 Use Case

Scope

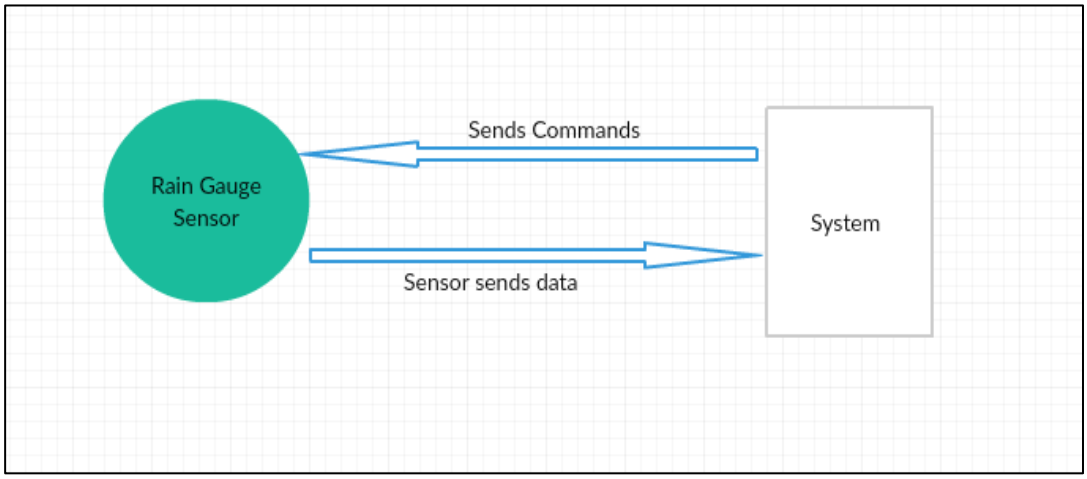
Getting the correct rainfall readings should be relayed back to the system. The sensor should be responsible for sending, and receiving data so that the data can be sent back to the system, then stored to a relational database and made available for reading on a website using data charts.

Description

The system shows the interaction between itself and the rain gauge sensor when sending data.

Use Case Diagram

Figure 5:



2.1.5.2.1 Flow Description

Precondition

As seen with the rain sensor use case diagram in (figure 4) we know that the system sends the commands to the sensor which in turn collects the data from the weather elements outside and relays that data back for storage and analysis.

Activation

- The system is activated when the user sends the python commands to the system
- Activation starts when all requirements are satisfied
- System then activates sensors
- Sensors start collecting the data and sends it back to the system
- The data is stored
- Data is displayed on a website

Main flow

1. The system identifies the relationship between the sensor
2. The system sends instructions to the sensor
3. Commands are received by the sensor
4. The data is processed and sent back to the system
5. Data is received by sensors and stored on the system

Exceptional flow

E1: <Rain gauge sensor error>

6. The system might run into difficulties due command errors
7. The user might not have connected the correct sensors to their pin slots
8. The user must use common knowledge when running the system

Termination

The system terminates when the user sends the python exit() command.

Post condition

The system has collected the wind pressure sensor data and is relaying this data correctly.

2.1.6 Requirement 4: <Ultra Violet Sensor Reading>

2.1.6.1 Description & Priority

1. Sensor should collect data on a regular basis
2. Sensor should send back data
3. Sensor should display correct rainfall
4. Sensors should be located in the correct system position
5. Sensors should extract correct data from the python code
6. Sensors should understand its correct function

2.1.6.2 Use Case

Scope

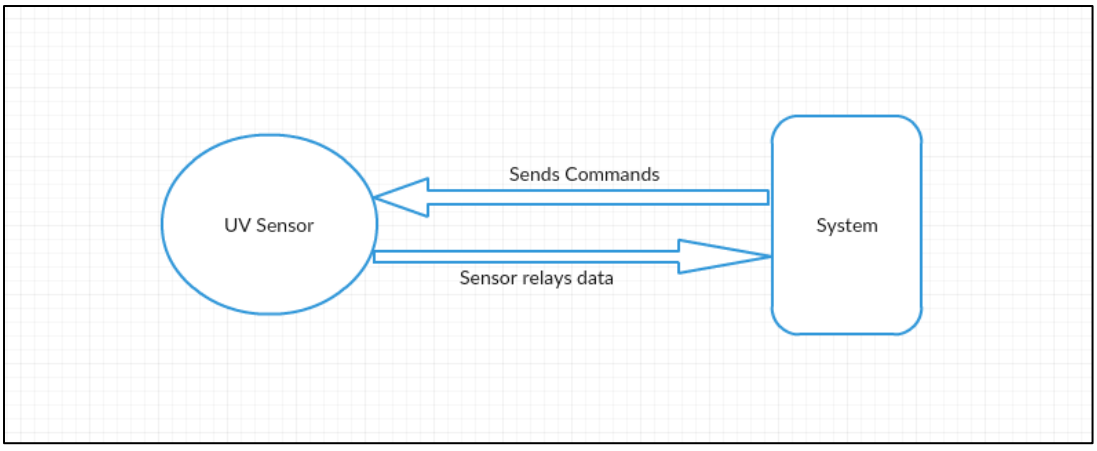
Getting the correct readings from the UV sensor should be relayed back to the system. The sensor should be responsible for sending, and receiving data so that the data can be sent back to the system, then stored to a relational database and made available for reading on a website using data charts.

Description

Figure 5 below shows the interaction between the system and the UV sensor when sending data.

Use Case Diagram

Figure 6:



2.1.6.2.1 Flow Description

Precondition

As seen with the UV sensor use case diagram in (figure 5) we know that the system sends the commands to the sensor which in turn collects the data from the weather elements outside and relays that data back for storage and analysis.

Activation

1. The system is activated when the user sends the python commands to the system
2. Activation starts when all requirements are satisfied
3. System then activates sensors
4. Sensors start collecting the data and sends it back to the system
5. The data is stored
6. Data is displayed on a website

Main flow

1. The system identifies the relationship between the sensor
2. The system sends instructions to the sensor
3. Commands are received by the sensor
4. The data is processed and sent back to the system
5. Data is received by sensors and stored on the system

Exceptional flow

E1: <UV nan sensor error>

6. The system might run into difficulties due command errors
7. The user might not have connected the correct sensors to their pin slots
8. The user must use common knowledge when running the system

Termination

The system terminates when the user sends the python exit() command.

Post condition

The system has collected the wind pressure sensor data and is relaying this data.

2.1.7 Requirement 5: <SQL Database>

2.1.7.1 Description & Priority

The database is a vitally important part of the system. The database used for this system will be a relational database. It will be responsible for collecting all the data from the sensors and store them for future analysis. Having the database will enable manipulation and management of data. Having a database included in the system lets us display data directly to our website so that the sensor data can be safely used with problems.

2.1.7.2 Use Case

Scope

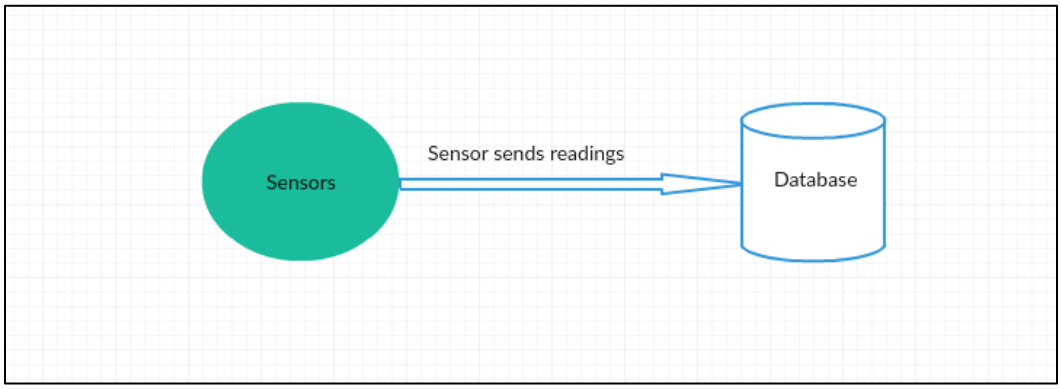
Having a database in place will allow for a smooth-running website. The database main role is to store sensor data but it is also responsible for relaying that data to the system for analysis and modification. Without the database, the system wouldn't be able to cope with the volume of data used during sensor readings as the system doesn't have large storage capabilities.

Description

The diagram in (Figure 6) below shows the interaction between the database and sensors.

Use Case Diagram

Figure 7:



2.1.7.2.1 Flow Description

Precondition

The data will be sent to the database by the sensors so that the database can store the information for analysis, retrieval and manipulation.

Activation

Activation begins when the sensors been used start to pull in the data which in turn is sent to the database.

Main flow

1. The database begins collecting the data from the sensors
2. The database analyses the data
3. Data is stored
4. Data is ready for easy retrieval for display with GUI

Exceptional flow

E1: <Server connection lost>

5. Is the SQL server running?
6. Is the password correct?
7. SQL server has different name from your PC

Termination

When the database has completed the first batch of sensor readings, it begins to read the sensors five seconds after the previous readings.

Post condition

When the database receives the data sensors readings it stores this data for execution at a later stage should the data need retrieving. The data can be displayed to the GUI when required.

2.1.8 Requirement 6: <Website LCD>

2.1.8.1 Description & Priority

The website will be used to view the system in its entirety. It will feature in the backend of the system to access the python command line which runs the code for the sensors. The LCD will show the website when it's complete and will display the sensor data charts.

2.1.8.2 Use Case

The website displays the sensor readings in easy to understand high charts that show trends in the weather over the course of time.

Scope

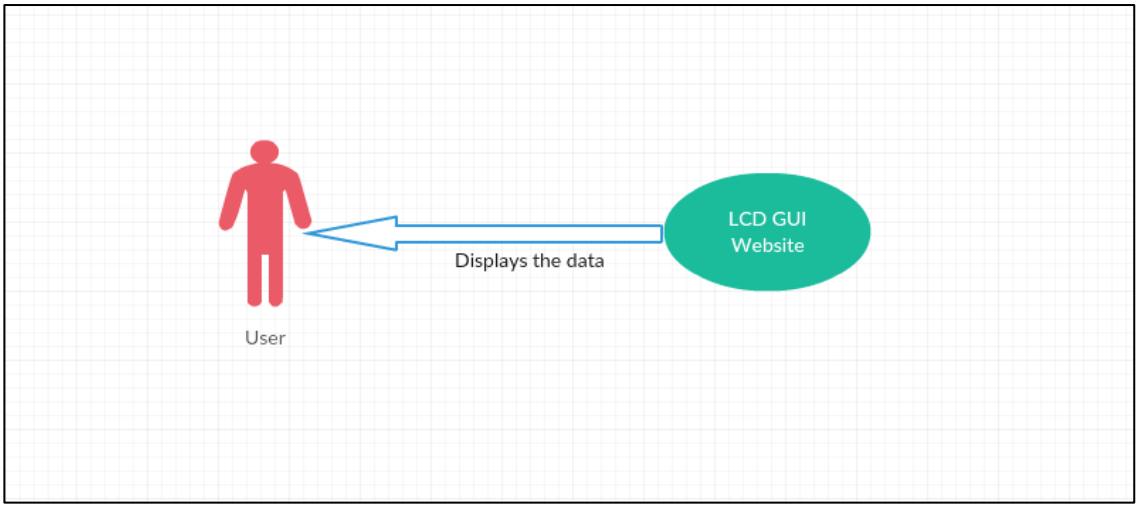
Been able to view the data in the website is the last step of the system so everything will be readily available for the user to work with and to easily understand.

