**Greenhouse System Scale Model**

By: Harshit Walia, Dhyey Farasram and Jainam Shah

Course: CENG 355

Date: 03/23/2020

# Declaration of Joint Authorship

We, Harshit Walia, Dhyey Farasram, and Jainam Shah confirm that this work submitted for assessment is the joint work of ourselves, and is expressed in our own words. Any uses made within of other works of any other author, in any form (ideas, equations, figures, previous technologies, tables, programs, texts) are properly acknowledged at the point of use. A list of the references used is included. Everyone equally contributed to all aspects of this project, with Harshit's and Dhyey's main focus being on mobile application and hardware, while Jainam mainly worked on the database and user interface aspect of the project.

# 

# Approved Proposal

## Executive Summary

As students in the Computer Engineering Technology program, we will be integrating knowledge and skills we have learned from this program into this Internet of Things capstone project. This proposal requests the approval to merge the hardware built by us in the course CENG 317 which consists of the soil moisture sensor, temperature sensor, and a relay module with water pump connected to the Raspberry Pi 3B+ ( the CPU we have used) and the software, an android application, which we developed in the course CENG 319. The app allows the user to view the temperature and soil moisture readings which we will get from the sensors that are connected to the RPI through the internet. The Firebase database which we have set will be used to store the temperature and soil moisture data. As this project has a lot to deal with plants, we decided to collaborate with Valeria Wuschnakowski, the greenhouse technician at Humber. This semester we are working on merging the project from the previous semester. So, in the final project, we will establish a connection between RPI, database and the android application, so that the sensors will send all the readings to the RPI and we can store them in our database. The user can use the app to view those readings.

## Background

The major idea behind this project is to help the employees of Humber Arboretum and Humber Greenhouse as they do not have any self-monitoring system to measure the temperature and soil moisture. Also, the plant watering system is not fully automatic. By developing this project, they will be able to keep records of the temperature and soil moisture of the plants. It will also allow them to water the plants where the moisture level has dropped to a below-average level.

E.g. If there is a reading in the app where it shows that the moisture level is below par, the plants will be automatically irrigated through the water pump connected to the relay module. I t will allow the user to monitor there plants remotely.

## Concluding Remarks

This proposal provides a plan to replace the manually working system with a fully automatic working model. This is an opportunity for us to provide a prototype of the solution for the Humber Greenhouse by integrating all the skills we have acquired in this program. We request approval of this project.

## Abstract

The Greenhouse System Scale project mainly focuses on providing a solution to monitor plant health by providing the real-time temperature and soil moisture data readings to the user via an app-based platform that will be intuitive and easy to use.

## The intention behind the project:

The project has been developed keeping in mind the needs of the Humber Arboretum staff, currently, their plant monitoring system doesn't have a user friendly and easy to use platform which can be solved by building a user-friendly app. The mobile application will display those levels on the screen when the user login his/ her account. The most recent levels and limited levels will be displayed on the screen along with the time when they were measured. It will also provide intuitive graphs regarding temperature and soil moisture data.

Secondly, the plants had to be manually watered, so we also wanted to automate the process by developing a plant watering system that will automatically water the plants when the soil moisture level drops below a certain limit.

This app will provide the air temperature readings and the soil moisture readings to the users via interactive bar graphs, and it would also allow them to supply water by controlling the water supply pump to any specific region where the moisture level is below par level. The temperature and the soil moisture readings are stored in the database which also includes a timestamp for the readings. The database being used is Google's firebase database. The app shows the temperature graphs and the soil moisture graphs in real-time. The water supply control allows the user to manually override the automated irrigation system and select a timer to turn the water pump on and irrigate the plants. The user can set his/her time limit to supply water to the plants and also cancel whenever he/she wants.

Contents

[Declaration of Joint Authorship 2](#_Toc38053087)

[Approved Proposal 3](#_Toc38053088)

[Executive Summary 3](#_Toc38053089)

[Background 3](#_Toc38053090)

[Concluding Remarks 4](#_Toc38053091)

[Abstract 4](#_Toc38053092)

[The intention behind the project: 5](#_Toc38053093)

[1.Introduction 8](#_Toc38053094)

[2. Parts and Budgets 9](#_Toc38053095)

[2.1 Components 9](#_Toc38053096)

[2.2 Component Specification: 10](#_Toc38053097)

[2.2a) Computer being used: Raspberry Pi 3 B+ 10](#_Toc38053098)

[2.2b) Sunfounder 4 channel relay- module 11](#_Toc38053099)

[2.2c) Submersible water pump 12](#_Toc38053100)

[2.2d) DHT11 temperature sensor: 13](#_Toc38053101)

[2.2e) Soil moisture sensor: 14](#_Toc38053102)

[2.2f) MCP3008 IC: 15](#_Toc38053103)

[3. Time Estimation 16](#_Toc38053104)

[3.1 DHT11 and Soil moisture sensors 16](#_Toc38053105)

[3.2 Relay and Motor pump 16](#_Toc38053106)

[3.3 Combining the individual hardware parts 16](#_Toc38053107)

[4. Hardware Assembly 17](#_Toc38053108)

[4.1 DHT11 and Soil Moisture sensor: 17](#_Toc38053109)

[4.1 Soil Moisture and DHT11 18](#_Toc38053110)

[4.2 Relay and Motor 18](#_Toc38053111)

[5. PCB Board Designs 20](#_Toc38053112)

[5.1.a: PCB Board Design .fzz file 20](#_Toc38053113)

[5.3.b: PCB Board Top view 21](#_Toc38053114)

[5.3.c: PCB Board Design Bottom view 21](#_Toc38053115)

[6. Database setup 22](#_Toc38053116)

[6.1 Data Structure 22](#_Toc38053117)

[6.2 Database Design 23](#_Toc38053118)

[7. Android Design and Layout 24](#_Toc38053119)

[7.1 App Design and Specifications 24](#_Toc38053120)

[7.2 Design layout 25](#_Toc38053121)

[7.3 Android Components 29](#_Toc38053122)

[**8. Device Setup** 30](#_Toc38053123)

