

**Smart Eco System - Future of Gardening**

Dorota Marczak, Scott Allan, Nader Sobhi

*CMP311: Professional Project Development & Delivery*

2018/2019



*Please note that Information contained in this document is for educational purposes.*

**Abstract**

**Background to the paper (brief and client) and aim of project**

With the advent of new technologies, which have revolutionized countless areas of human life, many employers realized that multiple activities may now be automated, or even performed remotely at relatively low costs. Amid trending tech solutions such as 3D printing and wireless conference rooms there is one inconspicuous, yet very powerful tool - the Internet of Things. IoT is a concept which emerged barely a decade ago and has turned out to be a blessing especially for small companies, which constantly compete for shares on the market while having a limited budget and human resources. IoT technology can be used for a variety of simple activities performed with users consent and according to their instructions such as Amazon’s Alexa smart speaker, a staple in many homes today.

Dr Lynsay Shepherd, representing Abertay Plant Systems, decided to take advantage of the opportunities created and ask the local tech start-up - DNS Team – to design and develop a plant monitoring system utilizing the above-mentioned technology, as well as similar web-based solutions. The client stressed that the company expects a device that could be used by anyone; not only to appeal to their existing customer base but also to “tech obsessed Millennials who care about the environment but don’t keep any plants in their home”, as Dr. Shepherd put it. The idea to merge contemporary technology with the age-old art of plant keeping was perceived by DNS Team as a unique challenge.

The aim of the project, settled on after closer familiarization with the provided brief, was to provide an easy to use system that required no expert knowledge to operate and maintain. Ideally, a ‘plug and play’ device would be developed that would begin operation as soon as the end user registered the device on the companion website. Once registered, the device would start to collect data which would then be displayed via the website, as well as via a built in LCD screen. The user could then use this data to ascertain if their plant was growing in optimal conditions.

**Methodology**

In order to deliver a high-quality product within the agreed timescale, development of the hardware system and the website was conducted concurrently, which minimized the risk of a potential delay. Team members were assigned to certain tasks according to their strengths, which facilitated breaking the work down using the Sprint methodology. Following the Scrum methodology allowed the team to set clear goals over the course of the development phase, meaning the team leader could track the project progress over time. Regular meetings (see *minutes*) were held to ensure the project proceeded as planned, the workload was distributed fairly and finally, that all team members have a chance to air any problems they may have faced with a certain task.

The final product consists of three main elements: Arduino board with connected peripherals, a database which stores the sensor readings, as well as the website that provides an overview of the well-being of the plant. The physical device processes the data received from the attached sensors and sends them in JSON format to the database via Wi-Fi connection, provided by the WEMOS adapter. The data is held in normalized form and then accessed by the website, that utilizes it to create intuitive data visualizations.

**Outcome**

The professional attitude of each team member to their entrusted tasks resulted in delivering high-quality product a few weeks before settled deadline, what may turn out to be highly beneficial for client’s business and its growth. Smart Eco System, a user-friendly and visually attractive interface providing reliable information about a plants condition, has a chance to become a bestseller in the gardening market, which has not fully experienced a technological revolution yet. This fact, in turn, may attract the attention of gardeners devoted to their plants and therefore create a model for smart gardening devices.

Contents

[1 Introduction 5](#_Toc495312033)

[1.1 Background 5](#_Toc495312034)

[1.2 Aim 5](#_Toc495312035)

[2 Procedure 6](#_Toc495312036)

[2.1 Overview of Procedure 6](#_Toc495312037)

[2.2 Procedure part 1 6](#_Toc495312038)

[3 Results 7](#_Toc495312039)

[3.1 Results for part 1 7](#_Toc495312040)

[4 Discussion 8](#_Toc495312041)

[4.1 General Discussion 8](#_Toc495312042)

[4.2 Countermeasures (for a project in ethical hacking) 8](#_Toc495312043)

[4.3 Conclusions 8](#_Toc495312044)

[4.4 Future Work 8](#_Toc495312045)

[4.5 Call to action 8](#_Toc495312046)

[References 9](#_Toc495312047)

[Appendices 11](#_Toc495312048)

[Appendix A - Example 11](#_Toc495312049)

[Appendix B - Example 11](#_Toc495312050)

[Appendix C - Suggestions for formatting figures/tables/screenshots in the body of the text 11](#_Toc495312051)

[Appendix D - Deliverables & requirements 13](#_Toc495312052)

[Appendix E - Minutes 15](#_Toc495312053)

[Appendix F – Other 16](#_Toc495312054)

# Introduction

## Background

While it is true that you can happily grow your plants with a packet of seeds, a sunny patch of land or a simple flower pot filled with soil and a watering can, there are much better and faster ways to create your dream garden. Less than a decade ago, modern technology crashed into the gardening sector that was experiencing developmental stagnation, interrupted from time to time by new biochemical solutions. Despite almost unlimited opportunities brought by technological revolutions, many gardeners were still confining themselves to obsolete cultivation techniques, originating from the first half of the twentieth century and some even from ancient times. In many cases, the reason behind it is not related to the potential costs, but rather the habit of following traditions and not departing from the things we already know. However, with the advent of GMOs, people began to realised that it is possible to take control over their plants fate which, until recently, remained almost entirely dependent on weather conditions. That, in turn, switched on the green light for a whole range of modern solutions, including intelligent automatic watering and remote growth optimization based on cause and effect analysis. It must be noted, that it was Millennials with their concept of a need to be the mother of invention that have led the revolution to success. Instead of relying on a gut feeling and mercurial weather conditions, they prefer to influence nature by utilizing in-depth exploration of a plants environmental preferences and delivering the required mineral resources based on extensive analysis.

According to the Washington Post, Millennials began paying attention to the plants they place in their urban flats that not only serve as a live decoration or a reference to a favourite aesthetic pattern but are also a way of expressing the unique personality of their owner. As ubiquitous social media encourages the younger generation to share every aspect of their life online, modern society is under pressure to be perfect in every aspect of contemporary life. However, the first steps in gardening often turn out to be harder than expected. They browse the Internet to find the tips for geeks, but then forget to water their plants regularly and end up trying their luck again. Finally, they reach for gardening applications as a last resort in their gardening adventure. Unfortunately, instead of fully-fledged systems they find crawling applications with basic advice and watering reminders being the extent of their possibilities. Having no choice, they install one of them on their phone and begin anew. This time they do not forget about basic needs of their plants and that is usually enough to keep them alive for a few weeks. Proud of themselves, they enlarge their little collection with a few more specimens, which turn out to be even more demanding. Irritated by imprecise advice available on potted plant labels or seed packaging, such as ‘place in partial shade’ or ‘requires medium temperature environment’, as well as hard to estimate irrigation needs, they look for a tool that would facilitate taking care of their flowers. They are eager to pay for a solution to their problem, but unfortunately none of the available products fulfil their needs, either due to being too expensive or requiring specialised knowledge to construct.

## Aim

The fundamental high-level aim of the Smart Eco System project was creating a product that would meet the needs of modern gardeners; both digitalized Millennials just discovering the world of plants, as well as older members of society experienced in the field, but wary of new technologies. In practice, the project aim is to deliver a device that would collect environmental data on a plant and place the readings into a database. This data would then be used by a web interface to power intuitive and visually-attractive graphs. The complete system itself, being the outcome of the project, will give the user the means to monitor their plants as precisely and meticulously as if they were placed in a scientific laboratory. For an experienced gardener, such an opportunity could become the beginning of plant growth optimization, which could potentially save them time and money.

Although the main goal of the project was to design and build the Smart Eco System along with a user interface, there were several other objectives identified. Firstly, it is expected that the system will provide a set of features, such as an accessible and responsive website, that will adapt to different screen sizes. The system should be easy to maintain, as basic training provided to the members of staff will not cover the full technical specification and the staff themselves may not have technical expertise. Furthermore, it is crucial that the application is user-friendly and visually attractive, otherwise it may dissuade potential customers from purchasing the product. Another important factor is safety – malicious attacks may cause interruptions in service and these, in turn, may lead to claims of a lesser product if they occur frequently. By treating sensitive data with particular care, the company protects itself against credential leaks, which in turn ensures that they will not have to pay out of pocket to compensate for any damages they may have caused. This would also prevent the tarnishing of the image of a reliable brand.

Regarding the physical device, a vital feature must be reliability combined with a smooth flow of data, as the company cannot afford releasing a poor-quality product. Reducing possible data loss is a minimum requirement. Last but not least, collected data must be stored in a manner this is compliant with the relevant laws in operation and its integrity should be preserved throughout the system’s life-cycle.

# Procedure

## Overview of Procedure

According to Duncan Haughey, author of Project Smart platform, the key to a successful project is detailed planning. That is also why the next step, taken after setting the goals of the project and identification of the stakeholders, was dividing the development stage into sections corresponding to the deliverables. Each of them consisted of a number of features, which had been established based on an extensive list of functional and non-functional requirements. Before any development work began, the organization of the database had to be completed in a way that would enable reception of data from the physical device, as well as to allow the website to read and process the data. The team then proceeded to the prototyping stage, where each part of the system was analysed in terms of both its internal operations, as well as cooperating with other elements in the set.