Description

The image below in (figure 7) shows the interaction between the LCD and the user when the user views the data charts on the website.

Use Case Diagram

Figure: 8



2.1.8.2.1 Flow Description

Precondition

Having the website fully functional gives the user the ability to view the data charts on the website.

Activation

This use case starts when the database has stored the readings from the sensors. The database will then proceed to send the data to the graphs.

Main flow

1. The Website loads the data upon accessing
2. The website displays the data to the user
3. The charts display that data in easy to understand format

Exceptional flow

E1: <Not displaying content error>

4. Data is not being displayed on the website
5. Run error checks on chart JavaScript
6. Check the HTML
7. Use a validator to check the code

Termination

The website displays all the charts when accessed.

Post condition

The data will display the reading on weather charts on the website after which it will update on five seconds intervals.

2.2 User Requirements

2.2.1 User requirements

A user should be capable of using the system with ease so below are a list of the user requirements that should be adapted in order to create a smooth-running system.

- The system should give accurate readings.
- The GUI will be menu driven
- The system must boot quickly
- Sensors should collect accurate data
- The system should be controlled by one computer

2.2.2 Usability requirements

Performance/Response time requirement

With the system, the user gets updates from the system every 5 seconds when the charts trends change in to display the different types of weather events that are occurring outside. This feedback keeps been relayed to the user while the system is in read state. The charts should show temperatures in Celsius and give readings between -5c and 35c. Relative humidity should measure between 40% - 90%.

Availability requirement

The system will be available 24 hours a day when it's up and running as this is required to take weather readings from the elements occurring outside. The components of the system will be designed for permanent runtime as I will have a cooling system attached to the system enclosure.

Recover requirement

The most important thing to do when putting a recovery plan into operation would be to limit the occurrence of a disaster from ever taking place. To do this you would make sure that all the data is regularly backed up to the cloud or some type of storage that's off site and secure. Make sure that the system itself is secured with strong passwords and is encrypted.

Then you would have security that is physical meaning that the area were the system is kept would be equipped to deal with a fire and electrical faults.

Security requirement

- Make sure that the data is confidential
- Backup the data on a regular basis.
- Encryption of all data store on the system.
- Correct level of authorisation

Reliability requirement

All the software and hardware components will be consistently reliable when in run mode as the system will be check thoroughly for any runtime code errors before it can proceed with the data readings. As well as the data readings, the website will also be check for any errors that might stop the site from displaying the data charts.

Maintainability requirement

- The system will be able to easily see and fix any minor errors
- Modifications will be kept simple so that they're easy to apply
- All maintenance or upgrades will be kept at a minimal cost
- Costs are kept low as only one user is needed to run the system.

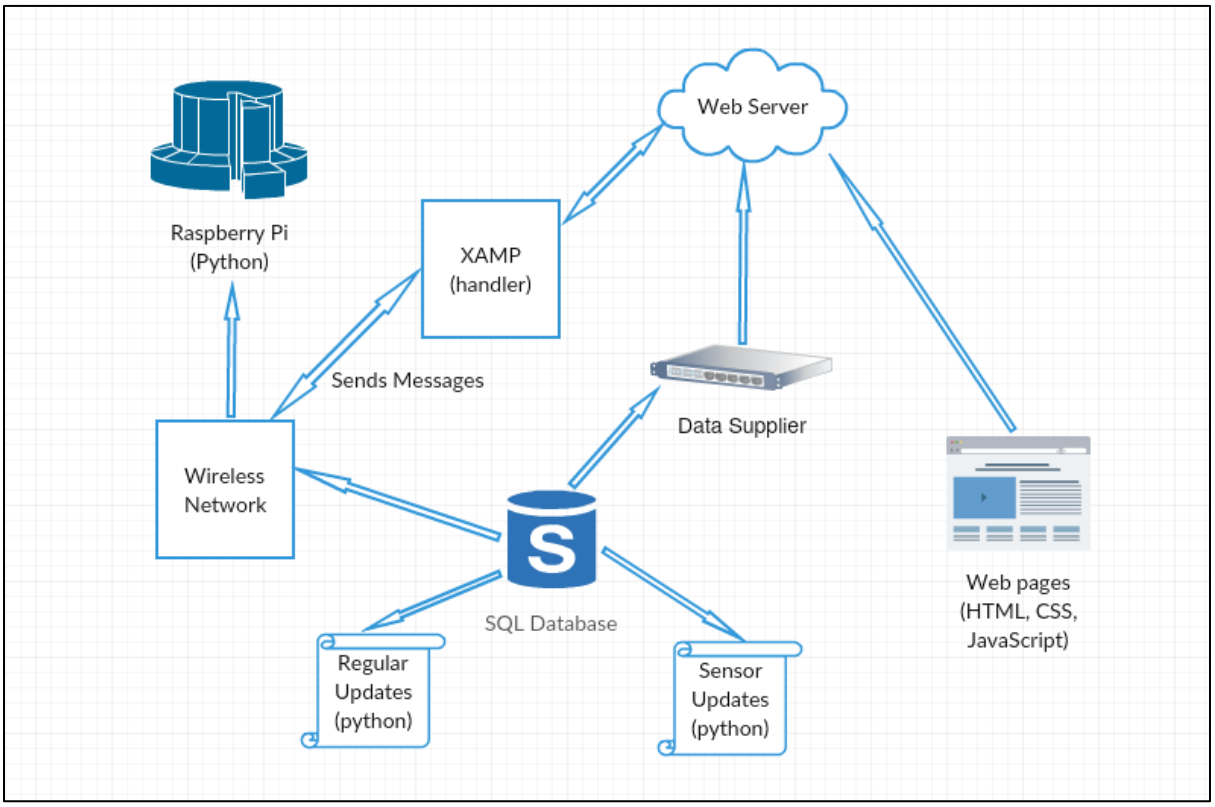
Portability requirement

The system (Raspberry Pi) runs on a Linux based system, and the website will be available to run with all the major operating systems. The GUI will be capable of running with all the main OS. The website will be designed using HTML5, CSS3, JavaScript & API's so that it will give the system a lot of portability and dynamics.

2.3 Design and Architecture

2.3.1 System Architecture

Figure: 9



2.3.2 Admin Login

The system will have an admin login page that lets tech savvy users make changes to the system. The user would have an assigned default password that he/she would be prompted to change when first accessing the admin page.

2.4 Graphical User Interface (GUI) Layout

Webpages used in the website for the Weather Analyser are shown below:

- Home Page
- Temperature/Humidity Sensor page
- Wind Speed Sensor Page
- Rain Gauge Sensor Page
- UV Sensor Page
- Sign in Page
- Registration Page
- Reset Password
- Contact Us Page

Home Page

This is the most important page of the website as it's the first page a user will see. So, with that in mind I wanted to make the homepage user friendly, informative and easy on the eye.

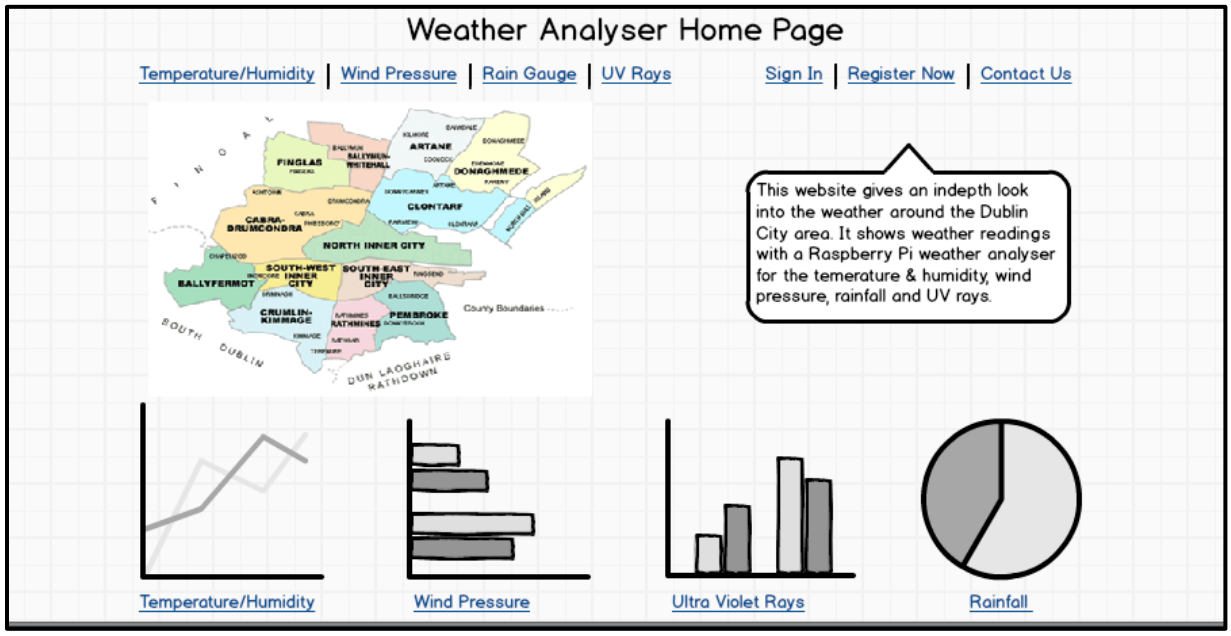
The homepage will give a brief description of what the website is about and what it will do. I will make the home page and relevant pages as professional as I can so to draw the attention of the user.

The page will also display hyperlinks for all the pages linked to the website.