[**9. Problems Encountered** 30](#_Toc38053124)

[9.1 Interpreting reading from the Soil Moisture Sensor: 30](#_Toc38053125)

[9.2 Storing and separating readings from sensors for each user: 31](#_Toc38053126)

[9.3 Data Communication: 32](#_Toc38053127)

[10. Progress reports 33](#_Toc38053128)

[10.1 Report 1 33](#_Toc38053129)

[10.2 Report 2 35](#_Toc38053130)

[10.3 Report 3 36](#_Toc38053131)

[10.4 Report 4 38](#_Toc38053132)

[11. Testing 40](#_Toc38053133)

[11.1 Unit testing 40](#_Toc38053134)

[11.2 Production testing 40](#_Toc38053135)

[12. Conclusion 41](#_Toc38053136)

[12. Recommendations 42](#_Toc38053137)

[13. GitHub 43](#_Toc38053138)

[14. References 43](#_Toc38053139)

# 1.Introduction

The venture has been developed retaining in thoughts the wishes of the Humber Arboretum staff, currently, their plant monitoring machine doesn’t have a person-pleasant and clean to use platform which can be solved by using constructing a consumer-friendly app. The cellular utility will display those degrees on the display while the consumer login is his/ her account. The latest tiers and constrained ranges could be displayed on display at the side of the time whilst they were measured. It will also provide intuitive graphs concerning temperature and soil moisture data.  
  
Secondly, the flowers ought to be manually watered, so we also desired to create and automate the technique by using growing a plant watering device so one can robotically water the flora whilst the soil moisture level drops underneath a certain restrict.

This app will offer the air temperature readings, and the soil moisture readings to the users via interactive bar graphs, and it'd also allow them to deliver water with the aid of controlling the water deliver pump to any specific location wherein the moisture degree is under par degree. The temperature and the soil moisture readings are stored inside the database which additionally consists of a timestamp for the readings. The database being used is Google’s firebase database. The app shows the temperature graphs and the soil moisture graphs in real-time. The water supply control permits the user to manually override the automated irrigation gadget and choose a timer to turn the water pump on and irrigate the vegetation. The user can set his/her own time to restrict to supply water to the vegetation and also cancel every time he/she wants.

# 2. Parts and Budgets

## 2.1 Components

* SunFounder 4 Channel DC 5V Relay Module                 $15
* Raspberry Pi 3B+                                $65
* 2xMCP3008 - 8-Channel 10-Bit ADC                        $7.50
* Soil moisture sensor                                 $10
* DHT-11(temperature sensor)                          $7
* Submersible Motor Pump Water Pumps DC 3-4.5V 100L/H          $8
* Silicon Tubing                                     $10
* Battery Holder                                    $5
* AA Batteries                                     $3

**Total Cost: $130.50**

## 2.2 Component Specification:

## 2.2a) Computer being used: Raspberry Pi 3 B+



Raspberry Pi 3 Specifications:

SoC: Broadcom BCM2837

CPU: 4× ARM Cortex-A53, 1.2GHz

GPU: Broadcom VideoCore IV

RAM: 1GB LPDDR2 (900 MHz)

Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless

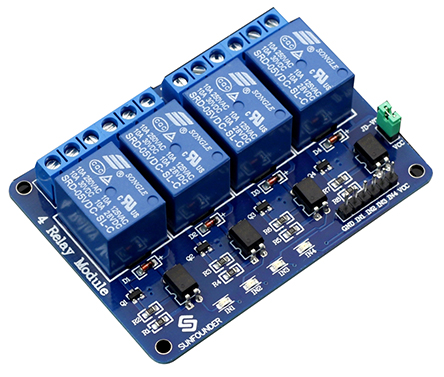
Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy

Storage: MicroSD

GPIO: 40-pin header, populated

Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

## 2.2b) Sunfounder 4 channel relay- module



This is a 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by a microcontroller.

Pin Description

Input:

VCC: Positive supply voltage

GND: Ground

IN1--IN4: Relay control port

Output:

Connect a load, DC 30V/10A，AC 250V/10A

## 2.2c) Submersible water pump



**Specification：**

            Working Voltage: 4v-12V

            Working Current: 0.8A

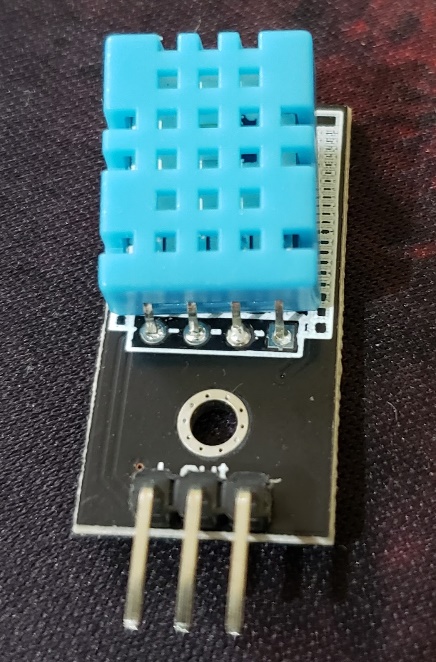
            Motor Diameter: 27mm

            Water Pump lenght: 52mm

            Drain Hole: 4mm

            Weight:70g

## 2.2d) DHT11 temperature sensor:



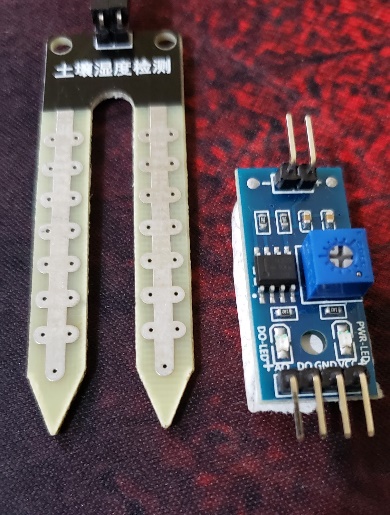
The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

Pin Description

VCC: Positive supply voltage

GND: Ground Output: Digital output

## 2.2e) Soil moisture sensor:



The Soil Moisture Sensor measures soil moisture grace to the changes in electrical conductivity of the earth (soil resistance increases with drought). The electrical resistance is measured between the two electrodes of the sensor. A comparator activates a digital output when an adjustable threshold is exceeded.

