Group 17 – SUMMER
Senior Design 1

SmartPlant Hub

Self-contained Plant Monitoring System With Companion App

Daniel Bohl - Computer Engineering
Hunter Cheung - Computer Engineering
Brendon Hales - Computer Engineering
Kristen Marks - Electrical Engineering

Table of Contents

1.0 Executive Summary

Owning and caring for plants, whether it be a small houseplant or a garden of vegetables or fruit, has always been a hobby for many people. Although some are not at all hard to take care of, many are; especially if you intend on ensuring that your plants are in the best environment possible. This will not only allow your plants to live longer, but for food bearing plants it will allow them to grow healthier as well. There's nothing worse in plant ownership than putting your all into taking care of a plant only to have it die because one of its environmental factors was lacking, and this is the problem we aim to solve.

When it comes to taking care of plants, nothing is more important to their livelihood than the environment they are living in. This is the reason many fruits and vegetables have to be grown in certain states or countries, as their climates are best suited for those plants. We want to make monitoring the most important factors of the environment easy for anyone, and the best way to do that would be to allow an owner access to all the information they might need in one place: in their pocket.

Our project aims to allow for easy monitoring of one plant with the eventual possibility of multiple at a time. These monitors include sensors for moisture, humidity, light, temperature and pH. The physical sensors themselves will be all put together into one main monitor PCB that is as low cost and low power as possible while maintaining accuracy.

These monitors will be available on a mobile app that is accessible on Android that utilizes wireless connection via WiFi or Bluetooth, depending upon where the user is located relative to their device. The companion app will also have access to a database of information regarding many different plants so that even the most uninformed plant owner can know the specifics of caring for whatever plant they may have. Although there is something that can be said about being self-informed and monitoring your plants yourself, our goal is simply to make this easier and more accessible for everyone.

2.0 Project Description

The motivation and the goals of the project were one of the first things that were nailed down when first discussing this project. During the early portion of researching, the team found a number of similar projects. These projects were analyzed, and critiqued by the team, identifying portions of the systems that were liked, as well as portions that were disliked. As the project moved forward, using these goals and motivations as a baseline, we were quickly able to move forward with developing the desired functions, requirements, and specifications of the project.

2.1 Motivation

One of the most popular hobbies in today's age, especially during and after the Covid-19 pandemic, is gardening. A large issue, however, is most people consider themselves very poor plant parents. This lack of "green thumb" could be due to the lack of time dedicated to understanding the plant's requirements. Some people decide to go to a plant store, pick out any random plant that they think looks cute, and then take it home and place it anywhere in their home. They think that all plants are cookie-cutter, that they all require the same basic needs. Although this is true to an extent, plants all have individual needs and requirements that are needed to help the plant thoroughly thrive. These needs can be as basic as the amount of water needed, to elaborate as the amount of relative humidity in the air.

The main objective for this project was to attempt to design a product that helps even the most inept plant owners be able to grow any plant they desire and have it thrive with ease. On top of this, we wanted to provide information to anyone who wanted to learn about common plants that would be grown, whether that is vegetables, herbs or just common plants you could find even outside of a garden. We believe this will give us a head over similar products that just serve to allow someone to have a basic monitor of their plants by providing education as well through the application.

We also did not want to create a system that runs itself entirely, as we want to allow the user to have some control and allow them to learn themselves. This means that we can make the device a lot more portable and manufacturable for most people, as a smaller device ends up being easier to move as well as cheaper for the average consumer. This will allow even more people to have access to the knowledge we will be providing.

2.2 Goals

The team had multiple goals in mind when trying to come up with a solution for this problem that was identified.

- Cost effective
    - The project should be cost effective, down to a point that any individual would be very inclined to purchase the item.
    - Another consideration would be to have the project be low enough cost if the team decides to try and scale the project out, meaning that with multiple units so there could be multiple units on one home monitoring different plants, as compared to having a single unit that had to be transported
    - One of the bigger expenses of the project would be some type of enclosure as it did not want the monitor to be too exposed in case water was introduced into the environment
        - 3D printing was considered as it is relatively cheap as long as the system is already setup (strictly the cost of filament)
        - 3D printing could be utilized to create a custom enclosure, in any color that a customer might want, while still being conscious about the cost associated with it

