# **Introduction.**

This a final year, academic level 8 within Waterford Institute Of Technology a third level University Institution. The award for completion is a Higher Diploma in Computer Science.

## **Acknowledgement**

First and foremost, I would like to thank my Project Supervisor Mr. Richard Lacey who guided me in doing this project. He provided me with invaluable advice and helped me in difficult periods.

His motivation and help contributed tremendously to the successful completion of the project.

Besides, I would like to thank all the lecturers who helped me by giving me advice and providing the equipment which I needed. At last but not in least, I would like to thank everyone who helped and motivated me to work on this project.

## **Background**

The reason I decided to do an IOT project was that at the very start of the pandemic when everyone went into lockdown I was forced t work from my bedroom 8 hours a day working then I had to complete college work for another 2 hours a day, then I would have to sleep in the room for another 8 hours, what I found was that the oxygen levels in the room became very poor and I was actually having trouble breathing and sleeping.

I began explaining this to one of my work colleagues and they said that they had experienced a similar problem and what they did was invest in good quality house plants to clean the air.

The only problem I would have to face is that I would have to maintain these plants which I had very little knowledge about, but following advice from my friend I would learn very quickly.

In order for me to help and maintain these plants I needed to invent some type of smart system that would monitor and feed the plants when I am not around.

The final project for the H.Dip was the perfect opportunity to do this and with this in mind the next step was to research different ideas in order to make a project proposal.

# **Project Research.**

I researched 2 different online projects that had similar problems to my own, each problem was solved by using a raspberry pi but was usually outdoors in one case the author built their own garden.

## **The Raspberry Pi Powered garden.**

This IOT system functions using the following processes:

A Raspberry Pi is used to relay useful information of the garden, such as luminosity, and humidity from various sensors and relay this information into a cloud database.

Once the information is in the cloud, it can be accessed from anywhere using a smartphone app that the author built.

The following are some of the key features of the garden :

* Real-time feedback of the garden's various sensors
* Database of the garden's health status
* Global monitoring and operating capacities

In this project they used Google's Firebase as the intermediary of their IOT system, to create their own free cloud database.

They then used MIT's App Inventor to create a smartphone application which is compatible with the Firebase database and the Raspberry Pi.

It can also communicate with the database with the help of a free Python library.

A picture containing text, ground, outdoor, boat

Description automatically generated

Figure 1 - The Raspberry Pi Powered Garden.

* 1. The Automated Garden System Using A Raspberry PI.

This type of system uses a program called a MudPi.

A MudPI is an open source garden system the author made to manage and maintain garden resources, it is built on a Raspberry Pi.

A Debian operating is loaded onto the raspberry pi and MudPi application is then downloaded, the user can then add specific sensors to specific pins.

Diagram

Description automatically generated

**Figure 2 - Example Circuit Diagram.**

The sensors then relay information back to the user’s phone over the Wifi.

Graphical user interface, text, application

Description automatically generated

**Figure 3 – A MudPi Application**

Upon researching these 2 systems I able to put a project proposal together.

# **Project Proposal**

This project is an application of green technologies for sustainable living. An indoor garden will be created, where plants (Snake Plant, Peace Lilly and Spider) will help clean and recycle the air. The technological solution will measure the oxygen and carbon dioxide levels in the air, and display this using an android application. Building on this idea, other fruit and vegetables will be grown with the aid of robots to assist with irrigation by using thresholds for dryness and wetness.

This project is broken into 2 parts, the hardware, and the software.

The hardware includes different sensors to measure different quantities in the garden then the first piece of software will run on a Raspberry Pi that will interface and read the sensors then a native Android app will be built to monitor and display these values of the garden.

An analysis of green technologies based on IOT solutions will be carried to identify potential

solutions and features for my project. These include:

1. The Raspberry Pi Powered Garden.

2. The Automated Garden System Built Of Raspberry PI For Outdoors or Indoors.

## **Technologies.**

Hardware Requirements

1. A main mother board e.g. (Raspberry Pi, Arduino).

2. Sensors (light sensor, soil/moisture sensor, CO2 sensor).

3. Water pump.

4. LCD screen.

Software Requirements

1. Android Studio (Kotlin or Java).

2. Database (Firebase or MongoDB).

There are many different software lifecycles that could be used in this project, but for the purpose of this document I am going use either Kanban or SCRUM and implement a Trello board to monitor the progress of this project.

The whole idea is that all parts of the project be broken into can smaller tasks where I plan, build, test, and review, then put all finished pieces together at the end to create the finished product.

The software design methodology I will use in this case is the Waterfall method a.

**Figure 4 - Waterfall Method**

Implementation

Ghjgjghj

Planning

Analysis

Design

# **Project Implementation.**

The next step once the project proposal was made and accepted was to think of a way to implement this idea. In Order to track myself I used Trello as a Kanban board to track and keep this project moving, this allowed me to keep track of everything in once place.

The different columns meant the following:

Backlog: This was a backlog of works to be completed based on ideas from project meetings.

Doing: This is works and ideas that I am currently working on.

Testing: This is works and Ideas that I have competed but just putting them through a basic test.