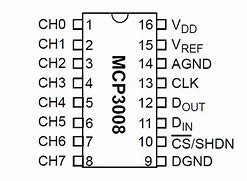
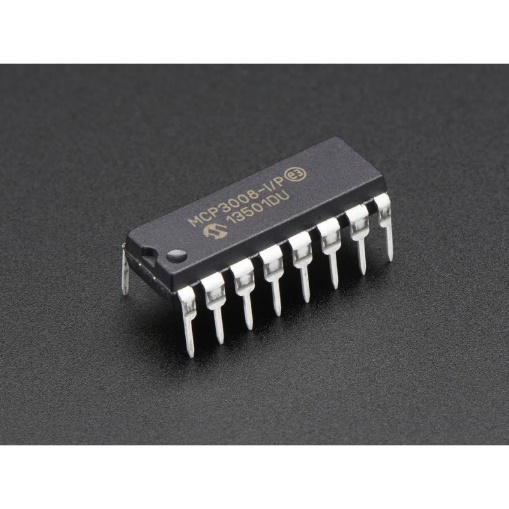
Pin Description:

VCC: Positive supply voltage

GND: Ground

Output: Analog output

## 2.2f) MCP3008 IC:



The MCP3008 is an 8-Channel 10-bit ADC IC, so it can measure 8 different analog voltages with a resolution of 10-bit. It measures the value of analog voltage from 0-1023 and sends the value to a microcontroller or microprocessor through SPI communication.

# 3. Time Estimation

The project was completed in 2 stages, in the first stage the soil moisture and temperature sensors were tested and assembled on a separate PCB and the real module and water pump on a different PCB board.

In the second stage of the project, they both were combined into a single circuit.

## 3.1 DHT11 and Soil moisture sensors

* Purchasing and delivering parts took approximately a week and a half
* Assembling, and breadboard testing took about 2 weeks
* PCB designing - 1 week
* PCB soldering – 1 week
* Implementing codes took over 2 weeks.
* Total: approximately 6 weeks to get them fully working.

## 3.2 Relay and Motor pump

* Purchasing and delivering parts took approximately more than a week.
* Assembling, and breadboard testing took about 2 weeks
* PCB designing - 1 week
* PCB soldering – 1 week
* Implementing codes took over 2 weeks.
* Total: approximately 6 weeks to get them fully working.

## 3.3 Combining the individual hardware parts

* Assembling, and breadboard testing took about 2 weeks
* PCB redesigning - 1 week
* PCB soldering – 1 week
* Combining the source codes and testing took over 2 weeks.
* Total: approximately 6 weeks to get them fully working.

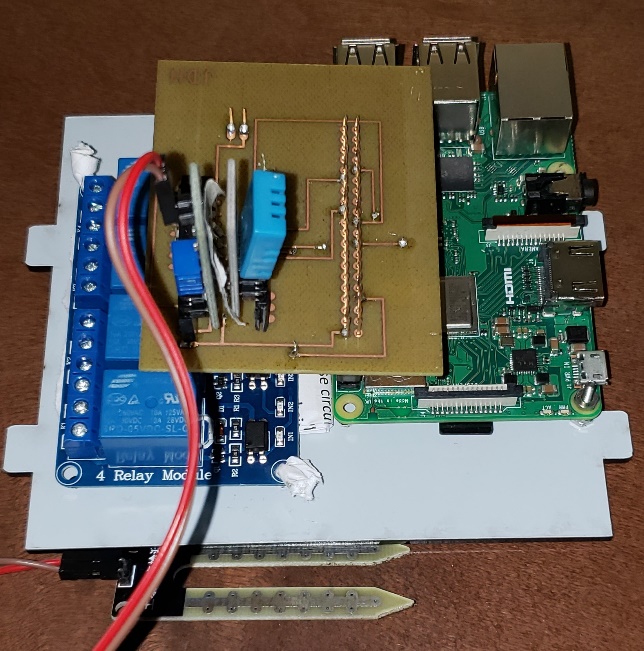
Total time is taken to complete the project – 18 weeks

# 4. Hardware Assembly

The sensors and the PCB were tested in different ways to make sure if they are working fine or is there any problem with them.

## 4.1 DHT11 and Soil Moisture sensor:

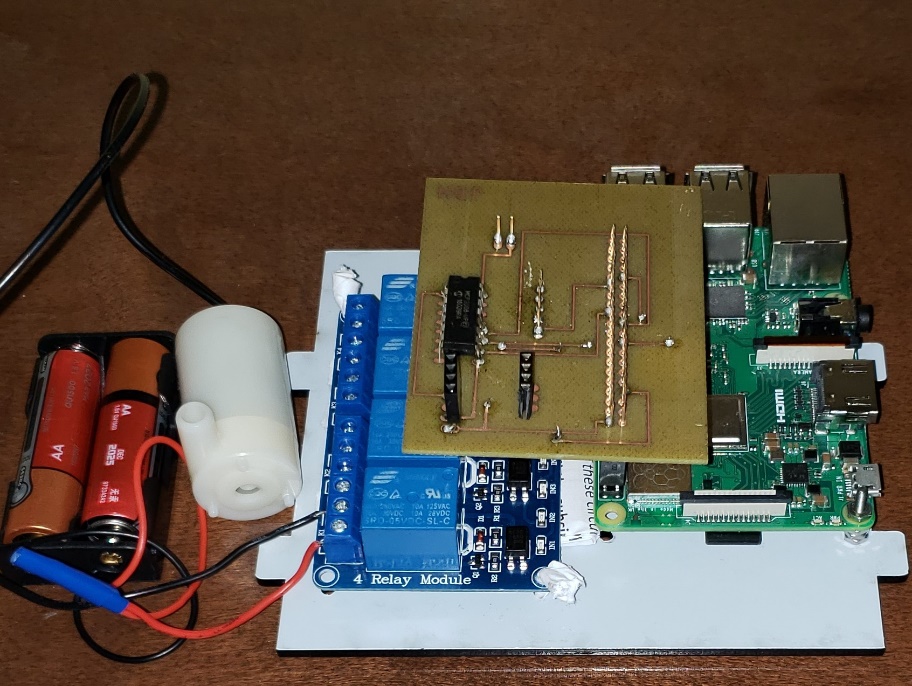
* This hardware must be set up on a plant that you want to keep track of its condition. This hardware will provide us with the ongoing conditions of plants, such as the current water level or the soil moisture and the temperature surrounding the plant.
* The probes of the soil sensor must be inserted into the soil to retrieve the health information of the plant. This sensor has two green led that will tell us if the sensor is working or not. The LEDs are named as PWR-LED and DO-LED respectively the pwr-led will tell us if the sensor is powered on and working properly and the other led will tell us if they detect water or not.
* DHT11 sensor will give as the temperature and pressure readings of the surrounding.
* The DHT11 and soil moisture sensors are plugged into the female headers which soldered to the PCB.