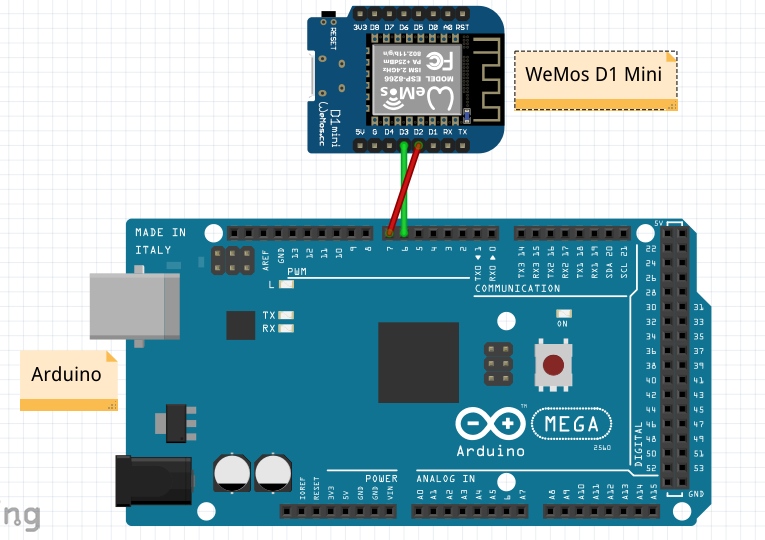
A vital unit considered during the analysis was to find a way to implement a connection to the WEMOS board from the Arduino board. Damage to one of the above could result in the potential loss of the ability to transmit data, which in turn would lead to the failure of the whole system. To avoid this situation, the team worked towards the optimization of the wiring, ensuring connections were secure. Meanwhile, the sub-team allocated to the development of the online platform performed broad research in order to gain inspiration based on professional, functional and visually attractive services available on the internet. They then utilized the ideas to prepare paper prototypes, which allowed them to achieve as close a match as possible to the client’s requirements. Once this was completed, the developers could estimate the server space required to host the website, so web hosting could be purchased. When both teams prepared a graphical visualization of the part of the system they were allocated to, the team leader called a general meeting in order to discuss the solutions and make corrections wherever necessary. The implementation process ran smoothly and without any major time delays, however it required some inner rotations due to certain unfavourable circumstances. The team members worked collaboratively on the features identified as more complex and time consuming.

## Procedure part 1 – arduino

The hardware device was developed incrementally to ensure that each stage was successful and to allow time for appropriate testing and modification as required. By building the device in small stages, it allowed the team to track progress as well as served as a troubleshooting guide should an error occur. For example, if the device was behaving normally before a sensor was added, only to misbehave upon integration, it would seem the cause of the error lay with the incorrect installation of the new sensor.

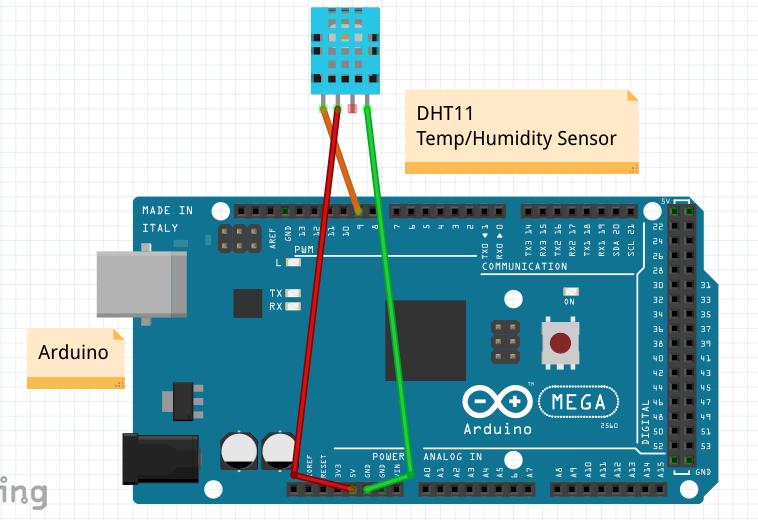
This section will look at how each stage was carried out and how the overall solution was reached. Connection diagrams will be provided for each stage, with a complete diagram incorporating the whole hardware system included at the end of the section.

**Connect Arduino and WEMOS**

****

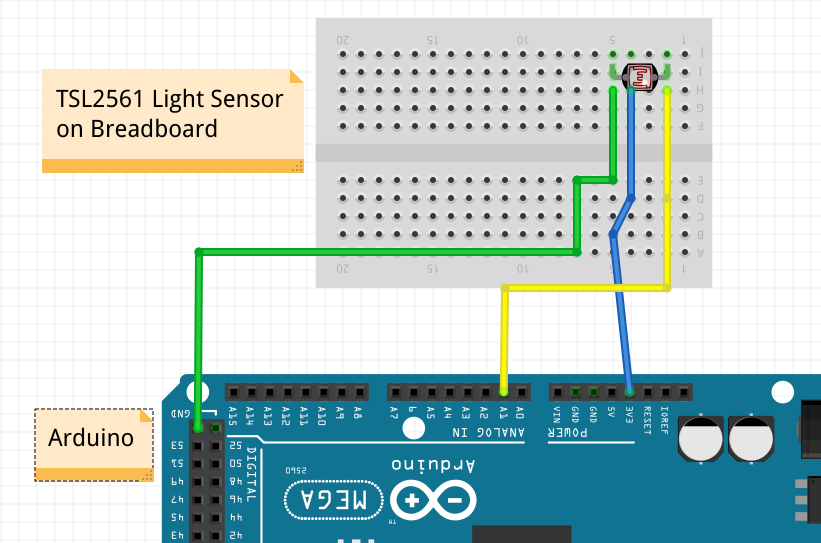
Before we could gather the required data from the sensors to send to the database, we had to give the Arduino the capability to connect to the Internet. This was achieved using a WEMOS D1 Mini board. The WEMOS is powered by a micro USB connection and uses two pins to send and receive data to and from the Arduino, making it relatively simple to connect and setup. The code running on the WEMOS board consists of establishing a connection with the database, listening for the data collected by the sensors via the Arduino, placing the sensor values within a JSON string and finally inserting the sensor values in the appropriate table. Once a connection with the database has been established, it will construct and send a JSON string every thirty minutes. Before any sensors were connected, placeholder data was used to ensure that a WiFi connection could be set up between the device and the database server, as well as to ensure that the data was being stored in the database correctly. Once this stage was complete, the next step was to connect a sensor and obtain useful data.

**Connect Temperature / Humidity Sensor**

****

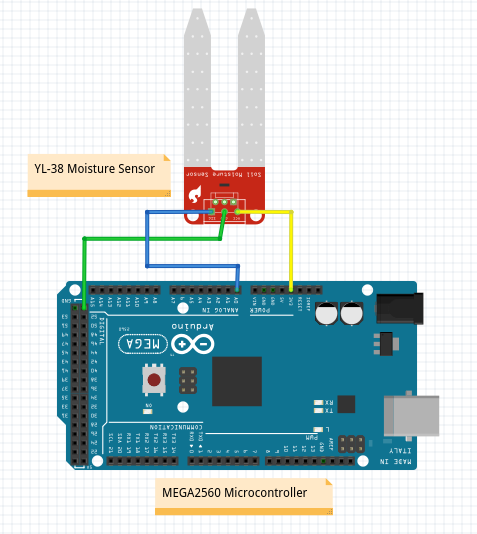
The DHT11 sensor was selected as it enabled the team to take readings on the temperature and humidity in one unit. This sensor requires three pins to operate. It is powered via a 5V pin, requires use of a ground (GND) pin and transfers data via a digital pin. A DHT library is also required for successful operation and enables the use of functions specific to the DHT11. The dht.begin() function starts up the sensor, whilst the dht.readHumidity() and dht.readTemperature() functions translate the data received from the sensor into a human readable format. These values are read as floats before being converted to String objects for use with the JSON string.

**Connect Light Sensor**

****

The TSL2561 was chosen for use as the light sensor as it is very simple and provides easy to use data. The sensor reads the amount of light falling on it in a given environment and outputs this data on a scale from 0-1000. This allowed the team to save the data values as an integer and display it to the user as a percentage with 0% meaning fully dark and 100% fully bright. The sensor requires three pins to operate: 3.3V to supply power, GND to ground the circuit and an analogue pin to provide the data, in this case the A1 pin. A breadboard had to be used as the sensor is very basic and is not attached to a board to provide the pin out functionality. The breadboard can easily be replaced with female-to-male connection wires for a full production model, but the breadboard was kept in place as it allowed the team to daisy chain power connections for other sensors.

**Connect Moisture Sensor**

****

The YL-38 moisture sensor was chosen as it is simple to connect to the system and provides data that is easy to interpret. The sensor is powered from a 3.3V pin, grounded via a GND pin and sends data through an analogue pin, in this instance A0. The sensor works by reading the amount of electrical resistance in the circuit. The two prongs are connected to the power pin and the input pin and the difference in the returned value is used to calculate how much moisture is present in the given medium. Should the sensor be placed in wet soil, a connection between the two prongs will be established and the value returned will be lower than if the sensor is placed in dry soil where the current finds it harder to travel through the medium. Using this data, we can inform the user whether their plant requires watering or not by using a simple if-else statement to determine where the returned value falls on a scale, as well as collect the specific data value and use it to track how the soil dries out over the course of a day or week.

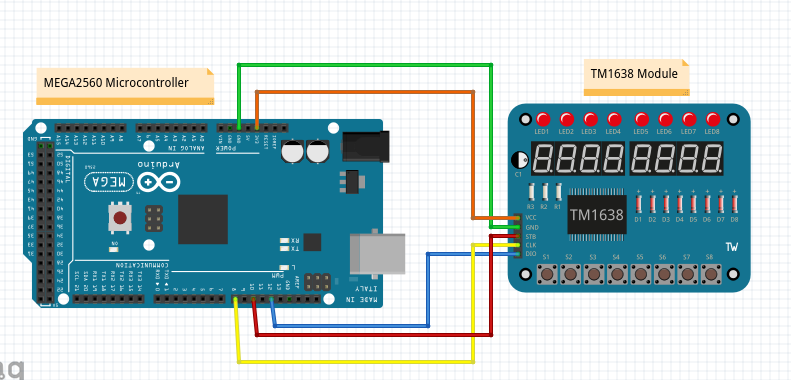
**Take Readings & Place in JSON String**

Once all the sensors were connected and providing useful data to the Arduino/WEMOS sub-system, the next step was to place these data readings in a database. There was a number of options the team could have chosen, but the decided method was to insert all the data values into a JSON string and store this string within a database. The advantage of this method was that the database table only required four columns: one which held the unique identifier, in this case the MAC address of the WEMOS, a column to hold a timestamp of the reading, and a column which contained the JSON string itself. The fourth column was an autoincrementing integer required for SQL operations. The team could then parse this JSON string using PHP and use the data to power the graphs on the website. The disadvantage of this method is that the JSON string won’t be sent to the database until it contains four separate readings [one for each sensor], meaning that database entries could be missed if a sensor became inoperative. Therefore, the advantages outweighed the disadvantage, which was deemed unlikely to occur unless the device or sensors were deliberately tampered with.