Below is an image of what the home page will look like:

Home Page

Figure: 10

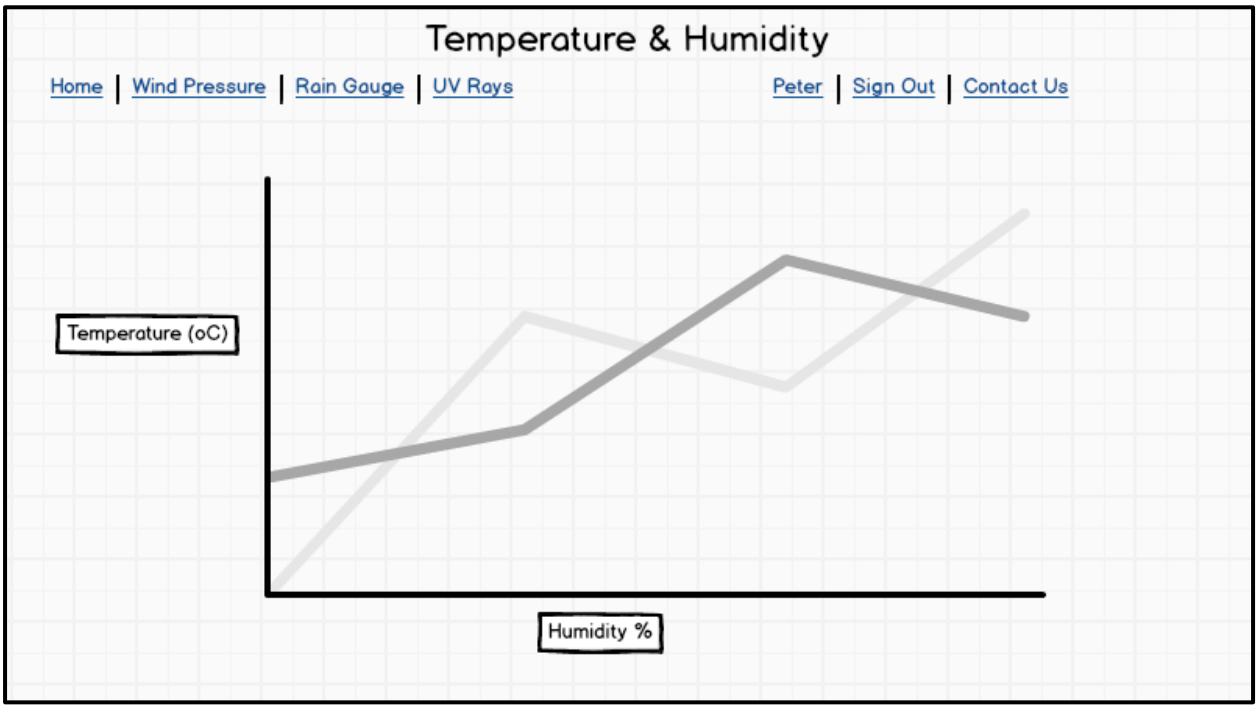


Temperature & Humidity Page

This page of the website displays the readings for the temperature and humidity. The information will be displayed on a chart showing the trends in the readings. The page will also display hyperlinks for all the pages linked to the website. Below is an image of what the temperature & humidity page will look like:

Temperature & Humidity Page

Figure: 11

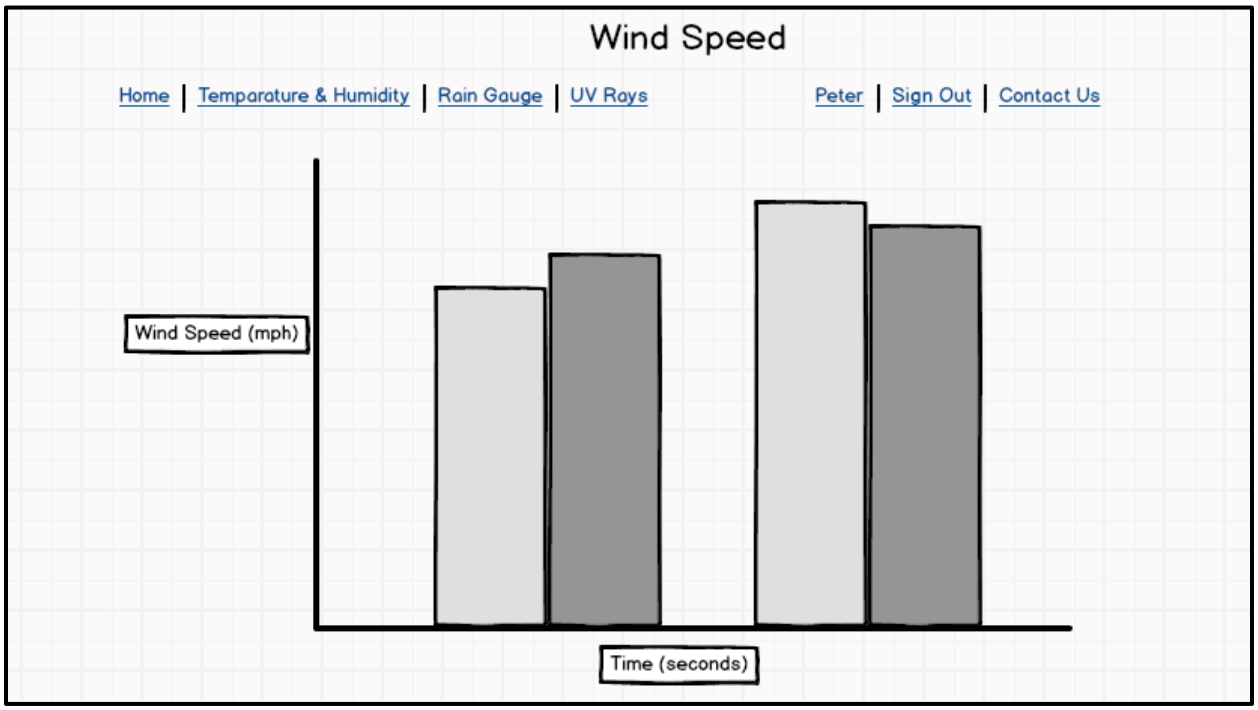


Wind Speed Page

This page of the website displays the readings for wind pressure. The information will be displayed on a chart showing the trends in the readings. The page will also display hyperlinks for all the pages linked to the website. Below is an image of what the wind pressure page will look like:

Wind Speed Page

Figure: 12

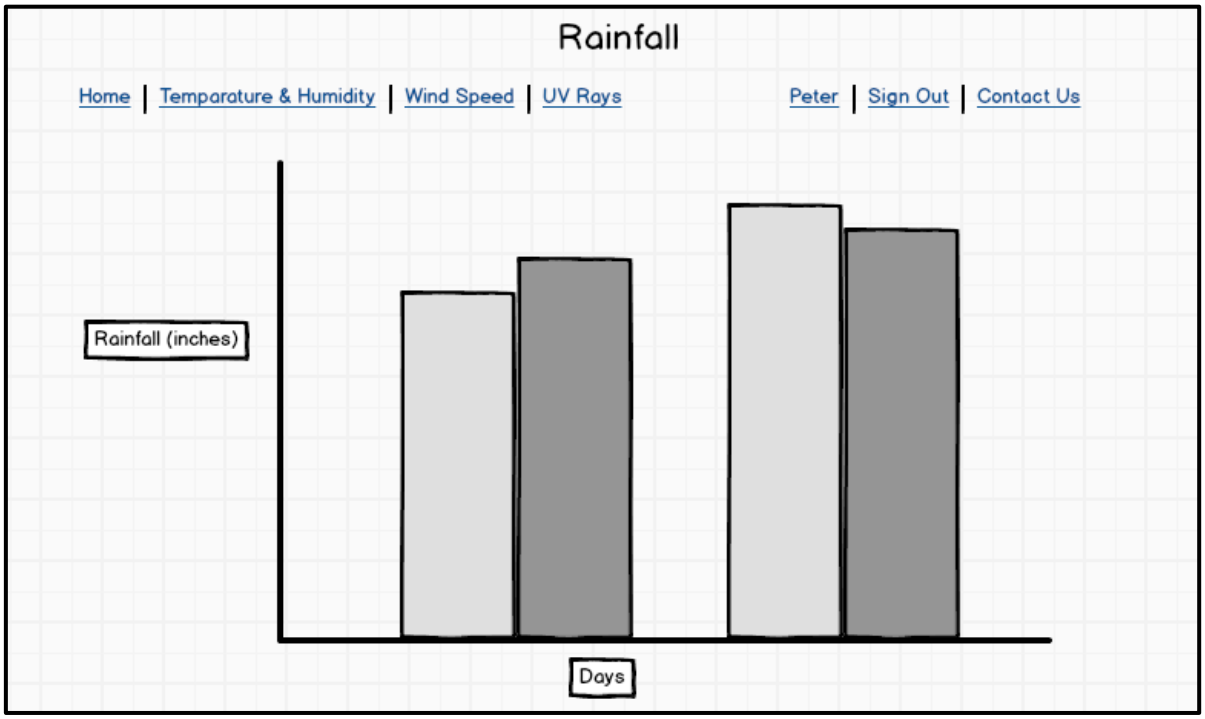


Rain Gauge Page

This page of the website displays the readings for rain gauge. The information will be displayed on a chart showing the trends in the readings. The page will also display hyperlinks for all the pages linked to the website. Below is an image of what the rain gauge page will look like:

Rain Gauge Page

Figure: 13

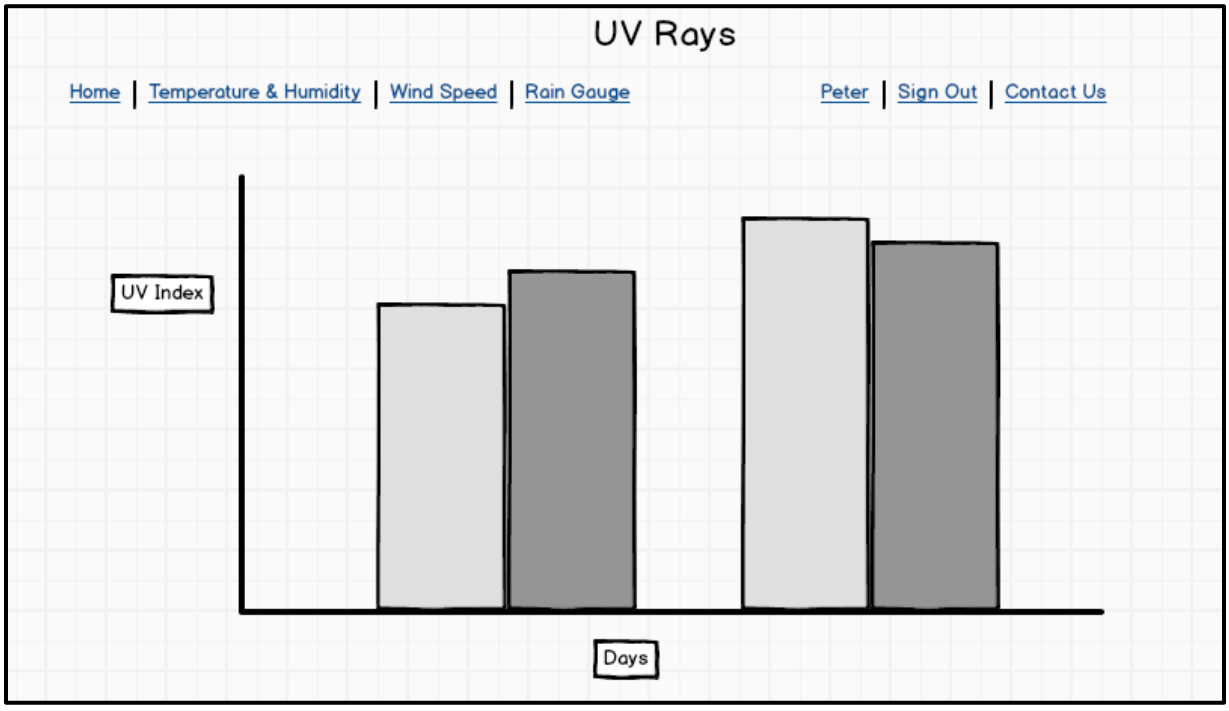


UV Rays Page

This page of the website displays the readings for ultra violet rays. The information will be displayed on a chart showing the trends in the readings. The page will also display hyperlinks for all the pages linked to the website. Below is an image of what the UV rays page will look like:

UV Rays Page

Figure: 14

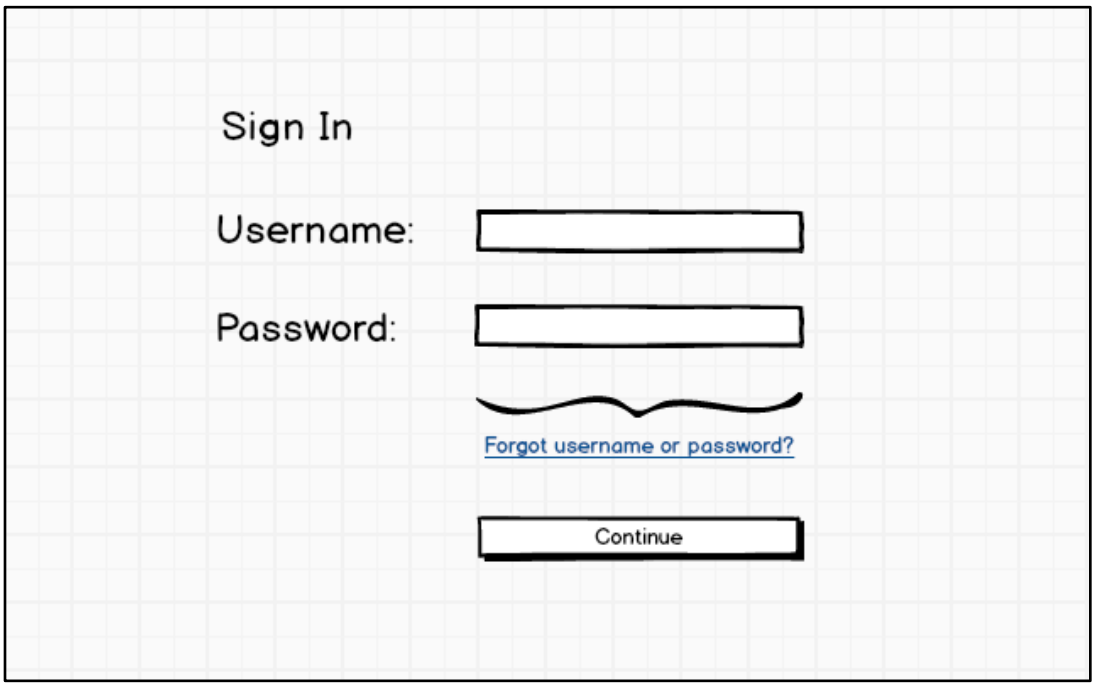


Sign in Page

This page of the website displays the relevant input boxes needed for signing into a user account. A username and password will be required to login into the system to access the data. Below is an image of what the sign in page will look like.

Sign in Page

Figure: 15

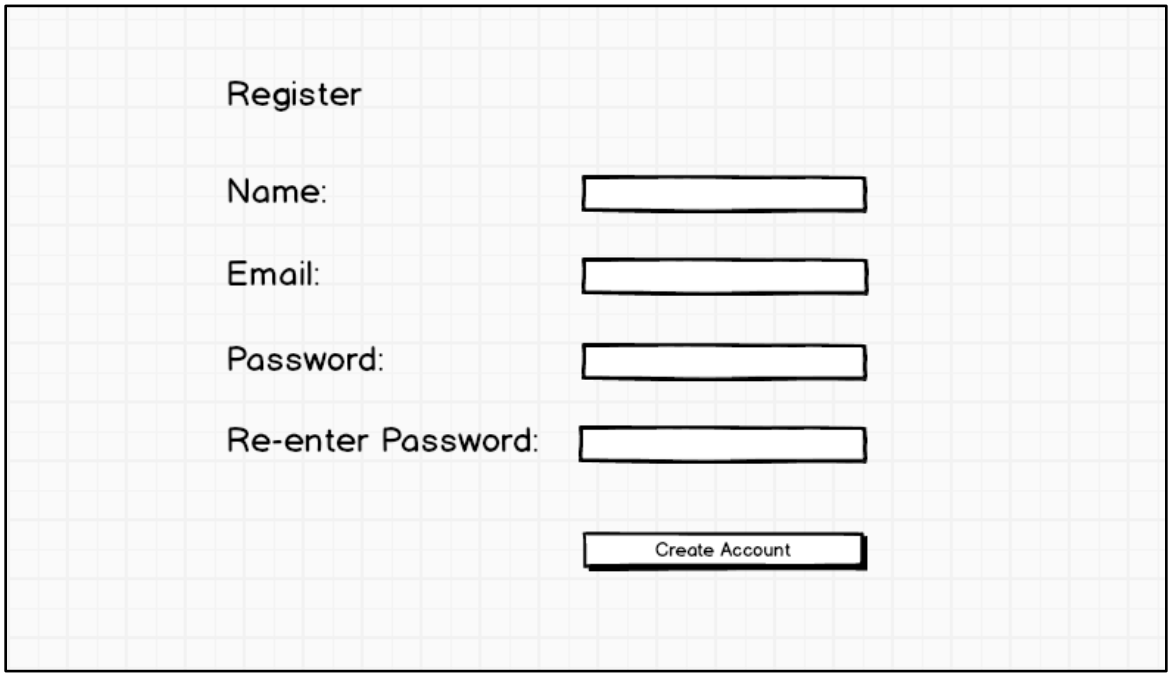


Registration Page

This page of the website displays the registration page and shows the relevant input boxes needed for setting up a new user account. The input fields needed for this include: name, email address, password and re-enter password. There will also be a create account button. Below is an image of what the sign in page will look like:

Registration Page

Figure: 16

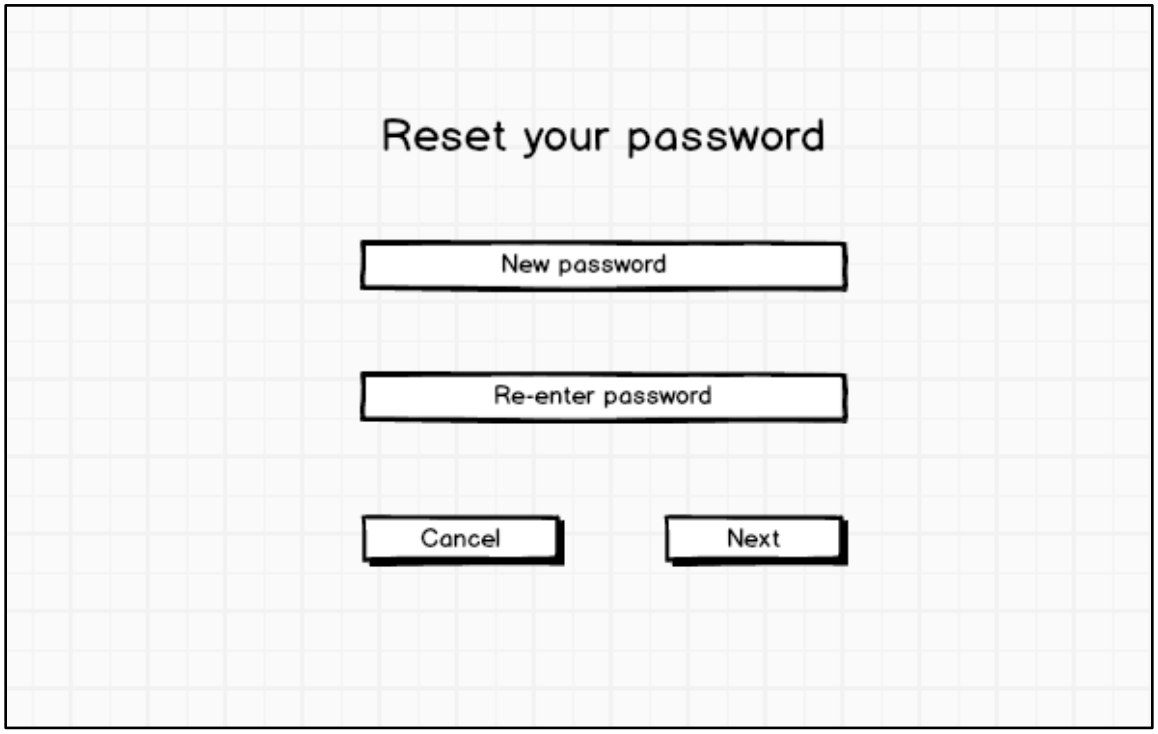


Reset Password Page

This page of the website displays the reset password page and shows the relevant input boxes needed for retrieving your account in the situation of forgetting your password. The input fields needed for this include: new password, re-enter password plus a cancel and next buttons. Below is an image of what the sign in page will look like:

Reset Password Page

Figure: 17

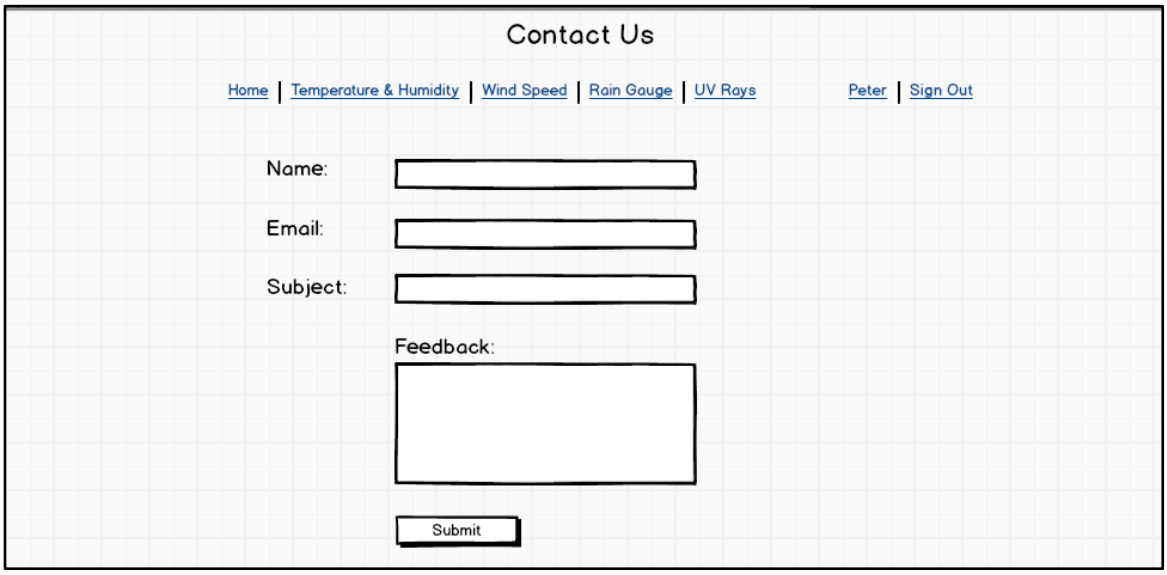


Contact Us Page

This page of the website displays the contact us page and shows the relevant input boxes needed for contacting the admin about any enquiries a user might have. The input fields needed for this page include: name, email, subject and feedback plus a submit button. Below is an image of what the sign in page will look like:

Contact Us Page

Figure: 18



3 Conclusions

I found the project to be extremely challenging as I was new to all of the technologies in the application. Trying and testing the technologies to work with each other took a long time to combine and get functioning. When I eventually got the application up and running and got to understand the technology better I found it all to be very satisfying. I hope to work a lot more with these technologies in the future and look forward to new challenges.