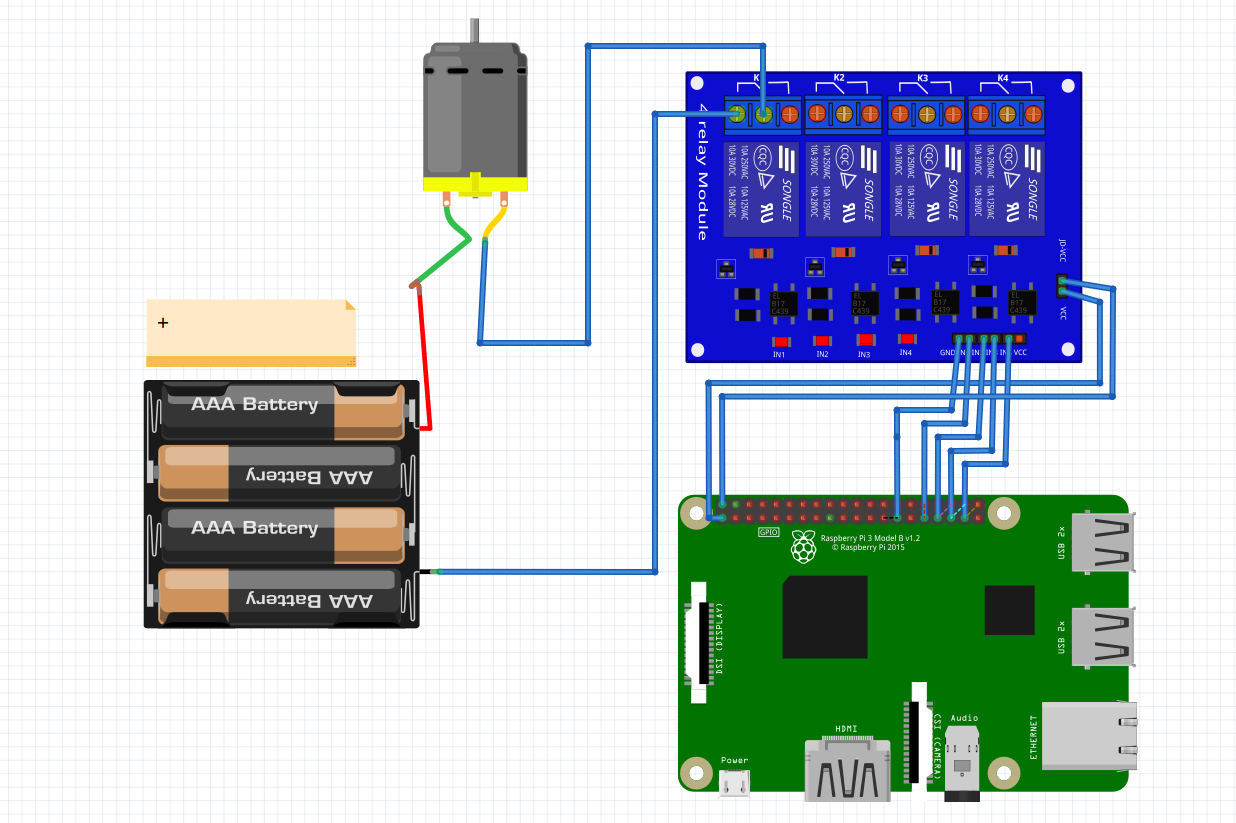
# 4.1 Soil Moisture and DHT11

## 4.2 Relay and Motor

* When the device receives the health updates from the sensor, the device will be able to control the motor. When the soil moisture will drop below a certain level the motor will start will pumping the water automatically. When the water is back to its level the motor will stop itself.
* The motor pump and the battery are connected to channel 1 of the relay module. Upon a signal from the relay, the motor is turned on/off.
* The relay is connected to the PCB via a female header soldered to the bottom side of the PCB.