- Small in size
    - The overall design of the project is needed to be small in size, as it is meant to be used in conjunction with a pre-existing plants
        - Too large of a size would not be good for perhaps a plant with a small pot, as the monitor could be too large for the plant and pot
    - The rough desired dimensions are large enough to contain all necessary parts of the monitoring project, while also being small enough to move if necessary

- Improved plant health
    - The device should ultimately aid in the overall health of the plant. This device should in no way get in the way of plant growth but only help the user keep an extra eye on it to help keep up with the plant.
    - This device should be able to detect what a plant needs based on what the sensors read
    - The device will be able to record plant data and should be able to estimate what will need to happen next

- Ensuring success when growing plants
    - The device will act as an extra hand for the user when they are not around. This device will alert the user if something is low and needs to be done asap for the plant
    - This device will allow for the user to be away from their garden or plant for a long period of time and able to care for the plant when the device alerts them of any problems

- Help user control what the can not see
    - This device should act like a second hand to the user as well as eyes where the user can not reach, feel or see.
    - The added sensors will allow for the user to see the health level of their plant which they would not be able to see without our device.
        - pH
        - Nutrients levels
        - Light levels
        - Moisture levels within the soil
        - Temperature surrounding the plant
        - Humidity levels surrounding the plant
        - Overall plant health under and around the soil

2.3 Related Work

In terms of related products, there are a number that are already out on the market. These all vary in execution immensely, from one utilizing the idea of an “all-in-one” plant monitoring and plant caretaker system, to another using an array of smaller probes to monitor various plants from a central hub station. Some companies also offer the ability to use an entire enclosed system from which a plant can be automatically grown. These systems take advantage of an overarching concept of hydroponics, where traditional soil is replaced with nutrient enriched water and supplements.

The issue with these devices is that they are often large and require maintenance, mainly being changing of pH solutions. Although there is a significant amount of overlap in the parts used between the hydroponics systems and our project, the methodology of being small and cheap are diametrically opposed to this design and therefore our project leans toward monitoring rather than growing. This is easily observable as in a hydroponics system, they utilize tanks for liquids as well as light systems meant to regulate the amount being received by the plant. The addition of those subsystems would require a significantly large rig/case in order to house the entire system.

We are aiming to create a product that is the culmination of all of these different executions. Having a user be able to monitor multiple plants, from a mobile application, without the need of purchasing a large, expensive self-contained system. We aim to reach a target audience of plant owners that want a quick overview of their plants' health, with easy to identify vital signs that can be easily monitored and fixed by the user in a clear, discernable way.

2.3.1 Northfifteen Connected Home Plant Monitor

This device keeps tabs on your plant and allows the user to use an app while on the go. This device uses sensors to track various plant life such as water life, overall lighting, fertilizer, and temperature. This device uses bluetooth to connect to the app for the user. The main feature of this device is the water tracker for plant life that you are able to see through the app.

Figure 1 - Northfifteen Monitoring Device

Our device differs from this because we plan on adding a water reservoir to water the plant when the water level is scarce. Their device has a user friendly app which we are going for as well however ours will be able to ping the user when any of the levels are near dangerously low and the user will be able to interact with the app to know exactly what to do for the plant. Their app will make suggestions for the user, which we will be going for as well.

2.3.2 Tulip Plant Monitoring System

This device is not only one monitor, but multiple monitor probes to form a system of probes for each plant in a garden. This also comes with a handheld spectrometer that the device sends the data to by bluetooth. The spectrometer can only be used when the user is in range of the plants it is trying to measure. The user can plug the spectrometer into their computer and download the data, which in turn charges the device. This device can measure moisture levels, air temperature, fertilizer, and light through wifi signals. The user can also use an associated app to create a virtual map of their garden environment.

The difference our device will make is that there will be an app that the user does not have to be in the general area to see the status of their plant. Our device will also be able to be put into a system, if given enough time we plan on making another or multiple devices to form a system of monitors.

2.3.3 Click and Grow Smart Garden

This is an all in one planter device that you put your plant in. It is quite bulky as the plant sits on top of the potted bottom with a bar along the top. This device waters your plant for you as well as changes lighting levels and tracks the nutrients levels of the plant. This is pretty much self-growing technology because this planter does everything for you to take care of the plant. This planter requires the user to purchase the required seeds for this device which can get quite pricey and make it hard to branch out with different types. And due to the side and shape of this planter, the plant will only have a certain amount of room to grow and live.