4 Further development or research

4.1 Wearable Devices

If giving the right resources I feel this system could become mobile for use as an app for smartphones and watches. The system would be extremely useful for walkers and hikers as a mobile wearable device connected with GPS. It would allow users to get regular checks and warnings of approaching bad weather systems that could prevent them from climbing peaks or walking trails.

4.2 Satnav's

It could also be installed into satnavs so that drivers of vehicles could get up to date alerts of the weather should they need it. Imagine long distance truck drivers having first-hand knowledge of the weather that lays ahead of their journeys. It could provide alternative routes should a route be obstructed.

4.3 Google Maps

The system could also feature in Google Maps as an added feature as the navigation API in their app is widely used by a lot of drivers around the globe.

5 Appendix

5.1 Project Proposal

5.2 Introduction

The aim of this project is to get a better sense of the weather of today, and more specifically the weather in my home city of Dublin and surrounding areas. The weather is constantly changing as the years pass so my project will chart the changes and compare it to past data from outside sources. I hope, when completed to get a better understanding of the ever-changing weather cycles that's been affecting our climate over the years.

5.3 Objectives

I plan on developing a weather analyser with a Raspberry PI 3 model and using Python as my main programming language. As well as this I will use a website to display the data using weather charts for all the different types of weather events. These events would include wind speed, temperature & humidity, rainfall and UV rays. Sensors are another piece of hardware that will be used in order to measure the weather temperature outside.

For this feature, I plan on using python library, and the "Open weather Map" open source API to pull weather data from around the globe. The website will be hosted online using XAMP webserver and FileZilla to support and encrypt the data online.

As the project is only in its infant stage, I will add more to it as I go along. I plan on testing all the components thoroughly before implementation so that I don't run into any problems further down the line. Below is a description of some of the software and hardware that I plan on using in my project.

Software

- Website to display the data
- XAMP
- FileZilla
- Weather graphs
- Python
- Python library
- Open Weather Map API

Hardware

- Raspberry Pi 3 Model B
- Raspberry Pi Starter Kit
- Raspberry Pi Monitor
- 16GB Class 10 SD Card
- LCD Display
- Power Supply
- Raspberry Pi Breadboard
- Enclosure for the Pi
- Sensors

5.4 Background

As a fourth-year student who is specialising in the Internet of Things (IoT) I decided to work with this project as it gives me the opportunity to work with the latest technologies, which include Raspberry Pi 3 model and components as mentioned in the objectives section above. I will liaise with my college supervisor in creating a weather analyser that will read the current weather in my hometown and surrounding areas. The early stages of the project included research and trying to understand how to create the necessary documentation for a software project.

After been approved for the project I began doing research into how I could design a weather analyser that could be better or would improve on what's already in the marketplace. The majority of the code I will use in my project and on Raspberry Pi 3 will be Python.

The website will be developed with HTML5, XML, CSS3, JavaScript and the relevant API's. This will be the first time using the Python language and it's something I'm excited about. It also gives me the chance to learn a new programming language going forward. The site will also use LAMP webserver to host the site and FileZilla to store and encrypt the data.

After meeting with the college supervisor, we both decided to go with the idea for the weather analyser. We both agreed that there would be room for this particular software if the idea was expanded upon. I'm going to carry out further research and see what features can be added to the project that doesn't exist in similar weather sites. The first phase will include the gathering of research and drawing up the documentation for the project. The documentation will continue for several months after which I will start working on the backend of the project.

During the next couple of months, I will liaise with my college supervisor to get feedback on my work and improve on my idea should the need arise. I also plan on listening to any new ideas or features that could be added to the project, as I'm new to this area and I hope to get as much advice and guidance as possible. There will be several changes to the project as time passes. Features could be added or taken out over the course of the project, but I plan on creating a project that fits into the marketplace of today.

5.5 Technical Approach

The purpose of this project is chart the weather using the latest technologies and to see if there is a need for this in the already crowded marketplace. I plan on building a weather analyser that has features not found in any other Raspberry Pi project.

Over time I will add improvements that I hope will be of use to me and hopefully other users when I launch the project for public use. I hope to release the project online when I have it completed so that other novices can get an understanding of Raspberry Pi projects.

Activities over the course of the project will vary from theory in the early stage that will consist of the appropriate documentation needed to carry out a Raspberry Pi weather analyser. This stage will take several months after which I will begin work on the backend end of the project. The backend will involve coding the commands of the weather analyser using Python and the website will be designed using HTML5, XML, CSS3, JavaScript and API's. The website will be used to display the weather charts which show the trends in the weather. XAMP webserver will be used to host the site and FileZilla will be used as the FTP server store, host and encrypt the data online.

5.6 Technical Details

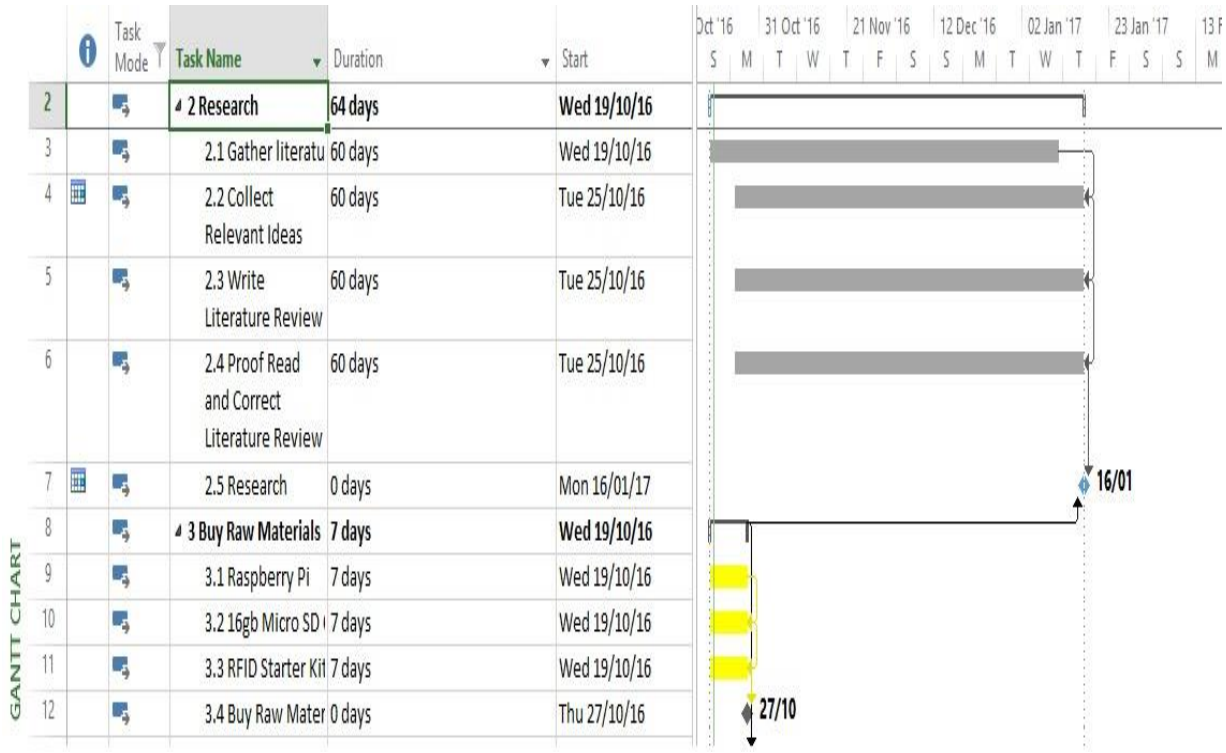
Below is a list of some of the technologies that will be used in the project. This will undoubtedly change over the course of the project.

- Raspbian OS
- Python
- HTML5
- CSS3
- XAMP
- FileZilla
- XML
- JavaScript
- API's

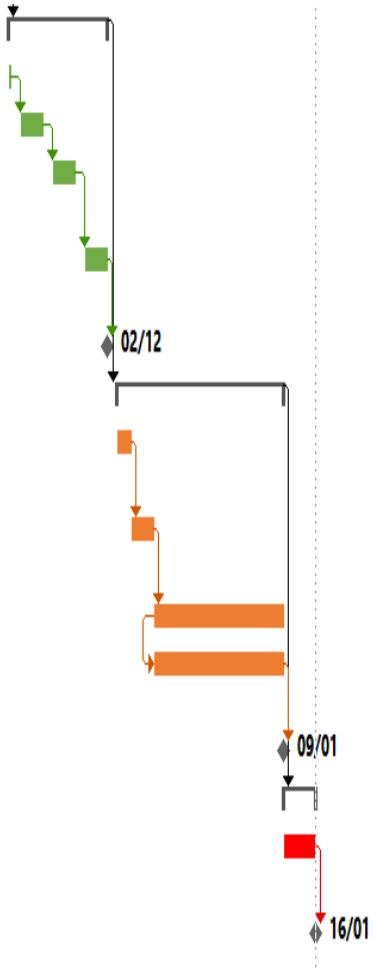
Peter Barry 11/12/2016

6 Project Plan

Figure 17



27		**6 Coding**	**15.5 days**	**Fri 11/11/16**
28		6.1 Set Up GIT HUB	0.5 days	Fri 11/11/16
29		6.2 Practice Code	5 days	Mon 14/11/16
30		6.3 Tutorials for Python Code	5 days	Mon 21/11/16
31		6.4 Store Data with Cloud 9 for	5 days	Mon 28/11/16
32		6.5 Coding	0 days	Fri 02/12/16
33		**7 Prepare Prototype**	**26 days**	**Mon 05/12/16**
34		7.1 Have Working Pi - Fix Bugs	3 days	Mon 05/12/16
35		7.2 Have Working Code - Fix Bugs	3 days	Thu 08/12/16
36		7.3 Collect Test Da	20 days	Tue 13/12/16
37		7.4 Collect Temp & Humidity Data	20 days	Tue 13/12/16
38		7.5 Prepare Protot	0 days	Mon 09/01/17
39		**8 Final Working Devi**	**5 days**	**Tue 10/01/17**
40		8.1 Present Final Project and	5 days	Tue 10/01/17
41		8.2 Final Working I	0 days	Mon 16/01/17



7 Definitions, Acronyms & Abbreviations

Class	Object that contains a description of the data within the project
CRUD	Create Retrieve Update Delete
CSS	Cascading Style Sheets is a language that's used to define the appearance of a document in another markup language.
Database	Storage location for the data
Developer	Responsible for the development of the system
GPIO	General-purpose input/output is a generic pin used on an integrated circuit or computer board.
HTML5	Hypertext Mark-up Language used for the structure and content on WWW
JavaScript	Object Orientated Programming language that's used in the creation of effects in web browsers.
Project	A collection of models stored in separate files
Python	Widely used high-level, general- purpose, interpreted, dynamic programming language.
Relationship	Connection between two classes.
Requirement	A condition needed by a user to solve a problem or achieve an objective.
System (RPI 3)	Raspberry Pi 3. Credit card sized single board computer.
GUI	Graphical User Interface.