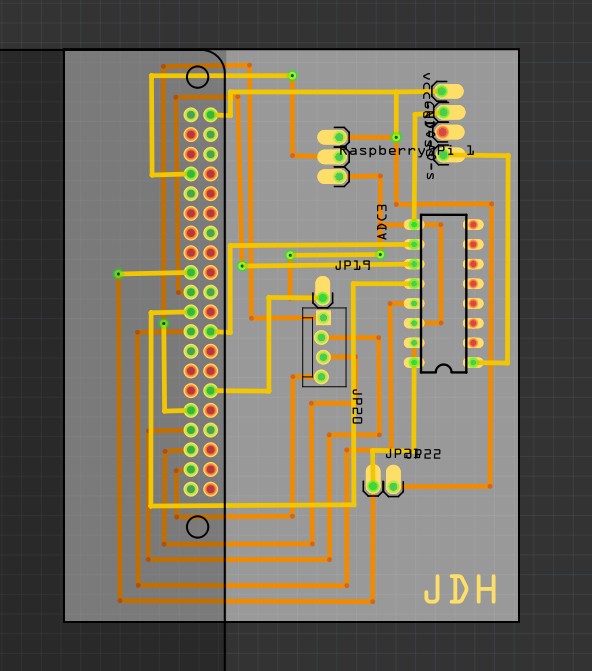


4.2.a: Relay and Motor

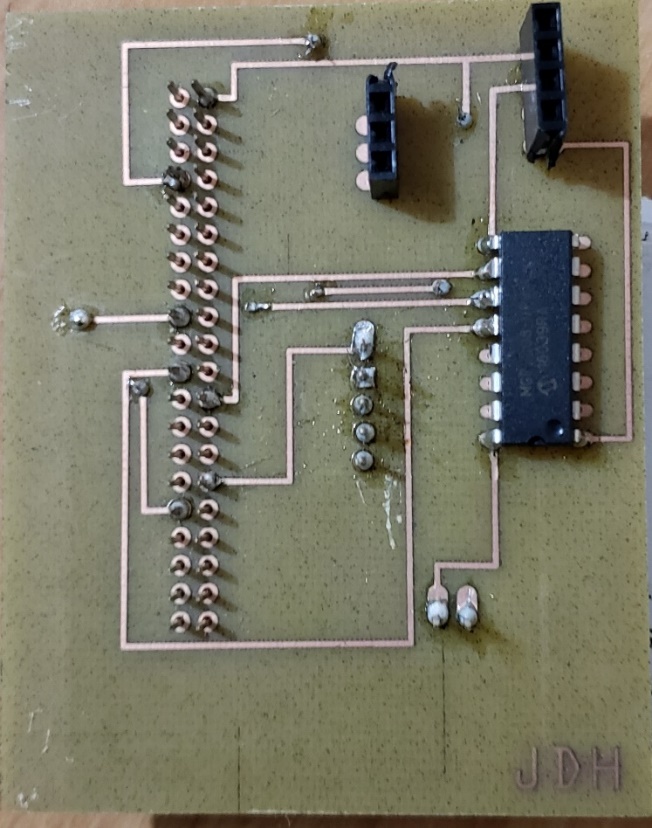


**Circuit demonstrating the working of the motor and relay module**

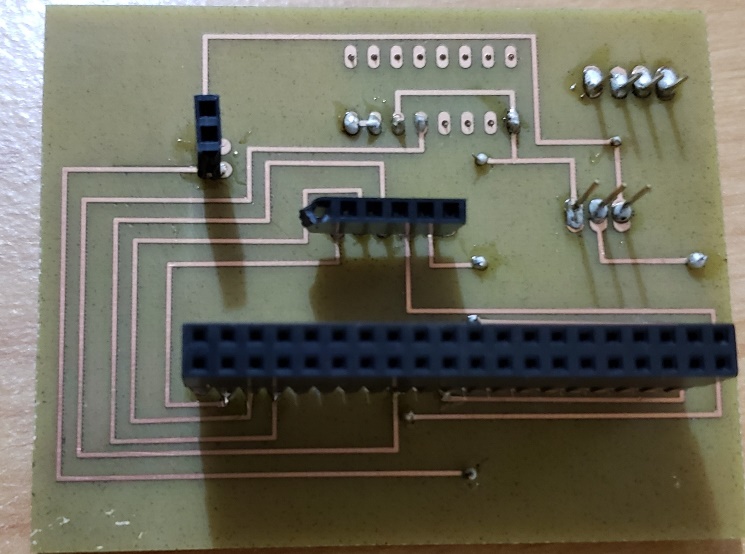
# 5. PCB Board Designs



### 5.1.a: PCB Board Design .fzz file



### 5.3.b: PCB Board Top view



### 5.3.c: PCB Board Design Bottom view

# 

# 6. Database setup

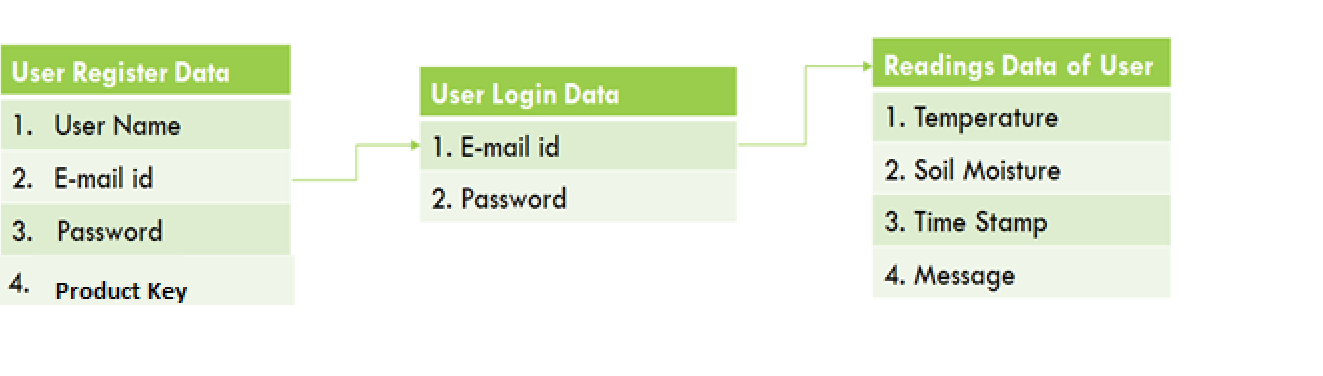
We used Google’s firebase real-time database Version 6.12.2 for our project.

## 6.1 Data Structure

Our database contains these fields:

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Use |
| User Name | String | Stores user name |
| Email ID | String | Stores Email of the user |
| Product Key | String | Stores the product key received along with the device |
| temperature | String | Holds the temperature readings sent by the temperature sensor |
| soil moisture | String | Holds the soil moisture level readings sent by the soil moisture sensor. |
| timestamp | String | Holds a timestamp generated when the readings were recorded |

## 6.2 Database Design



The database we will be using will be Google’s Firebase.

For user management, we will be using authentication and our sign-in method is via email Id.