Our device is lightweight and small to fit inside the plant's pot or even just be stuck in the garden. Our device is also going to be waterproof which this planter is definitely not. Our device also keeps the user in the loop and allows the user to take all the steps to care for the plant, it also will act for the user if needed to keep the plant healthy while the user is away or busy. We will also have an app which this device does not have one connected to.

2.3.4 EasyBloom 1000 Plant Sensor

This device is shaped like a flower to hide from plain sight near the plant. This device can measure lighting, temperature, humidity and water. The device needs to stay on for at least 24 hours to get the most accurate results of the plant per day. This device has a web interface that lets the user categorize plants by soil, color and more. This device has three different modes which include monitoring mode, recommendation mode, and water mode. Their water mode tells the user about the water level and the device needs to be removed when watering the plant.

Our device uses an app rather than an online interface which makes it easier for the user to test their plants from anywhere. Our device will also allow the user to leave our device in the plant and not have to frequently check on it. Our device also allows the user to water the plant without removing the device.

2.4 Engineering Specifications

The following table aims to show how we can take a given, or desired trait, in descriptive form, and put a tangible, executable value alongside it. As the project progresses, the team can view this table, and quickly tell if the project is moving in

the right direction by referencing the technical values along the right side of the table.

The highlighted portions, in Table 1, are the portions selected for demonstration during the final project demonstration. The team aims to demonstrate that our system can enter a sleep mode by ensuring that after a 30 second timer elapses following the last use of the system, the monitor will enter a “sleep mode”, also known as a standby mode. This mode will be activated to ensure that the system is focused on power saving.

On the attached LCD screen, the team aims to have the data values that will appear on the display update every 5 seconds, just so the customer can ensure an accurate readout, eliminating the possibility of viewing an anomaly with the sensors. The team aims to have all data transmitted from the monitoring system, to the user, in under 3 seconds. This level of speed is on the slower side, but it allows for all possible forms of communication from the system to the user to be considered. The team plans to have communication time significantly quicker than this threshold value, however.

Description	Technical Values
The system shall run at specified voltage	2.2V regulated
The system shall be rechargeable using specified method	3.7V rechargeable battery charged via 5V solar panel
The system shall enter sleep mode after specified time	≥ 30s after last use, system enters sleep mode
The system shall update monitor information on LCD/LED and database in the specified interval	≥ 1 update per 5s period
The system shall be connected to database/app through specified method	System connected via WiFi, transmits to database and subsequent user
The system shall transmit data to user within specified time after user asks/after wake	≤ 3s response time
The system shall be self-contained in specified dimensions	15cm x 15cm x 15cm

Description	Technical Values
The system's app shall run on specified OS	Application runs on Android OS
The system shall support single or multiple instances of monitoring units on different plants	≥ 1 monitoring unit

Table 1 - Requirements Specifications

2.5 House of Quality

The below house of quality, in Figure 2, was utilized by the team as a quick reference tool throughout the project development. It was developed early on as a visual representation of the requirements of the customer, in conjunction with the engineering specifications.

Viewing the figure below, the middle portion shows a relationship between the customer requirements and the engineering requirements. The legend on the bottom right shows the type of relationship as being either strong, moderate, weak, or having no relationship at all. The upper portion of the figure highlights the correlation between the various engineering requirements. As the legend reads, there can be either a positive, negative, or no correlation at all.