****

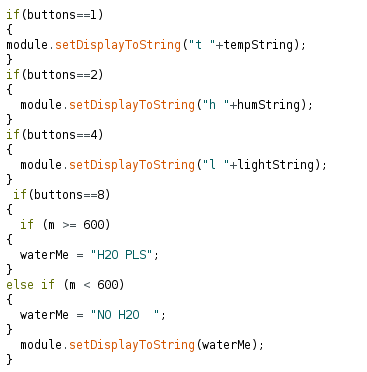
**Send json string to dB**

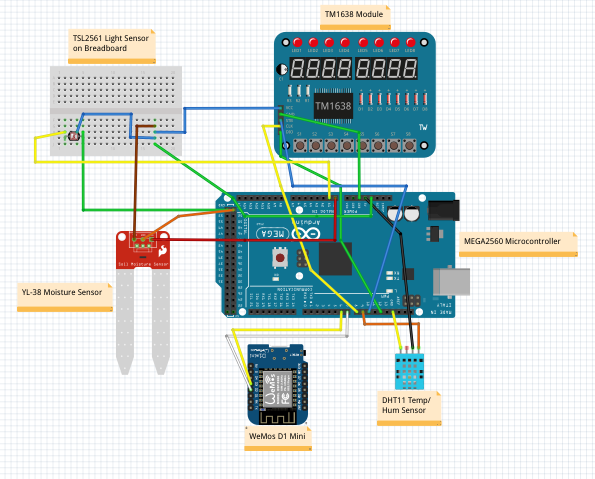
The WEMOS is used to construct the JSON string before transferring it over WiFi to the database. The Arduino collects all the sensor data and passes it onto the WEMOS via a digital pin. The WEMOS listens for the sensor values and once it has collected all four required readings, a JSON string is created with the values being appended in the appropriate location. The WEMOS then constructs a MySQL query consisting of the MAC address of the WEMOS which is used as a unique identifier, the current time and the JSON string itself. Once the MySQL query has been executed the JSON buffer is cleared so that it can be used to construct the next JSON string. A delay of thirty minutes is set after which the process begins anew.



**Connect LED display**

Whilst the data collected from the sensors has more use in powering graphs in place on the website, a simple display was implemented so that the user can see the real time data collected from the sensors. A TM1638 module was selected as it contains eight seven-segment displays that can be used to display basic letters and numbers and eight buttons which can be used to make selections. A TM1638 library was required to access the functionality of the buttons, via the module.getButtons() function, and allow data output to the seven-segment displays, via the module.setDisplayToString() function. The team decided to use the first four buttons to switch between which sensor reading was currently displayed, with the default display and button showing the temperature reading. The second button was set to display the humidity reading, third to display the light level and the fourth to display whether the plant required watering or not. First all of the sensor values had to be converted to String objects as this is what the seven-segment displays required. A series of if statements was then used to interpret which button was activated and the appropriate reading would be displayed to the user [see below code snippet]. The module also benefits from requiring only 3.3V to run, reducing the overall power draw of the system. More complex displays could have been used, but the data displayed would remain much the same but the power draw of the device would be increased.



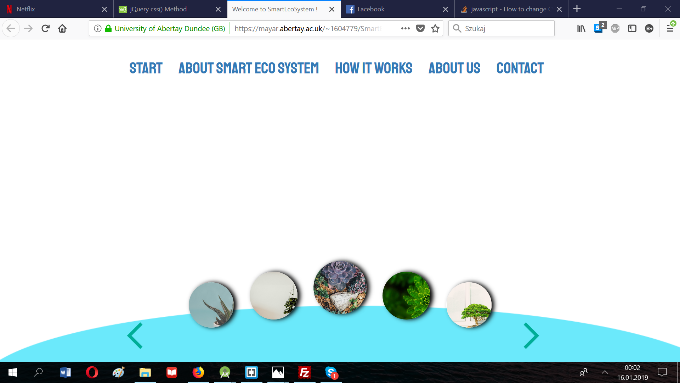


**Complete system circuit diagram**

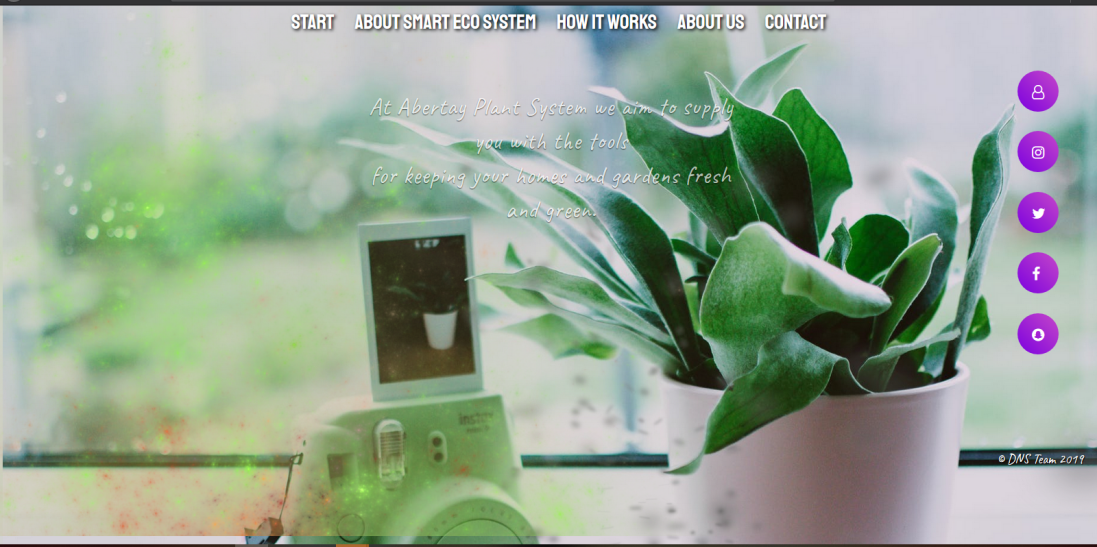
## Procedure part 2 – WEBSITE

Designing an online platform, which would provide the means to monitor the plants status, began with establishing a list of functional and non-functional requirements that it was expected to fulfil. First and foremost, the website aimed at the presentation of the collected readings in a graphical manner, which could serve as a base to perform precise growth analysis. That, in turn, may significantly facilitate optimization of plants health due to how important a role the environmental conditions play in the gardening world. Also, the platform is required to provide a way to register the device and the plant being monitored. By purchasing the product, the user obtains access to the activation code assigned to the physical device. Logging into the system and adding the device is all it takes to begin utilising the system. Aside from the aforementioned features, the requirements regarding the Smart Eco Systems website included accessibility (as a basic human right by virtue of UN Convention on the Rights of Persons with Disabilities), as well as responsiveness, protection against malicious attacks and a visually attractive, user-friendly interface.

Regarding the complexity of Smart Eco System, as well as the differences between the presentation and functional part of the website, it was decided that the most favourable solution would be splitting the platform into pre-login and post-login sections. By separating these logically independent parts, the company could make the most of the marketing opportunities created. Therefore, the front page, which is to some extent a business card of Abertay Plant Systems, was expected to provide basic information about the company and its innovative product. In order to organize the development process, all the requirements were converted into an extensive list of features, which were then allocated to sprints. Next, the first prototypes of both parts of the website were designed. Out of six front page versions designed by the sub team responsible for web development, the project manager decided to choose and implement three, which were considered as outstanding in terms of visual attractiveness and functional solutions. Although each of them made a good impression on paper, two of the implemented prototypes turned out to be potentially difficult to maintain and characterized by low responsiveness (Fig.x, Fig.a).

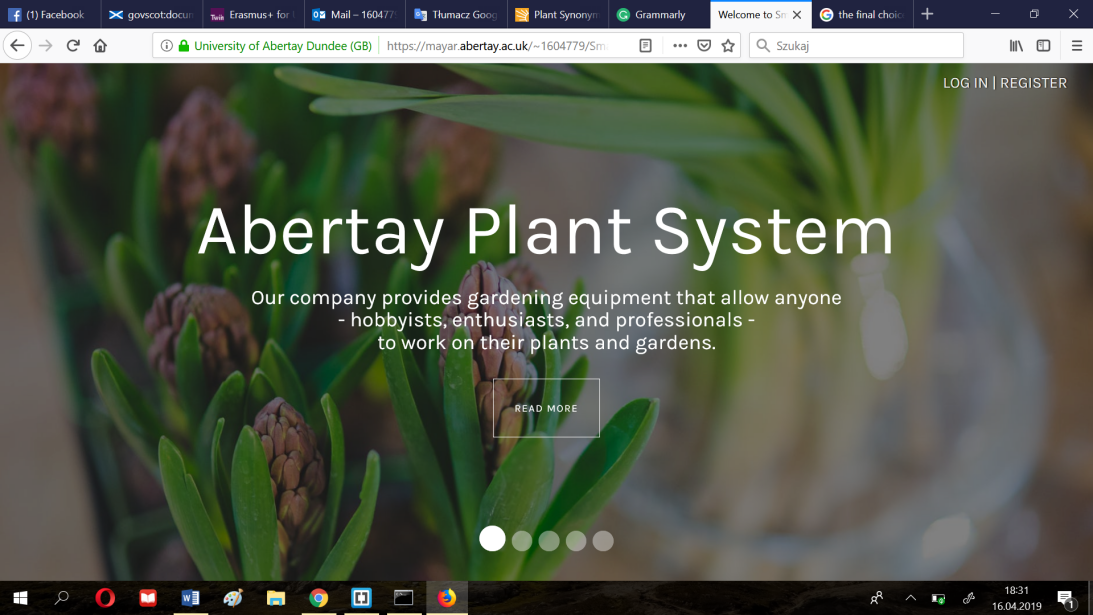


**Fig. x Prototype rejected due to scalability problems of one of the features**



**Fig. a Prototype rejected due to low level of responsiveness on mobile devices**

Therefore, the prototype chosen combined the simplicity of expression with visually attractive, minimalistic layout adapted to different screen sizes. Having added a stylish font and high-quality picture to the background, the prototype gained the desired look and could be extended with further elements.