Once a new user has registered and successfully logged in, he will be able to access the Readings data.

All the reading's data will be stored under the unique product key obtained along with the purchase of the device. This allows us to separated readings and data for every user.

# 

# 7. Android Design and Layout

The app was built and complied with Android Studio 3.6.

Running environments- The app is compatible to run on any android above Android 4.1 Jelly Bean (API 16)

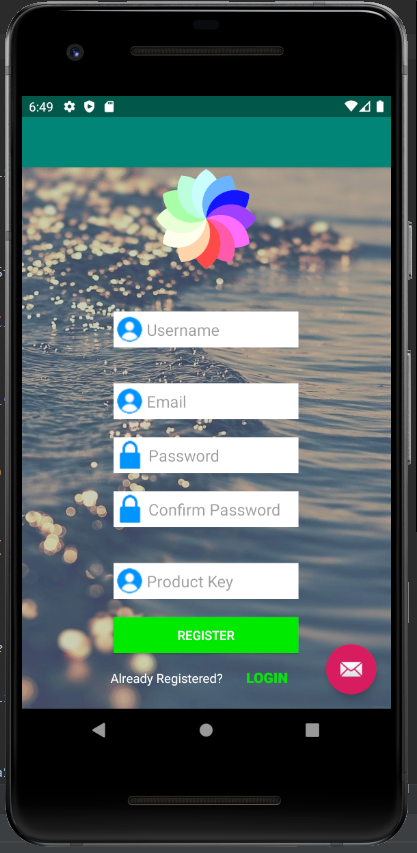
## 7.1 App Design and Specifications

* Easy to the interface, the Greenhouse Scale App is designed keeping in mind the employees of Humber Arboretum and the general public. It is easy to use and is pretty straight forward, the main readings page provides the most recent temperature and soil moisture readings with a timestamp indicating the date and time when it was recorded.
* The software will also be allowing the employees to supply water to the plants where the moisture level is below average.
* Intuitive and interactive, the app has provides a graphical interface of temperature and soil moisture readings in the form of bar graphs which are easy to read and provide information about the most recent readings. The graph will display the recent readings with a timestamp so they can have a brief idea of when the reading was acquired.
* The app will have some basic settings features too. The users can change the temperature reading unit from Celsius to Fahrenheit. They can change the app language from English to French.
* The water supply control feature will allow the user to supply water to the plants whose moisture level is below the ideal percentage. They can limit the flow of water with the timer feature which asks the user to select the no. of minutes for which they want to supply water to the plants.
* Realtime updates on readings and graphical interface, as soon as a new reading is uploaded to the database, the temperature and soil moisture graphs update to accommodate the new reading and the values on the main readings page

## 7.2 Design layout

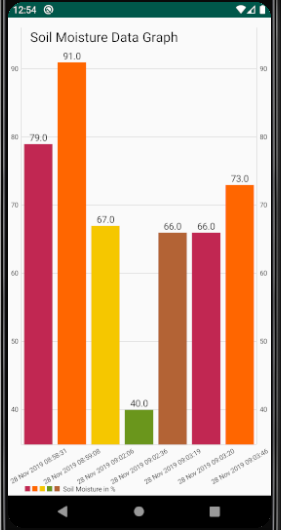
The app has the following pages: login page, sign up page for new users, main Readings Page, temperature readings page, soil moisture readings page and water supply control page, help page, Settings page. The app interface has a side menu to switch between pages which will be accessible from all pages.

**Login Screen and Register Page:**

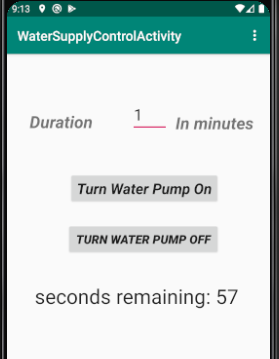
A screenshot of a cell phone

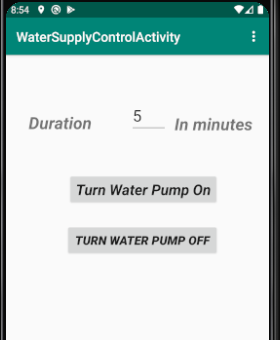
Description automatically generated

**Soil Moisture Readings Page and Temperature Readings Page:**

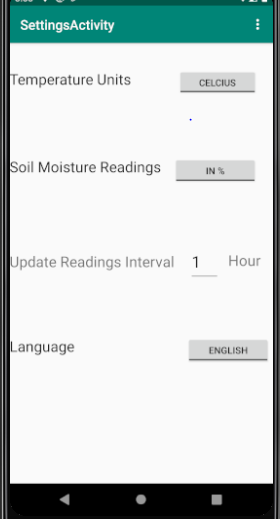
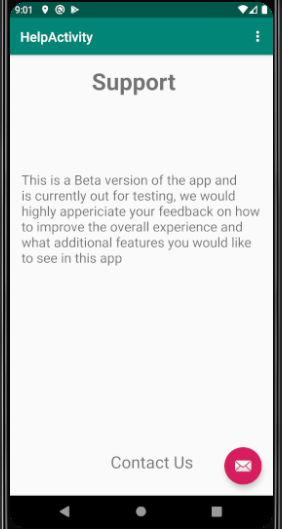
A screenshot of a cell phone

Description automatically generated

**Water Supply Control Page:**

****

**Settings Page and Help Page:**



# 7.3 Android Components

1. Image View: It displays image resources, such as Bitmap or Drawable resources. It is also commonly used to an image and handle image scaling.
2. Toggle Button: A toggle button allows the user to change a setting between two states. A user can add a basic toggle button to your layout with the ToggleButton object. Android 4.0 introduces another kind of toggle button called a switch that provides a slider control, by which you can add a Switch object.
3. Bar Graphs: It allows the user to set a bar graph that can be used for comparison. For our project, we used it to show readings of temperature and soil moisture on different days.
4. Countdown Timer: It schedules a countdown until a time in the future, with regular notifications on intervals along the way.
5. Time Stamp: A thin wrapper around “java.util.Date” that allows the JDBC API to identify this as an SQL TIMESTAMP value. We have used timestamp to keep a note for the user so they can know when the readings were captured.;

Number picker: A component that enables the user to select a number from a predefined range. We need to set the minimum value as well as the maximum value for the user

# **8. Device Setup**

Step 1

Install the android app on your device and register using the product key provided along with the hardware and create an account.