Correlations	
Positive	+
Negative	-
No Correlation	+

Relationships	
Strong	●
Moderate	○
Weak	▽
None	

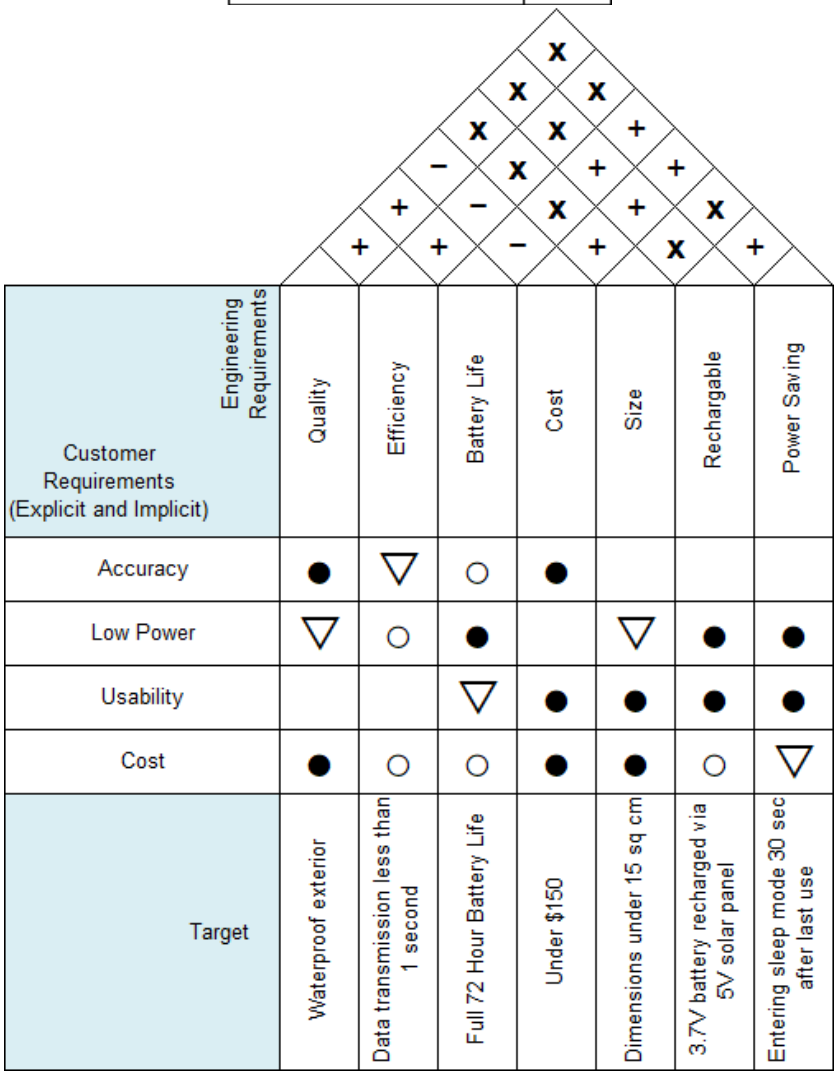

Figure 2 - House of Quality

2.6 Block Diagrams

The following diagrams, Figure 3 and 4, show the projected breakdown of all portions of development. Figure 3 shows the breakdown of all the integral portions of the hardware system. This figure includes all the major portions of the project, in terms of the various sensors, displays, microcontrollers, and the communication of that data to the customer via the software network. Figure 4 shows the other side of that system, the software side. The figure shows the projected step-by-step of the data, and how it will be analyzed, and dealt with, when the data comes in from the hardware side.

2.6.1 Hardware

The following hardware diagram in Figure 3 demonstrates the various devices that interact with the microcontroller, such as the sensors. The arrows represent that data is being transmitted from the relevant sensors to the microcontroller, and the transfer of said data from the microcontroller using WiFI or some other form of wireless communication. Each part is color coded in order to show which team member is responsible for the corresponding part.