**Fig. y Leading prototype of the front website**

The development process began by enriching the application with the sliding mechanism using Bootstrap Carousel, which added dynamism to the minimalist design (Fig.a).



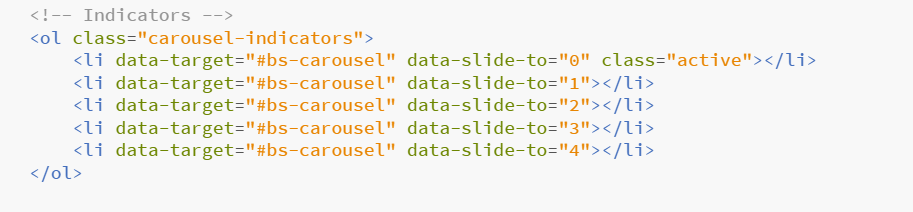
**Fig.c Carousel mechanism (JavaScript)**

The content of the landing page has been separated into five slides, which aimed at creating the effect of a digital brochure, directing the user to the most important section – contacting the producer (Fig.c).



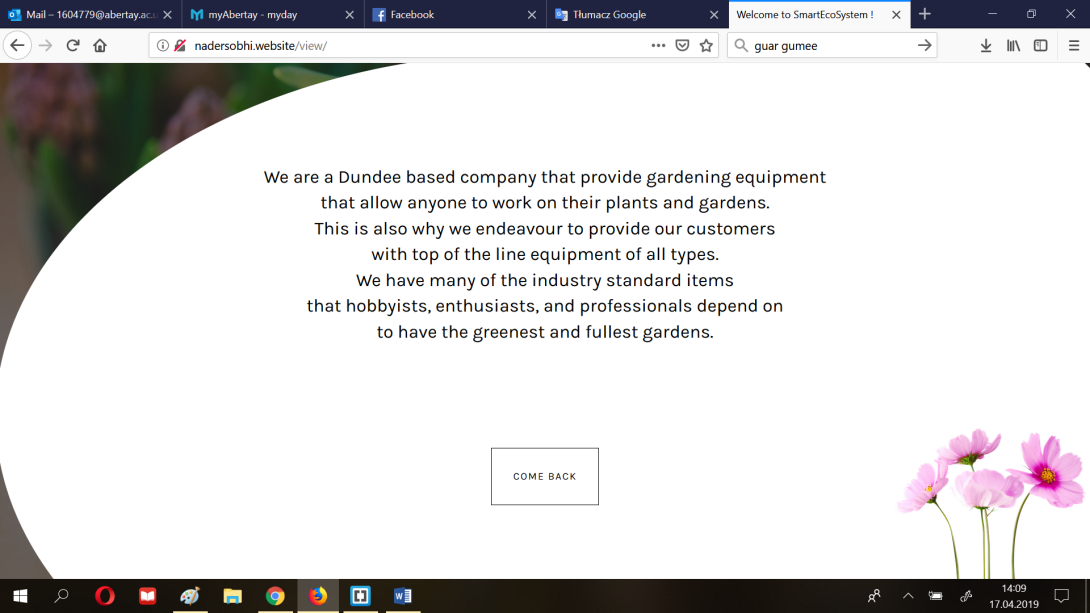
**Fig.c Construction of the carousel**

In order to allow the user to freely move between the slides, the team augmented the website with corresponding buttons, adapted both to desktop and mobile version (Fig.s). Next, the team decided to display the content allocated to each slide in an overlay in a way that would avoid fully covering of the main page, giving an impression of fluidity and lightness of the design.



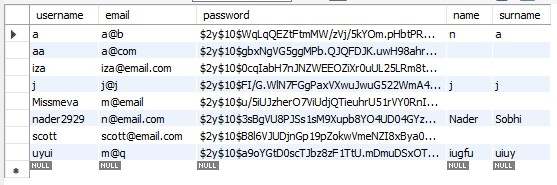
**Fig.s Slide indicators**

Having implemented the additional layer, as well as filling in the white space with the proper content, a return option and a decorative element, the platform finally gained a professional character and following this, fulfilled its informative role (Fig.d).



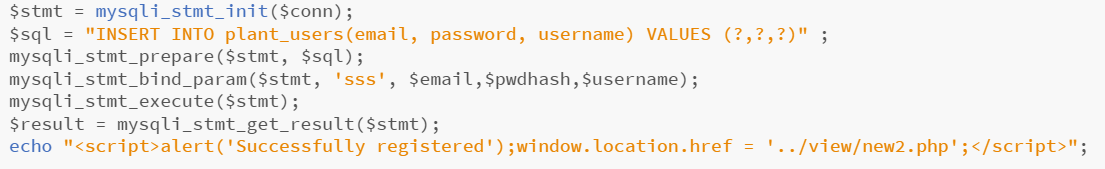
**Fig.s Example content of the ‘pop-up window’**

Prior to moving to the implementation of the login & register facility, the team established a user table in a database, which would correspond to data required in the form. It consisted of a username and hashed password, the elements required for user authorization, as well as additional data such as email and full name.



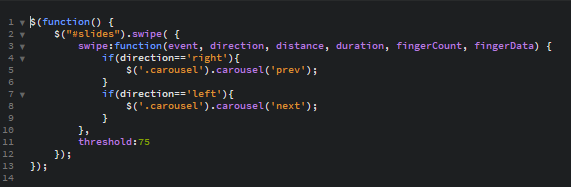
**Fig.d User table (filled with test data)**

Regarding the registration form, it was implemented by combining front end and back-end elements responsible for placing the data in the database. Once submitted, the data is sent with the POST method and is processed by the API with additional precautions, including parameterization of SQL queries (Fig.e).



**Fig.e Inserting the user into the database (API)**

Though the main functionality of the landing page was implemented, one of the requirements indicated that the website be navigable using mobile touch rather than buttons. After broad research it was decided that swipe functions available from the jQuery library would be the simplest solution amid all available frameworks and libraries available on the Internet (Fig.f).



**Fig.f JQuery function for swiping slides on touch (mobile devices)**

The implementation of the members area of the website began with establishing a responsive, collapsible menu, which would allow the user to move to the desired section of the website with a single click on one of the available subsections. Moreover, the menu was designed to highlight the subsection currently in focus, as the usability of the platform was one of the main requirements (Fig.r).



**Fig.r JQuery function for detecting the active element (subsection)**

## Procedure part 3 - Database

Storing the data that is produced and used by the system is one of the main foundations that the application is built on. This is due to the fact that the application relies heavily on the ability to store information for later use, the storage of data will mainly be performed by the Arduino System and the retrieval will be performed exclusively by the website. The format in which the data is stored is a combination of a Relational Database (RD) Model and a NoSQL model. The reason for this being that while some of the data will be highly structured the rest will be of varying size and types (i.e. integers, text, real numbers). In order to simulate the potential conditions that will be in place once the system goes into production it was decided to use a platform that could host the potentially considerable amount of data that will be collected. The platform that was deemed most appropriate was Amazon Web Services (AWS) which allows its users to lease all types of servers and server space ranging from simple RDs to servers setup to execute machine learning code. In our case a simple server setup to host a RD was all that was necessary. The setup process followed to lease the server space, create the database, and create the tables will be shown and described below.

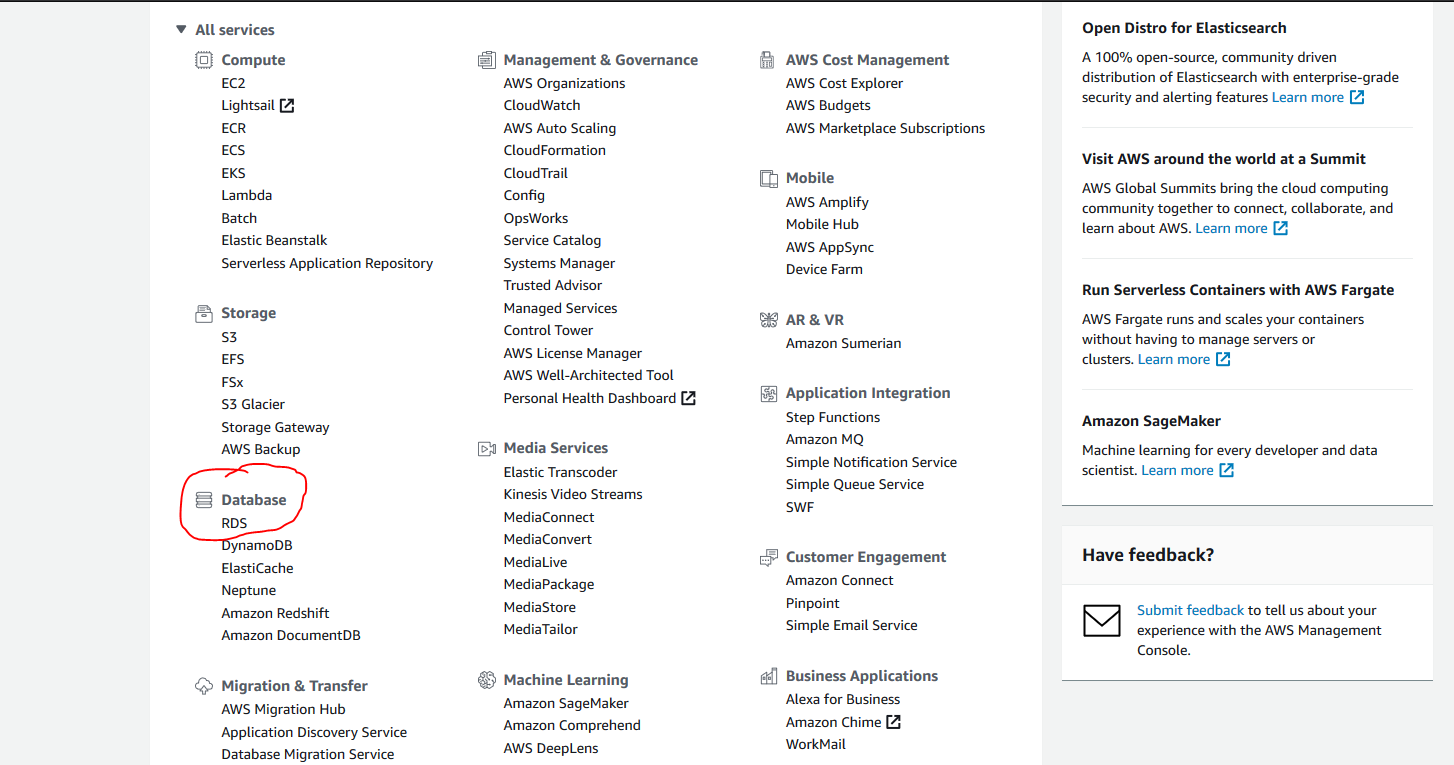


Figure X1

As seen in Figure X1 the user will be presented with this page once they sign up or sign in to AWS, from there they can select the type of service they want to use. In the case of this system an RDS instance was created. Following this the developer can then select the “Create Database” option, shown below in Figure X2, and follow the instructions displayed by the Setup Wizard.