Use Case	A method used to describe the behaviour of a component of the system.
User	The person using the system.

8 Monthly Journals

8.1 September Journal

Student Name: Peter Barry

Number: X12112631

(BSc in Computing): Internet of Things

Final Year Software Project

Month: September

Introduction

My name is Peter Barry and I'm a fourth year BSc Honors in computing student specialising in IoT (Internet of Things). This software project will be my final year project, and hope to cover most of the areas I've studied over the course of my college studies. I will also include some of the new technologies that I'm covering in my final year and bring them together to complete a working project. I plan on starting work on the project in the month of October, which will include some the project documentation. The project is due for completion in May 2017.

My Achievements

The month of September was spent on research and trying to come up with a project that wasn't already on the market. I decided to go with the idea of developing a photo sharing travel app. I did a lot of research into this and it seemed that there could be a market for this type of app as there wasn't an app solely dedicated to sharing travel photos.

My Reflection

I hope that my idea gets approval when the project pitch presentations come around in October.

I feel I can work to my best ability as I love to travel and take a lot of photos, so designing an app that lets a user share his/her photos while travelling is an area that's interesting to me. I'm still thinking up new features that can be added to the app so it will be good to design these add-ons if I get the go ahead with the project.

Intended Changes

I will add new features to the app if giving the approval from the college. I hope to add a loyalty feature and a GPS feature. I also intend on adding a secure register/login system for new and returning users of the app. As its still in the infant stage of the project I plan on adding many features to the project.

8.2 OCTOBER JOURNAL

Student Name: Peter Barry

Number: X12112631

(BSc in Computing): Internet of Things

Final Year Software Project

Month: October

My Achievements

The start of the month involved meeting with a college lecturer to discuss the idea for a weather analyser using Raspberry Pi. During the lecturer and I discussed how we could make the project better and who would be our target audience.

During this month, I also created my project proposal and discussed my ideas for the project go forward. I also highlighted what was needed to get the project off the ground.

My Reflection

I'm looking forward to beginning the project now it has been approved. I'm excited by the prospect of working with Raspberry Pi as it's a new field of technology for me.

I'm most excited about learning the python language and getting all the components of the Pi working in tandem. With this project, I hope to achieve something that is new and has not yet been put on the marketplace as I will add features to the project that are not used in similar Pi ideas.

Intended Changes

I was planning to add the Pi project into an app but I've decided against this as I feel I might not have the time to create both a working Pi project and the mentioned app. Having now going with the idea of just the Pi project it gives me more time to add more features to the project. I plan on adding a rain sensor so that I can collect rainfall data plus I plan on adding a small solar panel to the project in the hope that I can power the Pi form this source. More features will be added and taken out as the project is in its early stages.

Supervisor Meetings (Lisa Murphy)

Date of Meeting: 11/10/2016

Items Discussed: Lisa and I discussed how we could make the project more appealing and to target a specific audience. As it's a weather analyser we both said we would research it more and see where it can be improved and what audience could benefit most from the idea.

Actions Items: Research how it could be improved upon. Who would benefit most from the project.

8.3 NOVEMBER JOURNAL

Student Name: Peter Barry

Number: X12112631

(BSc in Computing): Internet of Things

Final Year Software Project

Month: November

My Achievements

November was spent finishing the requirements document and starting work on the technical report document. I also did a lot of research into my project. I looked up ideas on how to make the project better by adding new features.

My Reflection

The documentation stage is coming to an end so I'm looking forward to starting work on the coding stage of the project and to begin testing my code. I'm also happy that my documentation has been checked over by my supervisor and she is happy with my work so far. Overall, I'm happy with my contribution so far in regards to the documentation.

Intended Changes

As of yet I haven't added in any changes since my last journal. I think now that I'm getting to the coding stage this will change and be reflected in my next journal.

Supervisor Meetings (Lisa Murphy)

Date of Meeting: 14/11/2016

Items Discussed: Lisa and I discussed how the mid-point presentation. Lisa asked me to prepare my slides for the presentation.

We also went over my requirements document and Lisa listed the areas in the document that I should change. I was also instructed to start work on my prototype for the presentation. I'm required to have some of the sensors reading and displaying the data to a website.

Actions Items: Prepare slides for the presentation. Modify the requirement's document.

8.4 DECEMBER JOURNAL

Student Name: Peter Barry

Number: X12112631

(BSc in Computing): Internet of Things

Final Year Software Project

Month: December

My Achievements

During the month, I worked on my prototype for the mid-point presentation. I had to get everything in place and have a working prototype in place for viewing by my supervisor and a second lecturer. I successfully managed to have everything in working order for the day in question. I also worked on my technical report and have it completed for my presentation.

My Reflection

It was quite a busy month in terms of getting the prototype working, having the technical report to work on and preparing myself for the presentation. I was pleased with the work all complete and uploaded on the correct dates.

Intended Changes

As of yet I haven't added any changes since my last journal. However, I do plan on adding some changes soon. I plan on acquiring some outdoor components for my project so that I can test them against the outdoor elements.

Supervisor Meetings (Lisa Murphy)

Date of Meeting: 01/12/2016, 08/12/2016

Items Discussed: Lisa and I went through a checklist for the presentation plus we also discussed what should be mentioned in the presentation slides.

I also discussed with Lisa what I would be having on display for the presentation day in regards to the prototype.

Actions Items: Complete work on the prototype. Complete technical report and prepare PowerPoint presentation.

8.5 JANUARY JOURNAL

Student Name: Peter Barry

Number: X12112631

(BSc in Computing): Internet of Things

Final Year Software Project

Month: January

My Achievements

I have 3 of the raspberry pi sensors reading weather data so I decided it would be best to begin work on developing the database. Along with this I began working on the website as I only had a dummy website for the sensor readings.

This was the majority of my work over the month as the rest of the month was used for study in preparation for the end of semester exams.

My Reflection

The month was a really busy month as I was juggling between working on the project and preparing for the exams. Overall, I was pleased with the work I was able to do considering it was exams month. I think I have utilized the time well.

Intended Changes

I am thinking on changing the database from MySQL to SQLite3 as I find this to be a better database to work with my project.

Supervisor Meetings (Lisa Murphy)

Date of Meeting:

Items Discussed: There were no supervisor meetings in January as the month was taking up with the exams and the college only started back late in the month.

Actions Items: Complete work on the database. Begin work on the website

8.6 FEBRUARY JOURNAL

Student Name: Peter Barry

Number: X12112631

(BSc in Computing): Internet of Things

Final Year Software Project

Month: February

My Achievements

With all the sensors running I went ahead and built a MySQL database so that I could begin reading the weather data. I am now starting the work on the website and webpages which will start displaying the weather data on high charts. The next phase on the project will be to start work on the webserver.

My Reflection

I found this to be the most productive month thus far. I managed to accomplish all the tasks I set out to do. I hope this sets the tone for the remainder of the project.

Intended Changes

In the previous month I had changed from MySQL to SQLite3 but I've decided to stick with the latter. I have also changed the webserver from LAMP to phpMyAdmin as it works much smoother with MySQL.

Supervisor Meetings (Lisa Murphy)

Date of Meeting: 15/02/17

Items Discussed: I just had the one meeting with my supervisor and we discussed the work on a database, setting up the GitHub account to share with my supervisor and to look at website designs for inspiration.

Actions Items: Begin work on the website and server

8.7 MARCH JOURNAL

Student Name: Peter Barry

Number: X12112631

(BSc in Computing): Internet of Things

Final Year Software Project

Month: March

My Achievements

Completed the frontend of the website plus began work on the final documentation needed. I created several webpages that will be used for the site. I also set up the webserver and tested it with the database.

My Reflection

It was a good month for frontend development as I managed to get all the work done that I set out to do. All the frontend website work was completed to my satisfaction.

Intended Changes

The wind speed sensor and rain gauge will be replaced with different weather sensors as I have been unable to source these components due to the high cost.

Supervisor Meetings (Lisa Murphy)

Date of Meeting: 15/03/17

Items Discussed: Discussed the problem I had with the sensors. The other item was to have the documentation and backend development of the website completed for May.

Actions Items: Website backend development, documentation and presentation ideas.

Raspberry Pi Weather Analyser

Product Design Specification

Version 1.0

12/03/2017

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Peter Barry	03/07/2017	Lisa Murphy	14/04/2017	Draft Document

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to analyse the data readings gathered from a Raspberry Pi and connected sensors. The readings from the sensors will be collected and used to show trends in the current weather elements found in the city of Dublin on a daily basis. The system will generate this data and send it to a local database which then analyses the data and sends it to a website. The system is put in place to display weather patterns in easy to read format.

The document was also used to show updates and changes that were carried out on the project. During the early stages of the application methods were tested but some were not implemented as these methods did not suit the application.

2 General Overview and Design Guidelines/Approach

Quick registration, fast login and a user-friendly interface is the main concept behind this application. Couple that with easy to understand weather charts that links to an MySQL database, displayed through webpages and you have a fast-reliable website that will give the user accurate weather readings. The website was designed with HTML5, CSS3, JavaScript and PHP. When a user registers or logs in his/her details are stored in the database for easy retrieval. This makes for a smoother application.