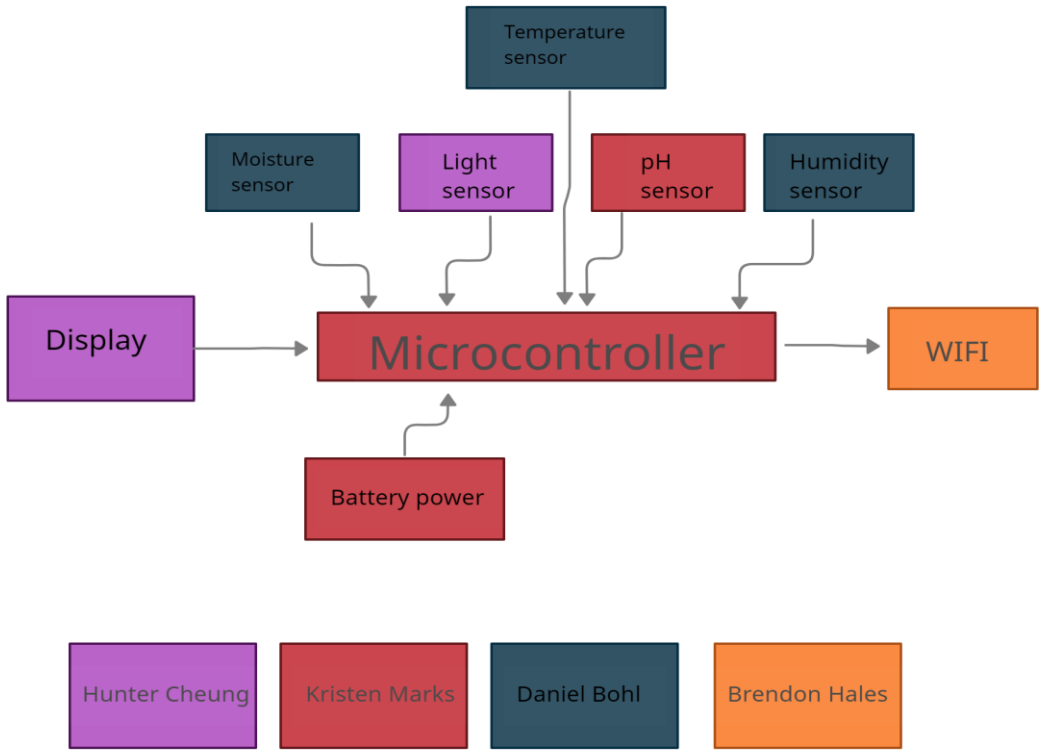

Figure 3 - Hardware Block Diagram

2.6.2 Software

The following diagram is representative of how the software in this project is intended to operate. Figure 4 shows how the system will deal with the data that is sent from the sensors, and where the information shall end up once taken into the software side. Code will be written for the microcontroller in order to handle the sleep/wake up functionality as well as process and send the sensor data to the database and display. The chosen market for this will be android rather than iOS and it will be done with java and include push notifications for the user to be told of necessary actions to be taken.