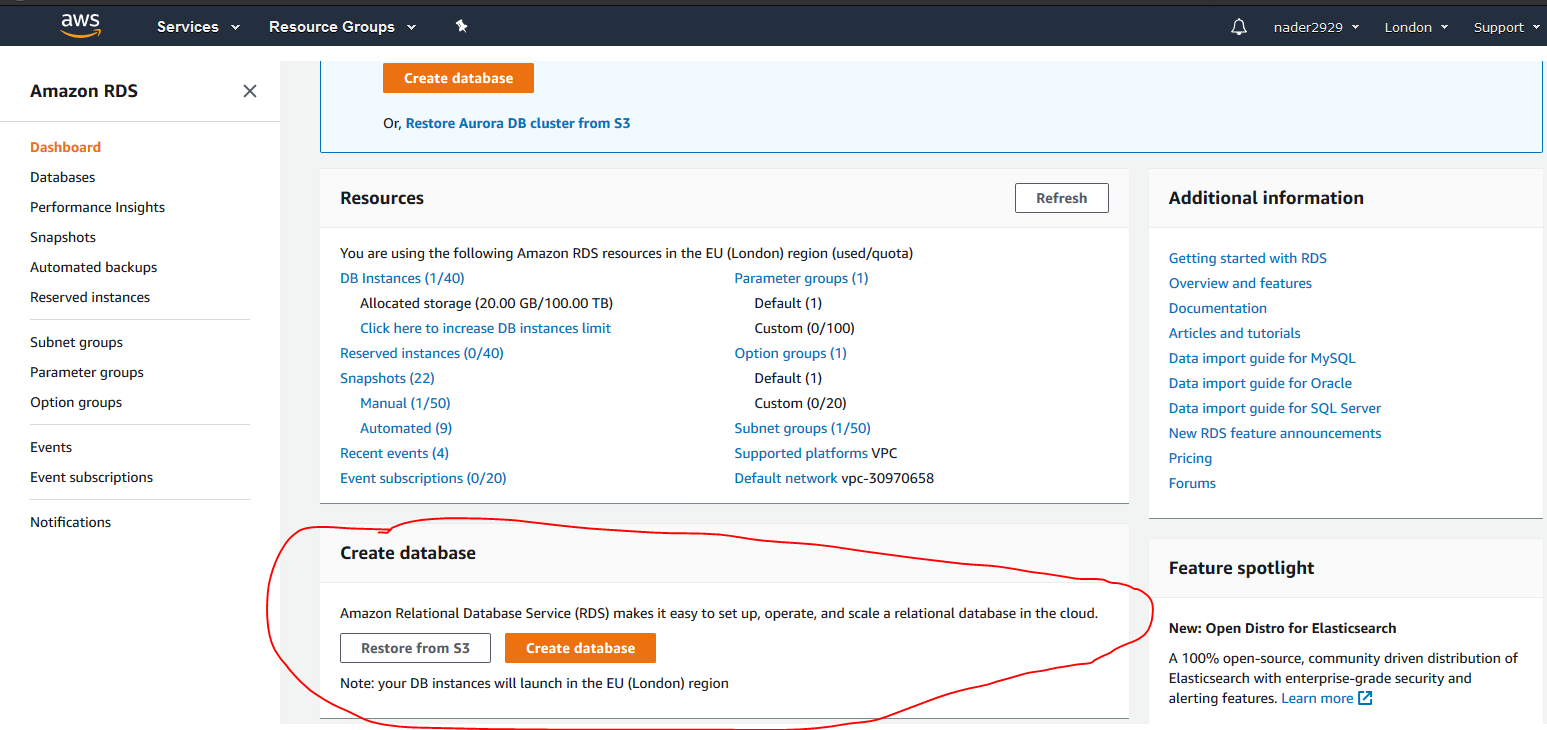


Figure X2

Once the setup was complete the user could select the instance they create and change any settings they wish using the panel provided, Figure X3 showcases this panel.

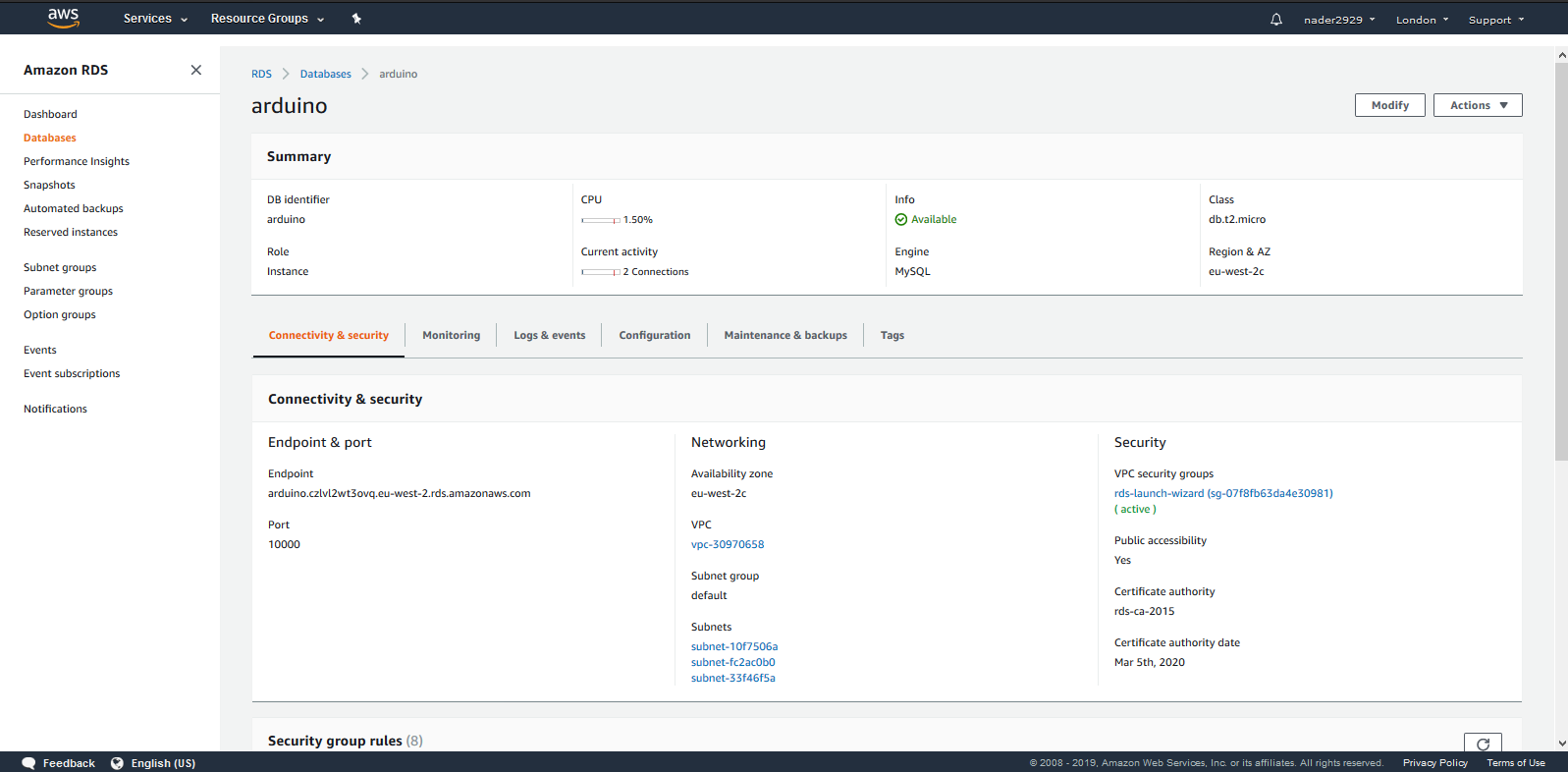


Figure X3

From there the user could use their choice of Database Management Software to create the tables required. In the case of this system the developer decided to use MySQL Workbench to implement the design. MySQL Workbench allows the developer to write SQL code that will perform the desired actions. The first table that needed to be created was responsible for holding the readings recorded by the Arduino devices. The code used to achieve this is shown below in Figure X4.

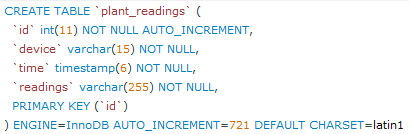


Figure X4

The table was then populated by the Arduino once it was operating correctly and could be used in the future by the website to present the readings in a graphical manner. The ‘readings’ column of this table is where the NoSQL aspect is evident. Due to the fact that the readings produced from the devices could in the future be extended, i.e. more sensors could be added to the device, this would not affect the database. Therefore no changes will need to be made to the database.

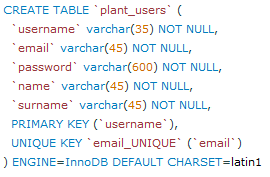


Figure X5

In Figure X5 the code used to create the user table is shown. Once this table was created and the code that was responsible for populating the table was functional, users could easily login and register on the website.

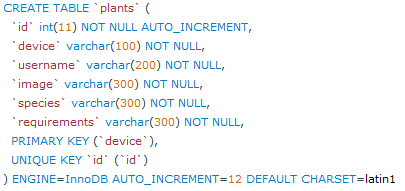


Figure X6

In Figure X6 the code used to create the plants table, that would link the users to their device. It would then be populated by the users once they add their devices to their accounts.