2.1 Assumptions / Constraints / Standards

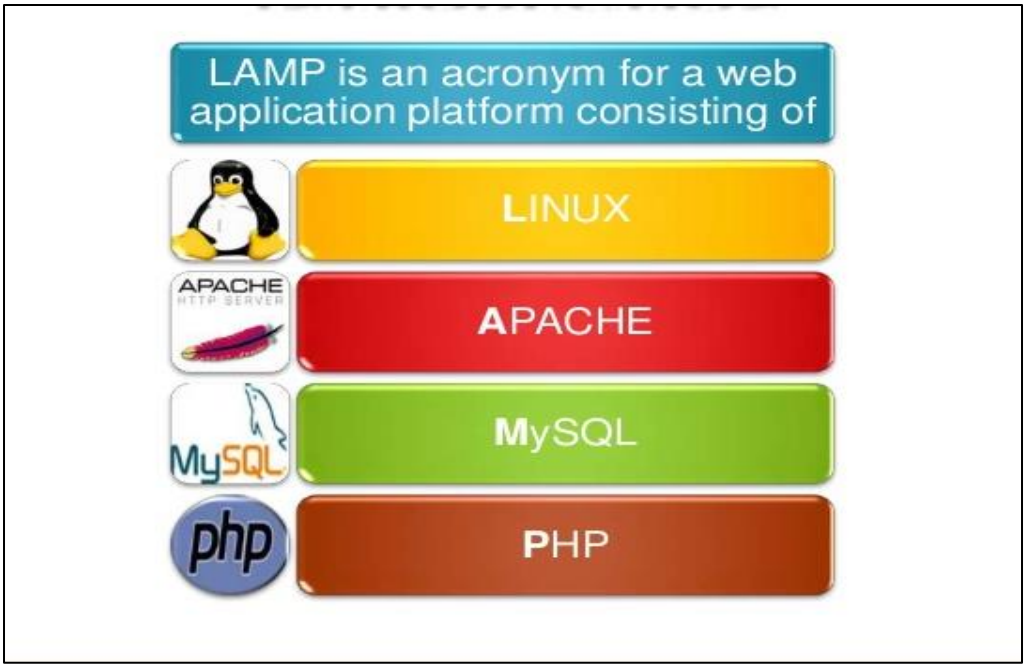
I was working with the latest Raspberry Pi when I first started this project and that model was the RPi 3. As with most technology this has since changed and we now have the RPi 4. This puts a constraint on the application as the technology I am now using is outdated but I will have to make do as the application is now in its final stage.

I also found that I had struggled a lot during the early phase of the project as a majority of the technologies I used were all new to me, so I spent a lot of hours teaching myself these new languages and technologies. Also, I found that the material online wasn't very helpful to me either as it was new territory for most users using the technology. It was a case of trial and error.

3 Architecture Design

The architecture in this application will use the LAMP configuration. The first been the application level. This will consist of a website that displays the weather graphs, signup, login, hyperlinks for the other sensors webpages and logout. PHP will be the server-side language. It adds functionality to the website as HTML can't do this alone. The next level is the webserver which handles the hosting of the website and gives the user the ability to view the website over the internet.

The webserver I will use will be Apache as it's the most reliable and works extremely well with the rest of the technologies used in the application. The last level used in the application is the database server level which will be MySQL. I found this to work well with the Apache webserver and the Linux command line used on the raspberry.



Bringing all these applications together gives us the (Linux, Apache, MySQL, PHP) LAMP configuration.

3.1 Logical View

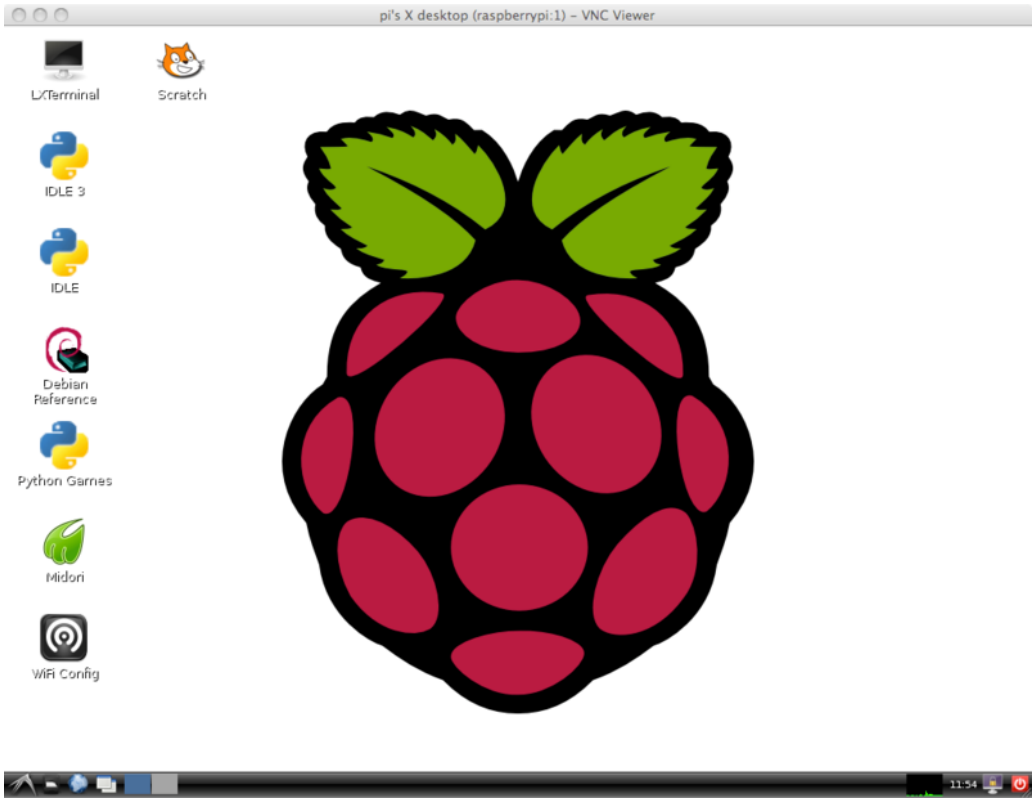
Website Interface

The logical view of the whole application consists of three interfaces.

The first interface is the website and this has all the features needed to run a weather analysis website. We have registration, login, logout and webpage links. This makes up the structure of the website interface.

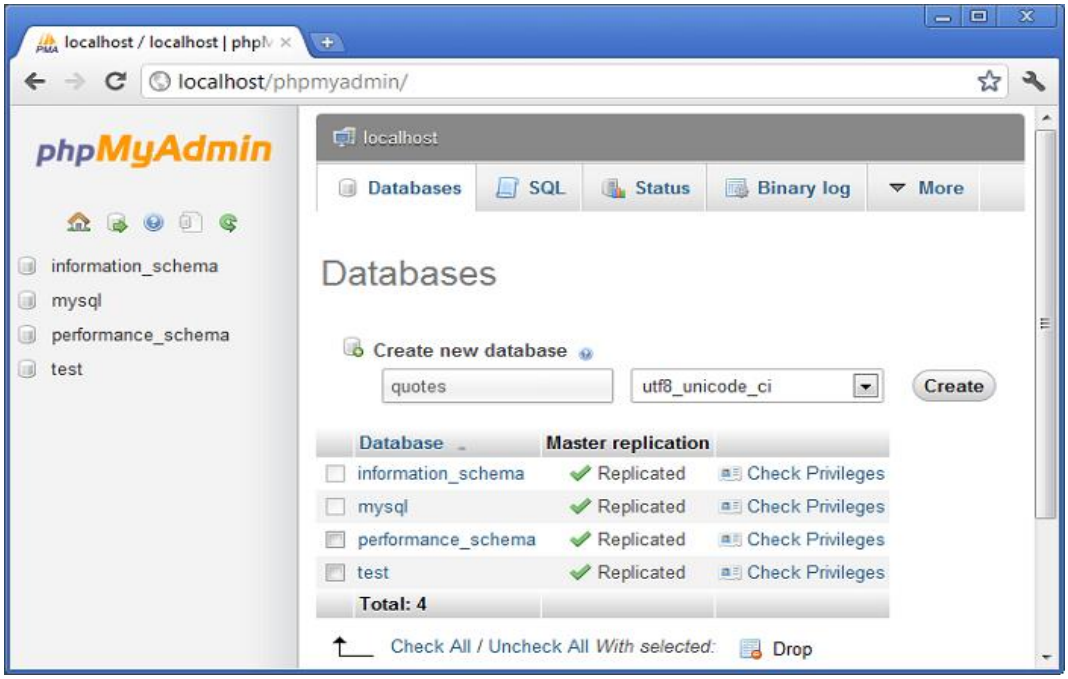
Raspbian OS Interface

Raspbian is the OS used by Raspberry Pi. The interface is used in this application to gain access to the LX Terminal which is the Linux command line used by raspberry. This is essential to the application as all the Python code is run through the Terminal. The Terminal is also used to install all the technologies needed for the application. These been MySQL, Apache and PHP. These needed to be installed on the Raspberry Pi before I could start work on the application.



PHPMyadmin Interface

This is used to control the PHP server-side language. It is also used to control the MySQL database as both run alongside each other in the application. It manages the tables, columns, indexes, relations etc. All the operations used by a database. PHPMyAdmin lets the user display this information on its server.

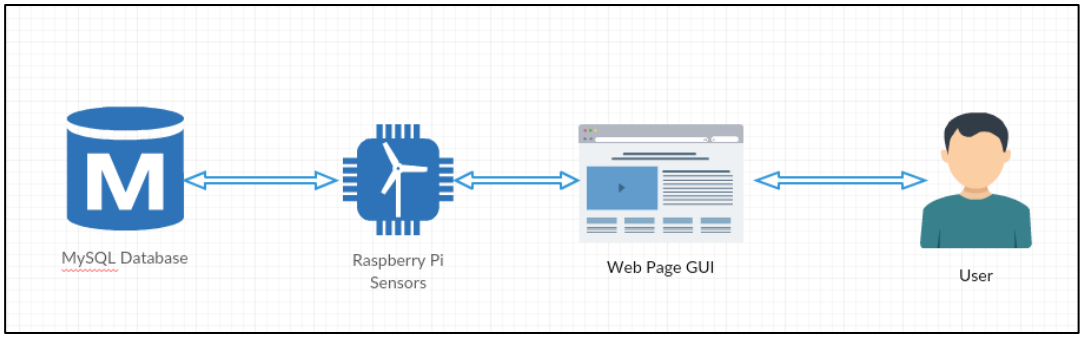


3.2 *Hardware Architecture*

Mapping out the architecture for this application would be to be extremely difficult if not impossible as Raspberry Pi is a fairly new technology so I thought it would be best to include hyperlinks with this section to give a better overview of the architecture. You will find the links in the bibliography section of this document. As we know already from recent mentions that the hardware used in this application was a Raspberry Pi 3 model and GrovePi sensors. These are the only hardware objects used.