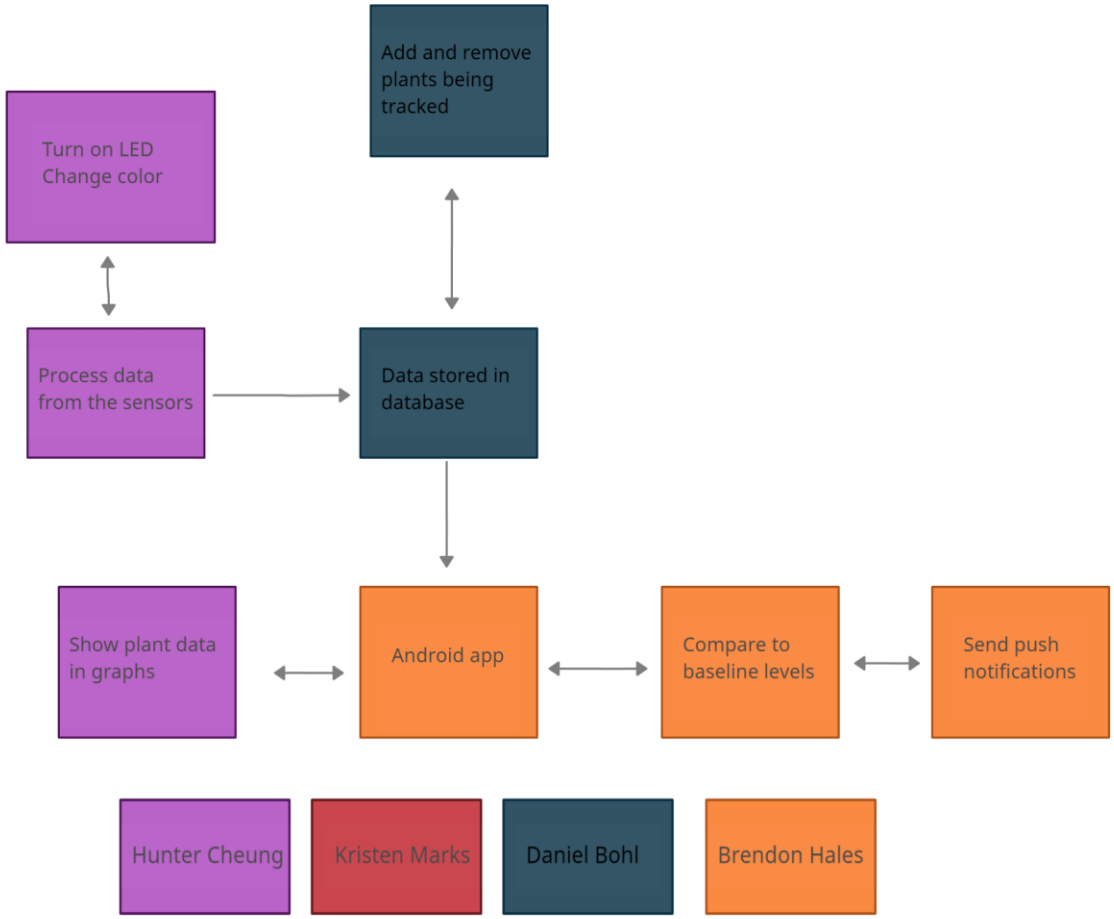

Figure 4 - Software Block Diagram

2.7 Project Operation Manual

The SmartPlant Hub is designed to be an all-in-one plant monitor that is portable and easy to set up and move if needed. There is no need to set up any of the sensors, as they are all held within the housing, even if some are slightly out of the main casing as needed. The only attachable component to this setup is the optional solar panel that can be used as a charging component to extend the usage of the battery pack as a whole.

Setting the hub up is fairly simple, ensure that the probes at the bottom are placed in the soil close to the plant you wish to monitor. Then, make sure the power switch is on and the casing is secured and not loose. After this, if desired, connect the

solar panel to the cord designated for it and ensure it is secured in its spot on the casing. Face the panel in the direction that gets the most sunlight.

After the physical casing is set up and secured, you can download the app off of whatever store you use and get to setting up the device on your account if you would like to access it remotely. Otherwise, you can simply connect via Bluetooth if you're close enough. Now just navigate the app to your plant and view the monitors and sensors data.

One of the best things available to you as the owner of the SmartPlant Hub is the in-app database.

Outside of the app, you will have access to a screen on the outside of the casing with a button nearby. Pressing this button will allow you to view basic data on the screen relative to the readings of the sensors. If you do not use the screen, you can also view basic data via the LED lights on the outside of the casing. Each will be labeled and will glow green or red depending upon how the sensors are reading, and you can use this to adjust accordingly.

3.0 Project Research

This section allows for in depth research, comparison, and final selection for all of the various parts for the project. Below covers the main portions of the project: enclosures, microcontrollers, sensors, power sources, LEDs, and LCDs. Price, sizing, availability, and overall capabilities were taken into consideration when making final decisions.

3.1 3D Printing

Due to the fact that we have multiple sensors, an LCD screen, battery, and multiple boards it is necessary for us to have a custom casing. Our solution to this issue is by 3D printing a housing case for all of our hardware and power systems.

3D printing has been a staple in engineering for decades now, the first documented mention of the concept occurred in 1945 by Murray Leinster in a short story. It would finally be realized less than three decades later in 1971, when Johannes F Gottwald patented the first 3D printing device for prototyping and manufacturing of patterns. The 1980s saw the largest advancements in the technology as three major patents were filed which developed the stereolithography process, FDM printers, and single nozzle inkjets for metal casting.

The ability to 3D print personal projects has become more and more accessible in the last 5 years, at an accelerated rate. Prior to 2018 there were few offerings for consumers to 3D print at home, and of the available options they were incredibly expensive, often thousands of dollars for an FDM printer. In 2022 you can get an FDM printer for as cheap as $200 and with the low cost of filaments, it is incredibly accessible for many people. There is also the ability to have a design printed by a company with the price varying depending on the material, but this option includes extended periods of time where there is travel time on top of build time.

3.1.1 3D Printer Choice

Our printer of choice is the Creality Ender 3 pro, an FDM printer that is incredibly affordable and offers great performance for the price. It boasts a 220 x 220 x 250mm magnetic bed which gives us the ability to print quite large pieces and in the case of our device, perfect size for plants considering our need for the device to be small. The printing precision is +-0.1mm meaning that if necessary we can be incredibly specific when designing the sensor casings.

3.1.2 Filament

When it comes to 3D printing, the printer is only half of the battle, using the term FDM to describe the ender 3, this refers to the way in which the printing process works.

FDM stands for Fused Deposition Modeling, in this the nozzle of the printer extrudes heated plastic filament and builds multiple layers from the bottom up in order to create the desired object. It is vital that the bed is leveled prior to starting the printing process as if not, the print may be compromised as slight offsets result in layers becoming increasingly off target. After the filament is layered, it dries and hardens into a much sturdier final product.