# Results

## Results for part 1

In this section, the results of the above outlined procedures will be discussed. The results will be discussed in three individual sections, with a brief conclusion of the overall result of the project closing the section.

**Results for Arduino**

The manner in which the hardware system was put together allowed for results to be gained incrementally, i.e. once one peripheral had been attached, it was tested to ensure it was sending the correct data and then the next peripheral was attached and so on.

**Arduino and WEMOS**

The WEMOS unit was the first peripheral to be attached to the Arduino unit because ensuring a stable WiFi connection was imperative to the success of the project.

To test that the system worked, the Arduino was given a string that it passed to the WEMOS. If the WEMOS received the string, it would display it in the Serial Monitor of the Arduino IDE. Once successful, the destination of the string was then changed to the database to test the WiFi connection and whether the data was held correctly in the database.

This result was important as it gave the team a solid platform to build upon.

**Arduino and Sensors**

The sensors selected for the project were chosen not only for their ease of use regarding system implementation, but also on how they returned data to the system. For example, the DHT11 sensor was used to collect temperature and humidity readings. Not only did this combine two sensors in one but also returned readings in terms of degrees Celsius which meant that accurate results were being obtained as soon as the sensor was connected properly. This scenario was similar for the other sensors chosen in the project.

Testing of the sensors followed the same process as testing the WEMOS connection. The sensors would return values and these would be displayed in the Serial Monitor. This allowed the team to fine tune how they wanted the data to be displayed as well as test that it was held correctly in the database.

**Arduino and JSON String**

The team decided that instead of storing the sensor readings in separate columns within the database, a JSON string would be used to collate the data and be held in one column of the database. The thinking behind this was based around how efficient the system could be at sending data as well as making it easier to parse this data for use on the website. Rather than sending four separate pieces of data to the database all at slightly different times, a JSON string is created that takes all four data ‘chunks’ and appends them sequentially. Once all data chunks have been received, the JSON string is then sent to and held in the database. This means the website only has to access three columns in the table (the ID of the hardware system, the time of entry and the JSON string) and can then parse out the required data from the JSON string where needed. This allows all four sensor readings to be sent at the same time, removing any discrepancies in the times of the collected data.

**Arduino and LED Display**

Whilst it was accepted that the main use of the data collected was to power the creation of graphs and charts on the website, the team decided that being able to view real-time data via a hardware display might also be useful. In order to achieve this, the TM1638 module was selected because it features a display and switch buttons on the same unit.

The display itself is made up of eight 7-segment LEDs, a rather simple form of display that was perfect for displaying numbers, which would mainly be the datatype on show. However, the display can have problems displaying certain letters of the alphabet, ‘M’ and ‘W’ being prime examples. Four readings had to be displayed on the screen, namely temperature, humidity, light level and moisture level. To ensure that the user knows which sensor data is being displayed, the first LED displays a lowercase letter relating to the sensor type; ‘t’ for temperature, ‘h’ for humidity and ‘l’ for light. Displaying the moisture level was more problematic, in that while the moisture level could be displayed, it could not be labelled similar to the previous displays. To overcome this, the code was modified slightly so that the display reads ‘H20 PLS’ if the plant requires watering or ‘NO H20’ if watering isn’t required. Whilst the display isn’t uniform in how it communicates the real time data to the user, the display device has been kept as the TM1638 module as it is simple to use, displays data accurately and draws very low power compared to more complex display modules.

**Results for Website**

The website is the main interface that the user will have to interact with in order to fully utilise the system, so it was necessary to thoroughly test the website to ensure that it was functional, easy to use and secure.

The facets of the system that were tested are going to cover multiple aspects, mainly the functionality of the website and mobile testing. More info on each aspect, as well as their results, will be given below. The overall website will also be evaluated for its browser compatibility as well as its speed.

The tests for this aspect of the system were performed by anonymous users as well as by the developers, and the feedback produced was used to make modifications where appropriate.

**Login and Register**

**Functionality Testing**

This is the testing aspect that will affect users the most as it is a gage of whether or not users can perform the actions they need using the platform created.

After experimenting with the website, the testers indicated that while the website works as it should, there is no need for storing the name and surname of the person registering. Upon receiving this feedback, the developers removed those input fields from the form used for registering. This also meant that the application would be handling a smaller amount of sensitive data and therefore worries about security breaches are reduced.

**Mobile Testing**

With around 50% of the world’s internet traffic being served to mobile devices it was vital to ensure that users could easily use the web application from mobile devices.

Once the login and register functions where tested on mobile it was deemed that they implemented in such a way that was suitable for mobile devices, as shown in the Figures below.

A picture containing electronics

Description generated with high confidenceA picture containing electronics

Description generated with very high confidence

**Front Page Navigation**

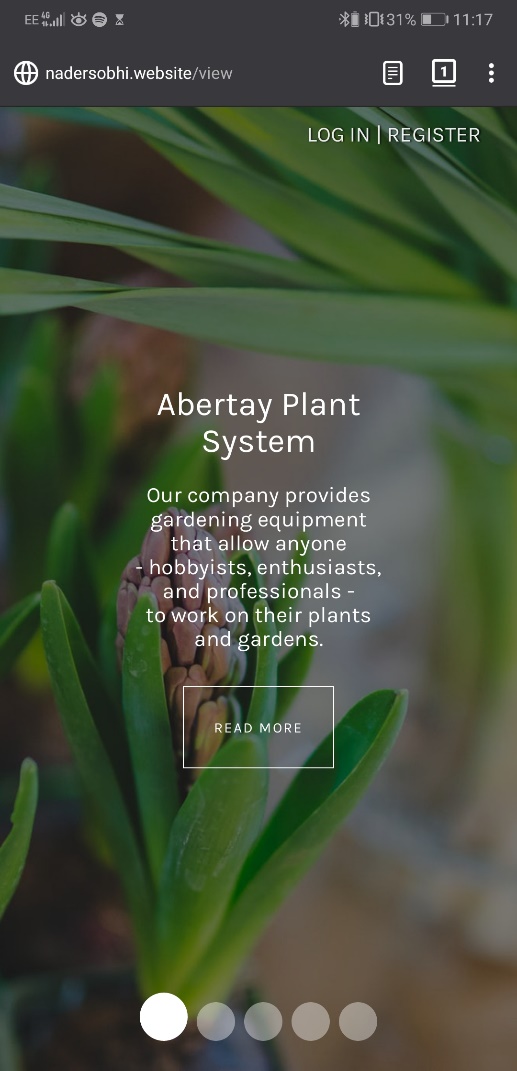
The page that greets the user upon accessing the website had to be functional as well as attractive to use.

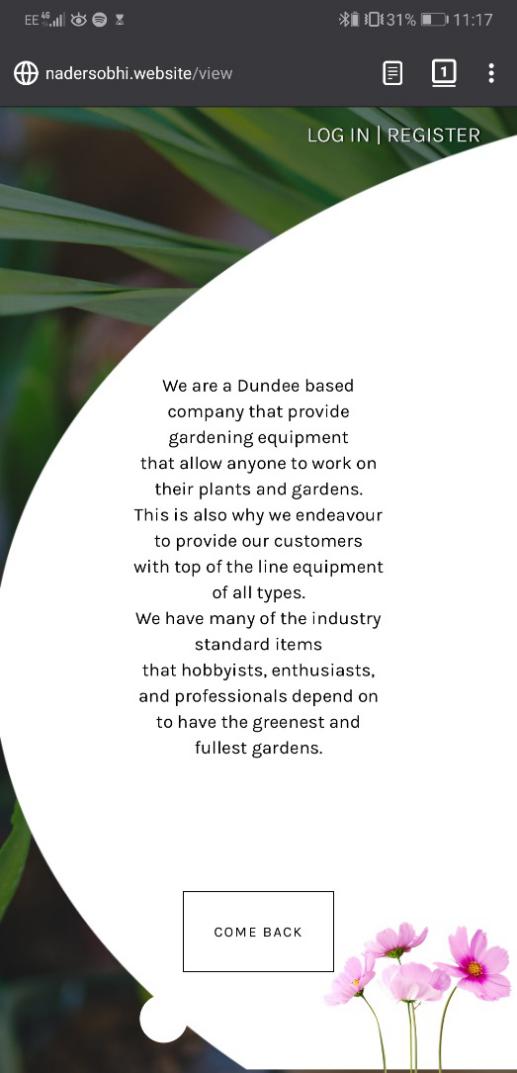
**Functionality Testing**

After allowing the testers to use the website’s landing page and try to casually browse the information presented in order to get a feel for the application, some feedback was generated. Testers stated that they found the site easy to use and browse. Testers also noted that they really enjoyed the smoothness with which the front page allows the users to navigate between the different slides.

**Mobile Testing**

To ascertain whether or not users could easily navigate the web app on mobile devices they were asked to try and perform the same action they had done on the desktop version of the website on a mobile device. Once they had done so the developers received feedback, it was determined that the testers found the front page a delight to navigate as it felt like an app more than a website. This was encouraging to the developers as that was the goal they set out to achieve when creating the web A picture containing fruit

Description generated with high confidenceapp. Example images of what the website looks like on mobile are shown below.



**Plants Panel Navigation**

Once the user has logged in they will be redirected to a panel that will allow them to view the readings that their device(s) has collected. They will also be able to add more devices should they need to.

**Functionality Testing**

The main feature that was tested was the navigation between the different sections and graphs, as well as the functionality of the following features: Adding devices and plants to the users’ accounts, Switching the graphs view from a Day view to a Month or Week view, and using the back to top button. Once the testers tried the features and casually browsed the page to get a feel for the ease of navigation of the webpage, they gave their feedback to the developers. The overall feedback was that the website was easy to browse, and the features were intuitive to use.

**Mobile Testing**

Making this panel functional and usable on mobile was vital as it would allow users to check the status of their plants using their mobile device. The feedback received was that while the website is functional and enjoyable to use on mobile devices, the font size on the graphs is rather small. This was noted down but not changed due to the fact that changing this might negatively affect the desktop version of the website. Examples of what the Plant Panel page looks like on mobile are shown below.

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidence

**Overall website tests**

The website was then tested for its overall browser compatibility and speed. The reason these attributes were separated from the features tested above was that these attributes contribute to the overall feel of the website and not specific to any page or feature.