3.3 *Software Architecture*

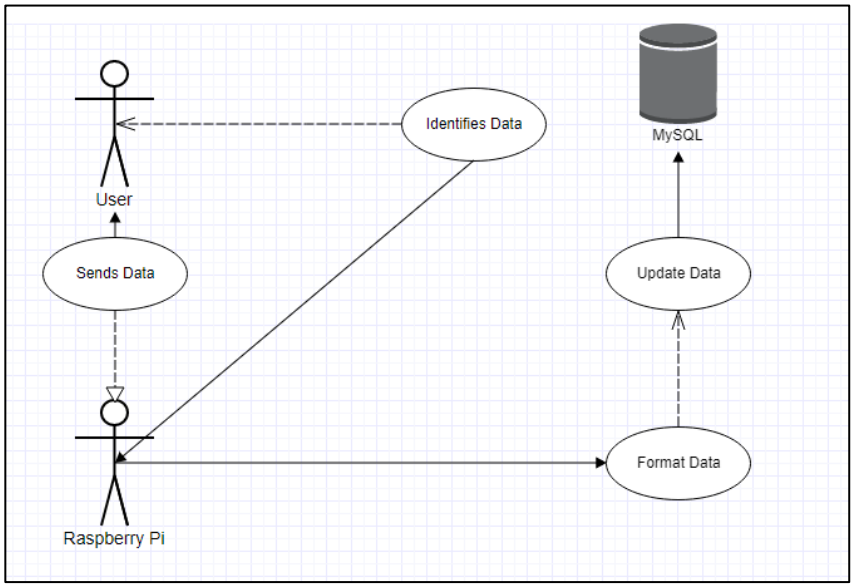
In regards to the software we will used several technologies. These include the Raspbian OS, Linux command line, MySQL, PHP, HTML, JavaScript, CSS3 and Apache. All these software technologies make up the application as a whole. The user logins into the system through a website. The website will display weather graphs that collect readings from the Raspberry Pi weather sensors.



4 System Design

4.1 Use-Cases

The application will give the user all of the CRUD options when he/she are logged into the system. All of the other options have been mentioned in this document.



4.1.1 Scope

The scope for the above use case is to display the process of the sensors which are responsible for reading the elements.

4.1.2 Description

The use case gives a description of the data manipulation between sensors, database and user.

4.1.3 Flow Description

4.1.3.1 Precondition

The user sends commands to the Raspberry Pi

4.1.3.2 Activation

The use case begins when the data starts displaying the sensors readings through the weather graphs.

4.1.4 Main Flow

I. Upon startup in the application the user is presented with the login page of the website.
II. The user provides email address and password to login
III. The user is directed to the homepage of the application.
IV. User has full access to the system.
V. The user can logout of the system.

4.1.5 Alternate Flow

I. The user logs into the Raspbian OS interface.
II. The user wants to add/remove a sensor.
III. The user updates the system.

4.1.6 Termination

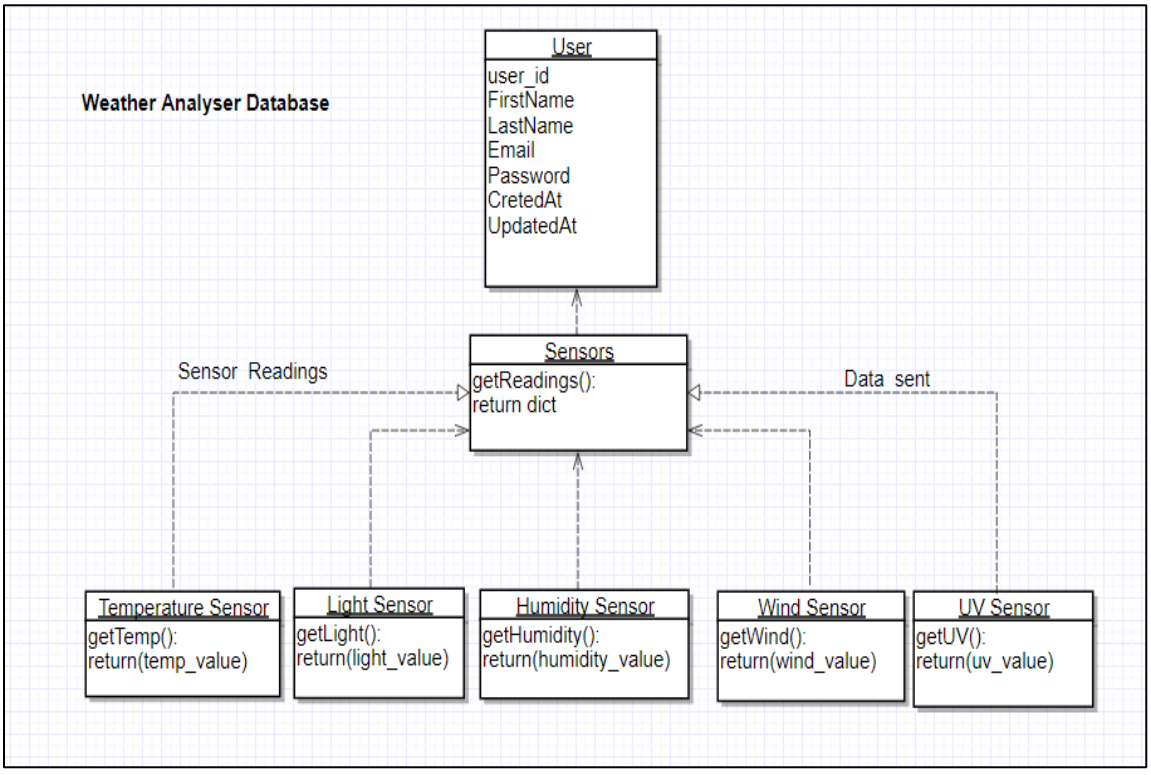
The user can update the sensors readings by saving and running the Python script.

4.1.7 Post Condition

The system confirms the new data through the Python command line by displaying the readings without errors.

4.2 Database Design

The database is the engine of the application. It will be responsible for storing the user's details, the sensor readings and any updates made to the sensors. This is best appreciated through a database diagram which will show the relationships between the entities.

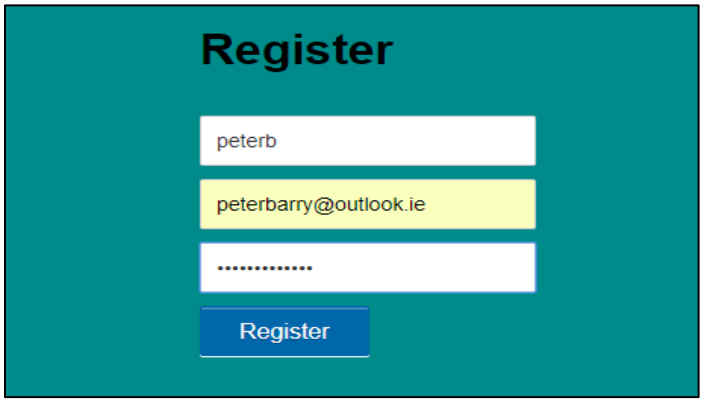


4.3 User Interface Design

For the purpose of this section I will only display four pages from the website as most of the pages look similar.

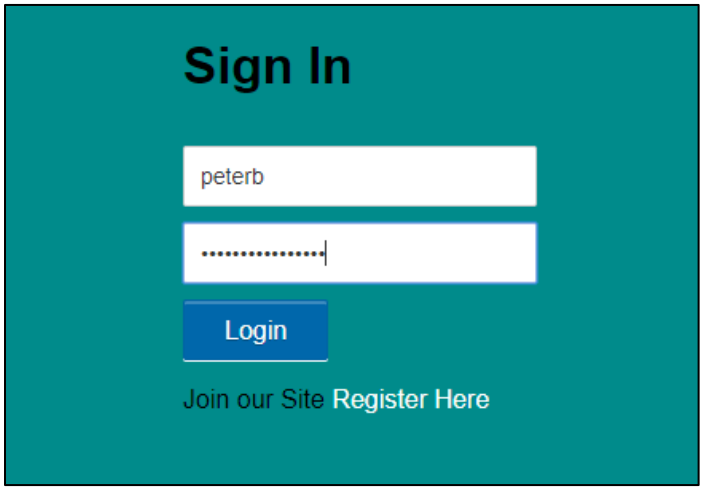
4.3.1 Registration Page

The first page of the user interface will be the registration. The user will be asked to register by entering their username, email and password.



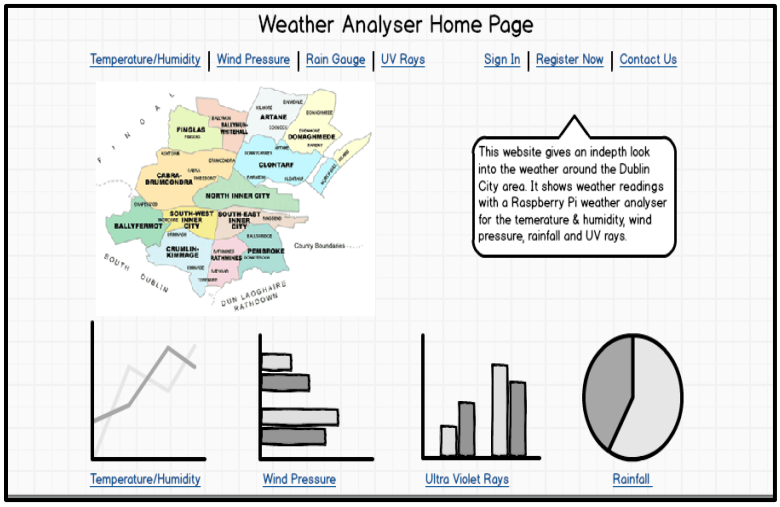
4.3.2 Login Page

The login requires the user to enter their username and password. Here the user will be directed to the homepage.



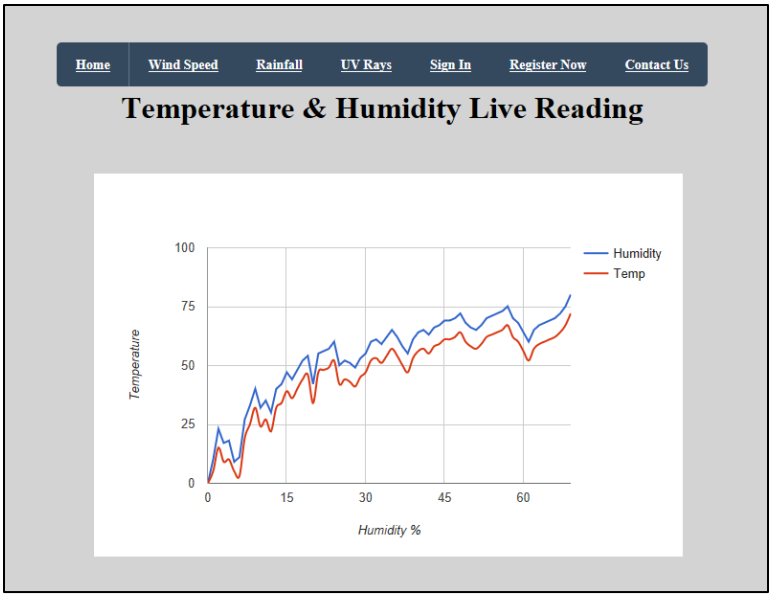
4.3.3 Home Page

Once the user is logged into the home page he/she has full access to the website.



4.3.4 Temperature & Humidity Page

The temperature and humidity page displays a graph with live sensor readings coming in from the Raspberry Pi



5 Bibliographies

http://www.macs.hw.ac.uk/~hwloidl/Courses/F28HS/slides_RPi_arch.pdf

https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-3B-V1.2-Schematics.pdf