Done: This was works completed and tested.

Weekly Reports: This was just my own weekly log.

Weekly Meetings: This was a photo of the project meetings notes that I had made with my supervisor.

Graphical user interface, application, website

Description automatically generated

**Figure 5 - Trello Board.**

Once I had figured a system to manage this project my next step was to research different technologies that would help me build a similar system to that I had researched.

* 1. Main Controller.

Upon researching and evaluating 4 different single board computer technologies I was decided to use the Raspberry Pi as the main controller.

The reason being was when we compared the Pi to other boards there was a lot more information and sensors available online.

### **Raspberry Pi Vs ODROID XU4.**

I chose the Raspberry Pi in this case, for one reason is for the greater RAM will help run applications faster but the main reason is the Pi has a huge global community which is unmatched.

This means there’s ample information and supports for new users as well as continued development and maintenance of software.

Although the ODROID community is growing fast.

### **Raspberry Pi Vs ASUS Tinker Board.**

### Overall, both the Raspberry Pi 4 and the Asus Tinker Board have strong online communities, and great support, for open-source projects available to try out.

### However, the Tinker Board is from 2017 and definitely shows its age in comparison to the connectivity options of the Pi 4.

### Also the current price of the Tinker board far outweighs my budget compared to the Raspberry Pi which is more affordable in my case.

### **Raspberry Pi Vs Arduino.**

The Raspberry Pi can do everything that an Arduino can do, but it does need a little help in the form of HATs and add on boards, because certain features like analog-to-digital conversion aren’t built in there are a lot more libraries available online and a lot more tools available for the PI compared to the Arduino.

The Arduino is a truly versatile board but the Raspberry Pi is a full computer.

If you need wireless communication, raw processing power and access to the GPIO then the Raspberry Pi gives you all of that in a small package.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Controller** | **Cost** | **RAM** | **GPIO** | **Bluetooth** | **WIFI** |
| Raspberry Pi 4B | $35.00 | 8GB | 40 pin | V5.0 | Wi-Fi 802.11b/g/n |
| ODROID XU4 | $95.00 | 2GB | N/A | N/A | Wi-Fi 802.11b/g/n |
| ASUS Tinker Board | $105.00 | 2GB | 28 pin | BLE | Wi-Fi 802.11b/g/n |
| Arduino | $24.05 | 2K SRAM 1K EEPROM | 20 pin | Add on Board | Add on board |

* 1. **Sensors.**

The sensors I found online include soil moisture, light, air quality , temperature and humidity.

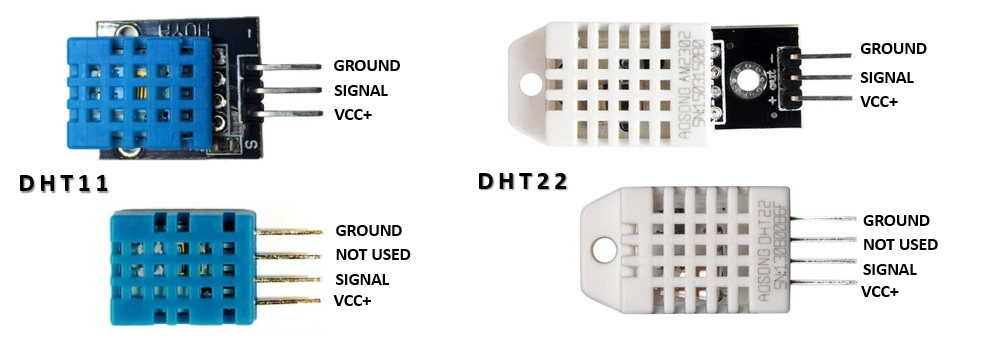
### **4.2.1 Temperature And Humidity.**

When researching temperature and humidity sensors, I came across two different family related sensors the DHT11 and DHT22, I decided to evaluate both.

The benefits of these type of sensors include great long-term stability and low consumption of power.

In addition, you can get relatively high accuracy in measurement at an affordable rate.

Both use the same family of internal chips but only one is more accurate than the other.



**Figure 6 - DHT11 and DHT22**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Temperature Range** | **Temperature Accuracy** | **Humidity Range** | **Humidity Accuracy** | **Cost** |
| DHT11 | -20 to 60℃ | ±2% | 5 to 95% RH | ±5% | $5.90 |
| DHT22 | -40 to 80℃ | ±0.5% | 0 to 100%RH | ±2% | $9.90 |

**Table 1 - DHT11 vs DHT22**

The DHT22 outshines the DHT11 in every aspect from temperature range, temperature accuracy, humidity range to humidity accuracy. The only downside of the DHT22 is, of course, the slightly higher price but you are paying for the better specs.

### **4.2.2 Soil Moisture.**

I evaluated two different soil moisture sensors, but the sensor I’m going to go with is the YL-69 Sensor. The reason being is that both send back an analogue reading for the soil moisture, but because I decided to use the Raspberry Pi it cannot read an analogue measurement.