Step 2:

Assemble the hardware components and put in the soil moisture sensor in the soil and place the device near the plant.

Step 3:

Place the water pump inside a reservoir and the end of the silicon tubing near the plant base.

Step 4. Power the device and monitor your plant health remotely via the android app.

# **9. Problems Encountered**

## 9.1 Interpreting reading from the Soil Moisture Sensor:

 The output given by the soil moisture sensor was raw data which was a fractional value between 0 and 1. This couldn’t allow us to find exactly how much moisture level was present in the soil.

Approach to the problem:

After working on the python code, the output was changed to either “Water Detected” in case

the original output is 0 or “No Water Detected” in case the output is 1but it still   Needed extra

component to get the moisture level in %. To work that out we had to order an IC chip

and fit it with the soil moisture sensor to convert it into an analog output.

Code: After adding the IC to the hardware, the outcome came out as a large number between 0 to 1024. So, in the code, we had to add an equation that would convert the raw value we got into an equivalent percentage number. After working that out we were able to present the data in percentage form between 0 to 100.

## 9.2 Storing and separating readings from sensors for each user:

The data from the sensor from the raspberry pi was being uploaded to the firebase database under a single node which restricted our model to single person use. To add multiple user functionality, we had to figure out a way to go around this problem to differentiate the readings for every single user.

Approach to the problem:

The data retrieved from the sensors were being uploaded to the database without being stored under user’s unique id as the code for every device manufactured had to the same and changes couldn’t be made to every piece of hardware manufactured, keeping that in mind we needed a way to separate data storage of sensor readings for each user without hardcoding user id in the raspberry pi’s code of each device. So, we introduced a new variable “Product Key” which is to be distributed with each greenhouse system scale model and will be required to register on the App. This allows us to upload the readings from that specific device to a node of that specific product key in the firebase database and allows for user data to be separated.

## 9.3 Data Communication:

The system is made of two platforms that are required to communicate with one another to gather and transfer data to a database. Because of this, there were many problems with receiving the data in a way that it could then be easily sent to the database as the database also was not able to be connected through the project’s original plan.

Approach to the problem:

To solve this issue, we decided to use Google’s firebase database as the libraries to upload readings to the database were easily available and the data was easily retrieved on the android app as well. This choice made it easier for the readings to be sent and retrieved on the app effective and easy.

# 10. Progress reports

## 10.1 Report 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Harshit Walia <meharshwalia@gmail.com>** | | Mon, Sep 16, 2019, 10:48 PM |  |  |
| |  | | --- | | to valeria.wuschnakowski, bcc: Jainam, bcc: dhyey | | | |

Hey Ms. Wuschnakowski,

I hope you are doing well. I, Harshit Walia along with my teammates Jainam Shah and Dhyey Farasram have selected the Green House System Scale Model for Humber Arboretum as our Capstone final year project for Computer Engineering Technology. Here's the link to the project <https://app.riipen.com/projects/7324>.

As a part of our project, our Professor Mr. Austin recommended you be our collaborator for this project. We will be glad to have you as our collaborator for this project and use your valuable feedback and suggestions to progress with the project. Your experience as a Greenhouse Technician will be highly valuable for us.

We will be building a system that includes a mobile app connected to a microcomputer which measures the temperature, soil moisture level of the plants in the arboretum's nursery and also automatically waters the plants when the moisture level goes below a limit. The basic outline of the app is also attached to the email. Through the app, the employees of the arboretum will be able to check the readings in real-time and will also be able to water the plants by just pressing a button on the app.

The project is divided into 2 semesters (Fall and Winter). There is a requirement that we meet with the collaborator at least 3 times each semester to get feedback on the progress we will be making with our project.

We will highly appreciate if you will be able to take some time out to meet with us on September 24(Tuesday)  anytime between 8:00 am to 11:35 am in J213 during one of our lab session.

Regards,

Harshit Walia

Team Leader

## 10.2 Report 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Harshit Walia <meharshwalia@gmail.com>** | | AttachmentsFri, Oct 11, 2019, 10:43 PM |  |  |
| |  | | --- | | to valeria.wuschnakowski, Austin, Jainam, dhyey | | | |

Hey Ms.

I am writing this email to provide you with an update about the progress we have made in the project. We have designed the layout of our android app which is attached to this email as SoftwareMockupFinal.pdf and have also coded 3 interface screens as of now and are continuously progressing to build the app. On the hardware side, we have received all the parts needed for this project and currently testing the sensors on a breadboard and have planned its layout and the schematic of how the hardware will be designed on a PCB. The Gantt Chart describing the task that we will be focusing on each week is also attached to this email as software\_gant.pdf.

Kindly provide us with your feedback or any comments you may have about the project. Please let us know if you would schedule a meeting to discuss further the project.

Regards,

Harshit Walia

(Team Leader)

## 10.3 Report 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Harshit Walia <meharshwalia@gmail.com>** | | Mon, Feb 3, 3:19 PM |  |  |
| |  | | --- | | to valeria.wuschnakowski, Austin, Jainam, dhyey | | | |

Good Afternoon Ms. Wuschnakowski,

Hope you are well. Thank you for being an excellent collaborator for us. As previously discussed, we built an app that would help the employees of Greenhouse to maintain the temperature and soil moisture readings along with the hardware which would sense it.  
  
This semester we are going to combine, both, the app and the hardware which we worked upon individually last semester, and build the final project. Till now, we have combined all the sensors into one working model. The PCB is ready. We also have the python code running through which the sensors can communicate with the Raspberry Pi.  
The App in the previous semester was built in mind keeping all the requirements in mind, so we do not have much to change in it. But, we have to still work on sending the readings from the RPI to the firebase database. Once they have been uploaded we will be able to display them on the app. We will be working on the database this week.  
  