SLA is the other printing process that is often available for consumers, this is Stereolithography Apparatus. Similar to FDM, the material is added on in layers but it comes in a liquid form and for SLA, is mostly resin. The liquid is hardened not by the result of hardening post heating up but rather is hardened by focusing light on it, almost always a laser. The main distinction here is that SLA allows for slightly more precise prints as it is done in an enclosed space due to the toxicity of liquid resin but lacks the durability of thermoplastics from FDM.

3.1.2.1 ABS Filament

Acrylonitrile butadiene styrene (ABS) is one of the most common filaments used in 3D printing. There is already standard use of ABS as it is a popular plastic due to the impact resistance, strength, and rigidity compared to something like resin. The temperature at which it becomes pliable is 220 degrees C, something which is not realistic from standard temperatures outside. Due to the cheap price and heat performance, many manufacturers will print prototypes in ABS to test products that do not need a high number of iterations.

In terms of reliability for our project's use, ABS would be a poor fit simply due to the fact that it does not hold up well to UV light. With our product designed to sit outside in the sun for days on end, we need a filament that can hold up to not only outside temperature, but also just UV light as a whole.

3.1.2.2 PLA Filament

Polylactic acid, PLA, is the second most popular thermoplastic used in 3D printing but is also incredibly popular in different fields due to its degradability. The filament is made entirely from renewable resources and because of this it is recyclable, making this filament future proof as we move toward renewable resources in general. One such application of PLA is medical implants due to the fact that it degrades into lactic acid. By degrading over time it helps the body to heal slowly over time similar to how stitches assist cuts. When compared to ABS, PLA has a far higher tensile strength at about 7,000 PSI compared to about 4,000 PSI of ABS.

Due to the increased strength, PLA is incredible for prototyping projects which need stress testing. Compared to ABS, the temperature at which PLA becomes pliable is much lower, melting at about 145-160 degrees celsius depending on the manufacturer. Despite it being significantly lower, both are suitable for our application as it is very unlikely to reach either temperature.

3.1.2.3 PETG Filament

PETG filament, polyethylene terephthalate glycol, is considered to be a mixture of ABS/PLA as it exhibits the durability of ABS and the printability of PLA. The glycol, which is the main distinguishing factor from PLA, the most similar filament, offers extra protection against corrosion which is incredibly useful for our project. Soil with high acidic levels or chemicals that could possibly impact the plastic would be less of an issue with PETG filament. Due to the increased strength of PETG, PLA can be foregone as the device housing will be able to withstand more mishandling by a potential user, if they were to drop the device on the ground. Typically ABS will crack after it has hardened in a similar situation or PLA would have similar issues with deformity as a result.

3.1.2.4 Filament Choice

This choice was obvious in the context of picking a filament for our project in general. PETG is our choice for multiple reasons: durability, stronger adhesion and resistance. Since our product needs to sit outside in varying weather conditions we need something that is durable enough to stay together through all of this. Sitting in sunlight especially has an impact on the material itself, and PETG stands up to sunlight the best. Also, another important aspect is being able to withstand rain and PETG is not only the most resistant to moisture, but also the easiest to waterproof in terms of the device as a whole.

Filament	ABS	PLA	PETG
Printability	Hard (Needs enclosure/filtering)	Easy	Hard
Print Fumes?	Yes	A bit	No
UV Resistant?	Yes	No	Yes
Withstand high	Yes	No	Yes

Filament	ABS	PLA	PETG
temperatures			
Water resistant?	Yes	No	Yes
Sturdy	Yes	No	Yes
Price	$21/kg	$21/kg	$24/kg

Table 2 - Filament Comparison

3.2 Microcontroller

A microcontroller is a small computer that is considered to be an all-in-one system. They, at the most basic level, contain a central processing unit (CPU), some form of memory, as well as input and output segments. As they are so small and compact, they are cheap to manufacture, as well as introduce into projects. Throughout the years, these have grown in popularity now allowing for embedded systems to use them as the basis for all computational needs.

Being the brains of the project, the microcontroller choice is one of the most vital portions of any project. In today's age, there are a multitude of projects that all utilize different boards, sensors, motors, etc. to achieve the same end goal. This device will need to handle multiple sensors and also have wireless connection integration so that the data can be transferred to the mobile application. On top of this, we will want to utilize a microcontroller that has good power options and supports rechargeable power sources as well.