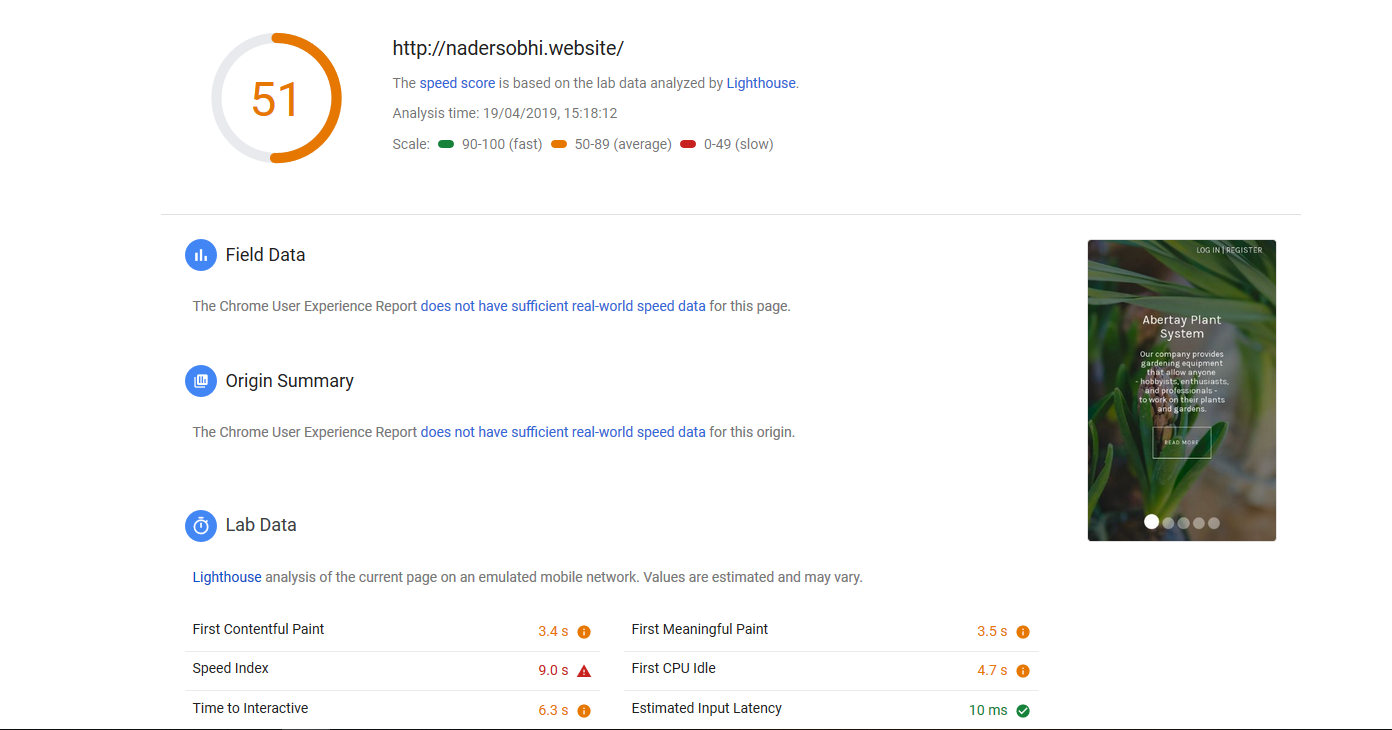
**Browser Compatibility**

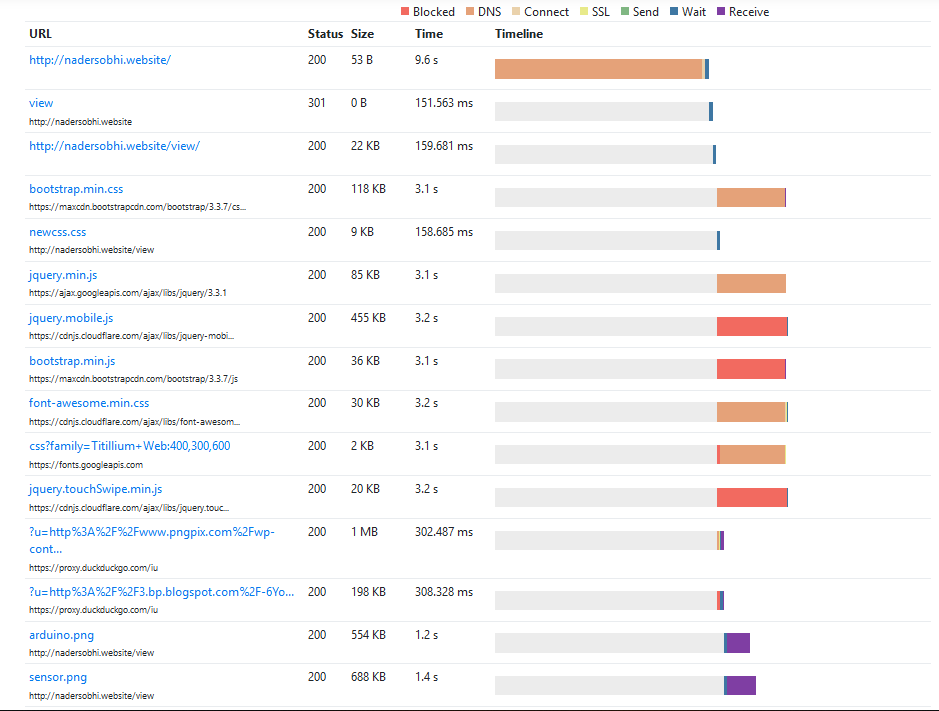
In order to test the website on multiple browsers in a timely and efficient manner the developers used an online service provided by [browsershots.org](http://www.browsershots.org). This service allows the developer to view screenshots of any website they choose on 95 different browsers. These screenshots are shown in the next page. As seen below, all modern browsers and a lot of the older versions of popular browsers seem to function well.



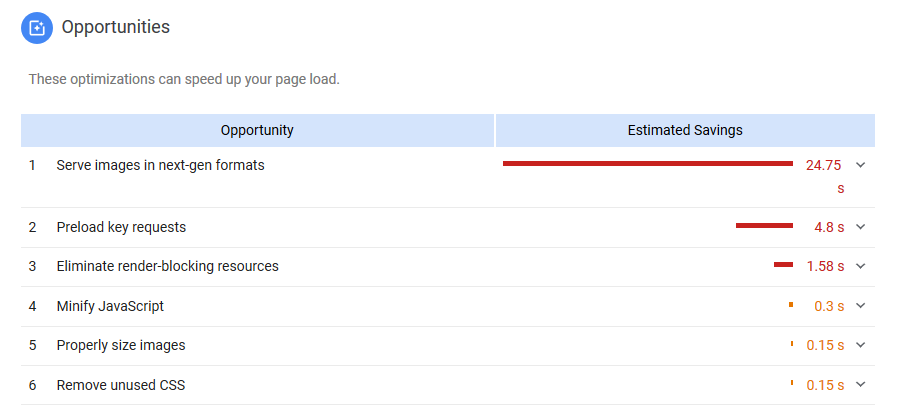
**Speed**

Another measure of responsiveness is the speed with which the website loads and responds to the users clicks and touches. There are a number of facilities online that allow a developer to view their websites performance, for the purposes of this website Google PageSpeed Insights and KeyCDN Speed Test were used. The results of these test are shown below.





Upon analysis of the results it is clear that the website is not performing optimally in terms of speed. This is mainly due to the images used on the website which use old, outdated formats, as shown below.



Unfortunately, these images are vital to the visuals of the website and therefore cannot be changed, a potential future change would be to convert these images to newer, more efficient formats.

**Results for database**

The database used to hold the data acts as a type of bridge between the website and the device so naturally the developers needed to ensure that it was working correctly and securely. Luckily, the testing and the subsequent results for the database are minimal as all that needs to be ensured is that the database cannot be accessed from the backend i.e. an unauthorized user being able to login to the database using a Database Management System. The other factor that could be tested was the prevention of SQL injection type attacks although this was mainly mitigated on the website and not directly through the database. In this section the backend security of database will discussed and shown.

The actions that the developers took was to set a unique username and password for the database that differed vastly from the default options, as well as switching the port that the database operates under from the default 3306 to port 10000. These changes are shown below in Figure Z1 with the password being redacted by the setup screen of MySQL Workbench.

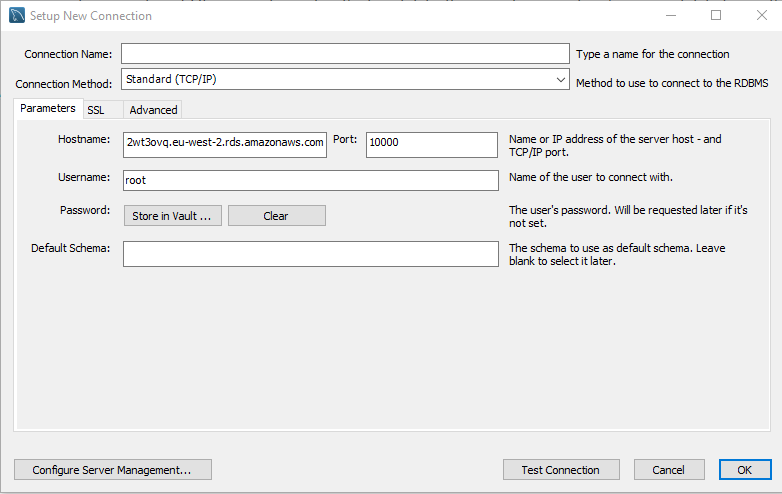


Figure Z1

In order to showcase the databases ability to hold data in an organized manner screenshots of some of the data held in each table will be shown below.

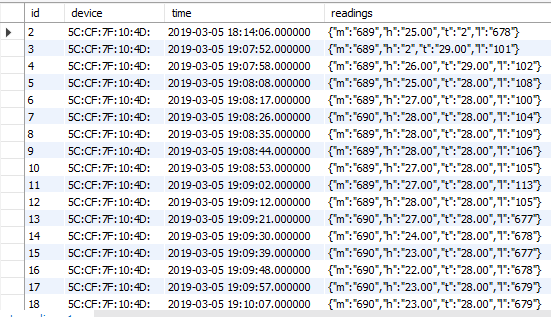


Figure Z2

Figure Z2 shows the *plant\_readings* table columns as well as some of the records held in the table.