So the YL-69 has an extra built on board that will allow me to set the sensitivity value once this value has been passed the board will send a logic one to the Raspberry Pi. If I was going to use the Aideepen I would have to use an Arduino or some other Microcontroller that reads analogue signals and then send this signal to the PI over some type of bus or even wirelessly.

|  |  |
| --- | --- |
| **Aideepen** | **YL-69 Sensor** |
| A picture containing text, electronics  Description automatically generated | Soil Moisture Detection Humidity Sensor – Kuongshun Electronic Shop |

**Figure 7 - Soil Moisture Sensors**

### **4.2.3 Water Pump.**

If the idea of this project was to try keeps the plants alive I needed to invest in some type of irrigation system, I investigated multiple different pumps but settled on a single one this was due to budget.

Graphical user interface, application

Description automatically generated

Figure 8 - Multiple Water Pumps.

The following is a 5V water pump that can easily be interfaced to the Raspberry Pi. .



Figure 9 - Single 5V Water Pump

### **4.2.4 Air Quality Sensor.**

The next and last sensor I investigated was an is MQ-135.

The MQ-135 Gas sensor can detect gases such as Ammonia (NH3), Sulphur (S), Benzene (C6H6), CO2, and other harmful gases and smoke. There are other MQ gas sensors in this series, but unlike them this sensor also has a digital and analogue output pin. When the level of these gases go beyond a threshold limit in the air the digital pin goes high. This threshold value can be set by using the on-board potentiometer. The analogue output pin, outputs an analogue voltage which can be used to approximate the level of these gases in the atmosphere.

A picture containing electronics

Description automatically generated

**Figure 10 - MQ-135 Gas Sensor.**

Once I had chosen the sensors I was going to try interface to the raspberry pi the next step was to evaluate some type of database that was going to hold and store data.

## **Database (Firebase vs Mongo)**

There were 2 choices, this was only due to studying them during the course.

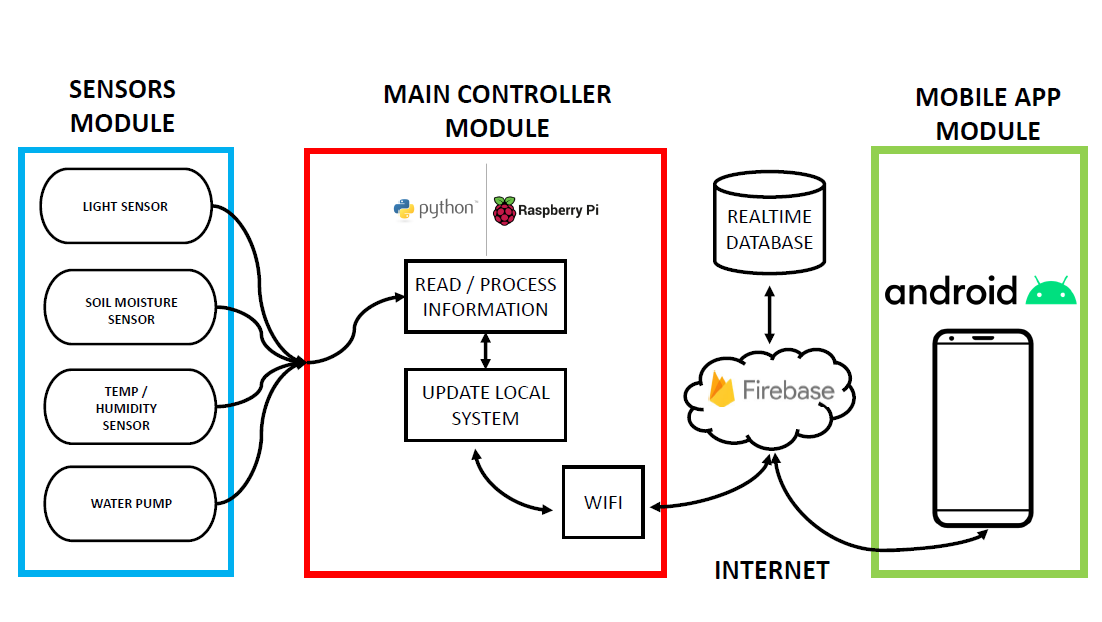
|  |  |  |
| --- | --- | --- |
| 1. **Name** | **Firebase Realtime Database** | **MongoDB** |
| Description | Cloud-hosted Realtime document store. iOS, Android, and JavaScript clients share one Realtime Database instance and automatically receive updates with the newest data. | One of the most popular document stores available both as a fully managed cloud service and for deployment on self-managed infrastructure |
| Primary database model | Document store | Document store |
| SQL | no | Read-only SQL queries via the MongoDB Connector for BI |
| APIs and other access methods | Android  iOS  JavaScript API  RESTful HTTP API | proprietary protocol using JSON |

**Table 2- Evaluation of Databases.**

I have chosen to the firebase Realtime Database only because there are API’s available for the Android operating system, and it will make the project development life cycle a lot more efficient, in the future there could be a possibility to change to a Mongo DB, but for now it will be a Firebase Realtime Database.

### 

The next step of this project was to draw up a system diagram of what I thought the system functioned, this allowed me to be able to explain to others what this project was about.



**Figure 11 - System Architecture Diagram**