Individual Status Report:  
Student A: Harshit Walia  
Student B: Dhyey Farasram  
Student C: Jainam Shah  
  
Student A did the breadboard testing, PCB testing, and also the PCB beep testing. Student B worked on building the PCB in Fritzing, which we got the final version after multiple attempts. Student C did the PCB soldering. All the members equally contributed to building the python code.  
  
Budget:  
The project's total cost around $130. Last semester, the budget was $120, but this semester we had to order an IC to get the soil moisture readings which cost $10. The RPI 3B+ was for $65, the soil moisture sensor was for $10, the DHT11 temperature sensor was for $7, the relay module was for $15 and the water pump along with the silicon tubing cost about $10.  
  
Github Link:  
<https://github.com/WaliaHarsh/GreenhouseSystemProject>

Regards

Harshit Walia

## 10.4 Report 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Harshit Walia <meharshwalia@gmail.com>** | | Mon, Mar 2, 3:08 PM |  |  |
| |  | | --- | | to valeria.wuschnakowski, Austin, Jainam, dhyey | | | |

Good Afternoon Ms. Wushnaowksi,

Hope you are well. Thank you for being an excellent collaborator for us. As previously discussed, we built an app that would help the employees of Greenhouse to maintain the temperature and soil moisture readings along with the hardware which would sense it.

As promised earlier we merged the app and the hardware which we worked upon individually last semester, and it's fully functional and ready to use. We have combined all the sensors into one working model. The PCB is ready. We also have the python code running through which the sensors can communicate with the Raspberry Pi.

The App is fully functional as of now. The python code we developed is providing accurate readings. We have successfully established a connection between the Raspberry Pi and google's firebase database, the readings from the pi are being uploaded to the firebase database. The app displays these readings in the form of interactive graphs.

Individual Status Report:

Student A: Harshit Walia

Student B: Dhyey Farasram

Student C: Jainam Shah

Student A has worked on the connection between the database and the Raspberry Pi, through which the app is getting desired outputs. Student B has worked on the App and added the missing requirements. Student C has worked on developing the Python code.

Github Link:

<https://github.com/WaliaHarsh/GreenhouseSystemProject>

Regards

Harshit Walia

# 11. Testing

## 11.1 Unit testing

The temperature sensor ought to be restoring the right estimation of the present temperature in degrees Celsius. The moisture and soil dampness ought to be restoring an incentive somewhere in the range of 0 and 100, which is the percent value. The soil dampness should display 0% if it isn't put in the soil. Should any of the qualities be returning incorrectly, ensure that the sensor is set in the mount accurately and that the bind is effectively applied.

## 11.2 Production testing

Guarantee that the motor is turning on once consistently for 5 seconds and that the values returned from then sensors are right. Should anything not be restoring the right worth or the motor isn't turning on (When not in damp soil), follow the unit testing guide above to help investigate your concern.

# 12. Conclusion

This framework has been created to meet the objective of making a little, cheap framework for checking a check on the plant habitat. It can measure the surrounding temperature, relative humidity, and soil dampness. It then forms this information and sends it to a database. From the database, this framework can deliver this information to a smart-phone application, to permit ease of access, diminishing the time and labour that would be required to screen these natural factors physically. The ultimate adaptation of the venture meets all of the prerequisites that were set out to be finished. Extra highlights that were presented amid the creation of the venture have been considered for future modifications of the framework. There are several extra highlights arranged to count but not restricted to.

# 12. Recommendations

Currently, with the hardware still inside the prototype segment, there are plans to lessen the value consistent with the unit. The current fee for a single unit is pretty small, only about $130.50 for a version without a database and a corresponding app may be made there are a few approaches to lessen this cost similarly which include buying the sensors in bulk and receiving a discount on the purchase, a reduced shipping fee associated in keeping with part and they would now not be shipped individually, and the possibility of the use of more efficient materials.  The code can be advanced, refined and be made personally to a customer’s needs. A light supply could be delivered to the growing place that might allow the plant life to receive light all through time in which mild level is not excessive enough, like for the duration of the night time or darker days at some point in the winter. This can be accomplished through the use of LEDs and virtual pins. As this might provide inaccurate records readings from the temperature sensor, it would be high-quality to have the lighting off before and throughout while the readings are being done. Modifications can be made to the field to permit extra water to escape within the occasion that a soil moisture sensor is wiped out and starts giving readings where the soil moisture is just too low, despite it being at the correct amount. Another option for this trouble is using AC energy with the sensor. Because DC energy is used, slowly the gold plating from one of the prongs on the sensor will elevate off and travel to the other sensor. Also, adjusting the orientation of the relay module and the connections to allow it to sit on top of the RPI and no longer to the side would reduce the quantity of place it occupies. Some connections at the board require you to have a small wire connecting one side of the board to the other to permit the pin headers to make a proper connection. Another approach to this is with the aid of having plated-through holes. A Bluetooth module could be used in soil sensors so that the framework can be established at the desired location and the soil sensor to the other location far away from the framework. This will help in maintaining the framework from circuit failure.

# 13. GitHub

The source code to our android application as well as the raspberry python program can be found in the

Following GitHub link:

<https://github.com/WaliaHarsh/GreenhouseSystemProject>

# 14. References

IDG Communication Inc. (2020).Temperature and Soil Moisture Graphs.  Retrieved from <https://www.javaworld.com/article/3226733/graphlib-an-open-source-android-library-for-graphs.html>

Google Developers. (2020). Android Developer Widgets. Retrieved from <https://developer.android.com/reference/android>

Szazo, Github Inc. (2020).DHT 11 Temperature Sensor. Retrieved from <https://github.com/szazo/DHT11_Python>

Bancha Koomwang, Youtube Inc.Soil Moisture Sensor. Retrieved from <https://www.youtube.com/watch?v=U7qG6BPVLLE>

Raspberry Pi Foundation UK Registered Charity 1129409. Water Pump with Relay Module. Retrieved from <https://www.raspberrypi.org/forums/viewtopic.php?t=144357>