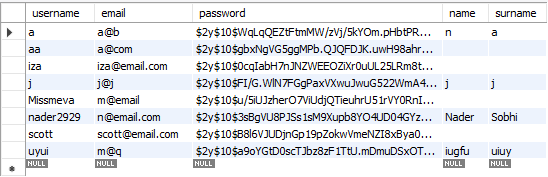


Figure Z3

Figure Z3 shows the *plant\_users* table columns as well as some of the records held in the table.

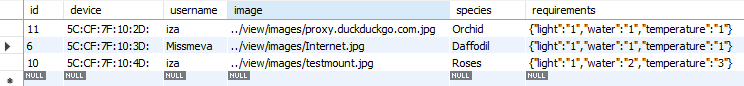


Figure Z4

Figure Z4 shows the *plants* table columns as well as some of the records held in the table.

# Discussion

## General Discussion

This section will look at the project overall, how it relates to similar solutions already on the market and how it meets the aims set out in a previous section.

The project has been very successful in terms of producing a hardware and software system that accurately collects, records and displays data about a specific plant. Whilst on paper this may seem like a trivial exercise, translating this into a system that is robust enough to deal with changing variables while collecting useful information is quite another challenge. Care had to be taken to ensure that the data collected from the sensors was in a human readable and useful format before it was sent to the database. The database itself had to be designed in such a way that the website could access it easily and simply extract the data it required. The website then had to use this data to power graphs and charts that the user could use to track the status of their plant over a given time frame.

It is important to evaluate the project in regards to similar devices currently on the market. The Plant Monitoring System created by Ryan Gill uses similar components to achieve a similar result, but it relies on slightly inflexible open source software to collect and render the data. The system developed by the team has enough scope that sensors can be added with only slight alterations to the code base. Gill’s system also requires technical expertise to put together, a factor the developed system negates. The total cost of Gill’s system is slightly less than the team’s system, but any potential saving is destroyed by the technical knowledge required to build it.

The Smart Plant Watering System by Elecrow again uses similar components, but adds in the feature of remote plant watering. However, this not only greatly inflates the cost to be over £100 more expensive, but the technical knowledge required to build it is vast. This system also avoids any data collection and uses only moisture sensors to coordinate its operations. However, the inclusion of remote plant watering makes it a very interesting device and should be considered in future iterations of the developed device.

The Smart Garden System by SwitchDoc is perhaps the industry leader in regards of smart plant monitoring. It uses a Raspberry Pi in place of an Arduino, which gives it much more processing power and the ability to add sensors and modify the code. However, it requires some technical knowledge to put together, takes up a much larger footprint and is also much more expensive that the developed device. The ability to monitor more than one plant from the same ‘brain’ is a highlight, as is its integration with digital assistants like Amazons Alexa.

All of the devices mentioned do very similar things as the developed device as well as features beyond the scope of this project. However, they all suffer from the same flaw: they require specialized knowledge to build, operate and maintain. This is where the developed device stands on its own.

In a previous section, the aims of the project were laid out. It is important to evaluate how well the project meets those aims discussed. The high-level aim that was set out stated that a product should be created that met the needs of the modern gardener, both those just discovering the field of horticulture as well as more seasoned individuals. The goal here was to develop a system that was not only easy to setup and use, but also one that would feedback useful information to the user. Overall, the project achieves this aim well. The physical device has been developed to include as little peripheral modules as possible whilst still maintaining a high degree of usability. To set up the device, the user has only to place the sensors where they feel is most accurate and then power the device on. Inserting data into the database is done automatically upon a WiFi connection being established. The user then has to register with the companion website to view the graphs and charts powered by the collected data. The website has been designed for use with mobile and desktop devices in mind.

In addition to this over-arching aim, several smaller aims were identified. Firstly, the website had to be responsive in order to display correctly on a mobile device as well as desktops, laptops and tablets. Consideration on which browsers being used played a part here, as older users may be prone to continued use of legacy browsers, such as Internet Explorer. Navigation and content of the website was also a crucial consideration, with the aim of ensuring as few button clicks as possible resulted in the user locating the desired information. Secondly, the device had to be easy to maintain with little or no expert knowledge required, as well as perform in a reliable manner. The developed hardware device is essentially ‘plug and play’, with the user having merely to place the sensors and power the device on. The components used are rugged enough to withstand repeated use but also simple in their operation. The reliability of the device is dependent on the WiFi connection. If a user’s WiFi service was to be interrupted, sensor readings may not be placed in the database. This, however, is out with the scope of the team but in attempt to lessen such an impact, sensor data is only sent to the database every thirty minutes. Lastly, collection of data had to be undertaken in a manner appropriate with current legislation, with consideration being given to how these laws may change in the future. While the hardware device is geared toward collecting data, the only piece of personal data the user has to provide is their email address. This is so that they can register a username and password for use on the site with the email address only being used should they request to reset their password. However, the website does contain a ‘Remember Me’ function that uses cookies to store session information. To inform the user of this, a pop-up is generated when they visit the site to inform them that cookies are used. After agreeing with this, they are free to use the website. This follows GDPR legislation dealing with the collection of cookies.

It is fair to say that the overall project meets the needs of the client as well as the aims set out. The client wanted an intuitive, adaptable system that could collect, record and store data on a given plant. This was achieved. The aims then expanded on the client’s needs to ensure the device was reliable, responsive and easy to use. This was also achieved.

## Conclusions

Smart Eco System is a unique product, designed to help the client attract a large cross section of society. Having access to the readings presented in the form of visually attractive graphs, as well as the indicators showing a plant’s demand for water, all they need to do is to regularly check its status by logging into the online platform or using the attached display.

Moreover, such a solution could potentially raise the prestige of the company and significantly contribute towards the development of the business, and thus increase the number of connections in the industry. It would change the perception of the company from a small, local shop to competition others have to reckon with. That, in turn, would translate into bigger profits due to their new-found reputation as a pioneer of smart home equipment.

However, if the client, for some reason, decided not to release the product, the aforementioned competitors would definitely take advantage of the company’s stagnation and introduce interesting technological solutions.

## Future Work

As with any project created to fit a client’s specification there are some features or aspects of the system that were identified as potentially beneficial but were not implemented due to some extenuating circumstances. Starting with the database, a NoSQL solution could have been implemented instead of the current model which consists of a mix of Relational Database Scheme (RDS) and a NoSQL scheme. The reasoning for this was that the information stored currently would only consists of some basic user information, the sensor readings from the device(s) and nothing else. However, in the future it may be prudent to switch the storage method to a NoSQL solution as it would allow for the easy expansion of any of the data-centric features.

The physical device responsible for collecting readings on a plants surroundings could be improved in a multitude of ways. These are mainly quality of life changes and therefore were not implemented at this point in time. Beginning with the implementation of some sort of interface for the Arduino Subsystem so as to allow the user to perform a few actions that are currently unavailable or only available through the web application. As far as the developer can identify there are only 2 ways that an interface could be implemented to allow the user to control the device’s functionality. The first would be a custom-built application that would interface with the device once it is connected to the user’s personal computer. This version would allow the user to make changes to the device’s functions through the interface and then the appropriate instructions would be loaded onto the device. This iteration would be the simplest to implement as the Windows OS can already recognize when an Arduino device is connected. The other method that could be used to implement an interface would be to use the website to allow the user to set some options that would control the functionality of the device and then load the appropriate instructions to it. This is probably the hardest interface to implement as it requires that two-way communication is possible between the device and the website. Additionally, it would require more storage for user preferences as well as more data to be collected on each individual user. Lastly and least importantly at this stage would be the design and construction of a plastic 3D printed case for the device to be housed. This was not carried out due to the extremely limited access the developer had to such a printer.

The website is the final component which holds potential for future upgrades or modifications. Beginning with the addition of a feature that would allow the user to regain access to their account should they forget their password. There are plenty of methods that can be used to implement such a feature, including email reset links or other two factor authentication methods. This was, again, not implemented due to time constraints and not due to difficulties in implementation. Another security feature that will be implemented in the future will be the use of the HTTPS protocol to encrypt connections between the users and the website. This was not implemented because the web hosting used did not support the developers preferred method of adding the required certificate to the website and thus would require more work and time than available.

Apart from security features there are some functional aspects that have not been implemented due to time constraints as well as the fact that they were deemed as small quality of life changes and therefore would not improve the functionality of the system by a significant amount. That being said, they will be discussed here as potential improvements for the system in the future. Starting with the implementation of a feature that would enable users to discuss plant care with plant experts or members of the user base who have proven to be extremely adept at plant care. This would allow for a kind of social network that would enable users to interact with other people who share the same passions as them. Finally, another feature that could be implemented in the future would be to add a “wiki” type section of the website that would host all kinds of information about plant care. While very easy to implement this can be very time consuming as it would require the developer in charge of the task to do extensive research so as to ensure that the information hosted on the website is accurate.

## Call to action - scotty

To find out more about how you can gain the power to track and optimize your plants lifecycle with the Abertay Plant Monitoring System, visit [www.abertaypms.com](http://www.abertayps.com) today to arrange a free, in-depth demonstration. The first fifty people to sign up for a demonstration session will receive 25% off the total cost of purchasing the system and six months of ongoing training and technical support. The first hundred people to sign up will receive six months of ongoing training and technical support. The only way to take advantage of these offers is to visit [www.abertaypms.com](http://www.abertaypms.com) today and register your interest in the Abertay Plant Monitoring System!

References

**For URLs, Blogs:**

Bremer, J. 2012. *x86 API Hooking Demystified*. [blog]. 2 July. Available from: [http://jbremer.org/x86http://jbremer.org/x86-api-hooking-demystified/api-hooking-demystified/](http://jbremer.org/x86-api-hooking-demystified/) [Accessed 15 April 2016].

**Ours:**

**https://www.smallbusinesscomputing.com/News/Networking/what-the-internet-of-things-means-for-small-business.html**

<https://www.w3.org/standards/webdesign/accessibility>

<https://www.projectsmart.co.uk/project-planning-step-by-step.php>