Making sure the microcontroller we select matches up perfectly with our goals is essential in the creation of this device. With this in mind, the choice of microcontroller is probably the most important choice to consider, as not having any of these vital components could change the device entirely.

3.2.1 Arduino

Arduino is the one of the most common choices when it comes to project microcontrollers. This is due to the wide availability of the boards, from a number of distributors, while also having the perks of having both software and hardware capabilities built in. These boards are serviced using the built in API called the

Arduino Language, which is very similar to C and C++. Arduino is typically considered the more lightweight choice of microcontrollers, as it does not have a strong enough CPU to handle complex, intricate computations. Arduino's are used in systems where repetitive, non-computationally intense actions are made, which means that they are used in simple systems.

Arduino is open-source, both in terms of hardware, and software, making them easy to find documentation on, as well as modify for your own needs. One such way to modify an Arduino is by utilizing a shield. A shield is typically a printed circuit board that is made compatible with a pre-existing Arduino using the header pins. These shields allow for many various applications, such as acting as a motor controller, to holding navigational equipment, to just containing a breadboard allowing for "on-the-board" prototyping.

The software of Arduino is written utilizing the built-in integrated development environment (IDE). This IDE serves as a single place for the developer to write the code, debug any errors, and then flash the code to the microcontroller for testing or execution. The IDE supports languages such as C and C++, allowing for a developer who has no prior experience with Arduino to be able to write a program quickly and efficiently. As the developer becomes more and more familiar with the system, there are a number of libraries that are available to import into a project, allowing for a vast number of applications. These libraries allow for a wide ranging number of projects to be developed, ranging from a fully functioning cube satellite to a handheld game console. With the open source capability of the Arduino, developers can utilize both hardware modifications as well as software, to build just about any project using an Arduino as the base.

3.2.1.1 Arduino MKR1000

The Arduino MKR1000 is a perfect merge between the computational power of the Arduino Zero and additional low power wireless (WiFi and Bluetooth) capabilities. This board comes in at a relatively small, rectangular form factor of 2.42 inches by 0.98 inches (61.5 millimeters by 25 millimeters). The MKR1000 has a Li-Po charging circuit built-in, allowing the system to run off of either a battery, or an external 5V power source. The circuit also lets an internal battery be charged while being plugged into an external power source. The price of this system comes in around $37 USD, but with the capabilities built in, makes it a valid overall choice.

3.2.1.2 Arduino Nano 33 IoT

The Arduino Nano 33 IoT is the smallest form factor of all of the boards that were researched. This rectangular board comes in the miniscule size of 1.77 inches by 0.77 inches (45 millimeters by 18 millimeters). The Nano 33 has built in Bluetooth and WiFi capabilities, and an always-on gyroscope and accelerometer. The

downside of this particular board is that it solely operates on a 3.3V circuit, rather than a 5V like the vast majority of other boards, making the Nano's power supply a very important factor to consider when moving forward with development. The price of this system is about $19 USD but could possibly be found cheaper. In terms of price to basic performance, this is by far the best choice, but may result in the rest of the system cost more due to the 3.3V input voltage restricitons.

3.2.1.3 Arduino MKR WiFi 1010

The Arduino MKR WiFi 1010 is considered the professional version of the Arduino Nano 33 IoT. Adding in the more traditional 5V power capabilities to the already more-than-capable Nano, makes this board an extremely solid contender. The MKR 1010 comes in at a relatively small, rectangular form factor of 2.42 inches by 0.98 inches (61.5 millimeters by 25 millimeters). This board utilizes a Li-Po charging circuit, allowing the system to run off of either a battery, or an external 5V power source, while also charging a LiPo battery while hooked up to the 5V source. The price is $35 USD, making it the cheaper choice than the MKR 1000, with even more functionality.

3.2.2 Raspberry Pi

The Raspberry Pi was originally developed to be a small, lightweight computer for students to learn basic computer skills. As the system grew in popularity, so did the overall functionality of the system, with improvements to the CPU, memory, as well as the number of peripherals attached. Now, in today's age, it is usually regarded as being a lightweight computer replacement. Having more peripheral adaptability built in from the factory, makes these the ideal choice for developers and hobbyists that do not want to try and make additional parts to be compatible with the board. The Raspberry Pi is very much plug-and-play in terms of not needing additional features. Using pre-existing standards for interfaces, such as WiFi, or USB, the Pi can communicate with any number of additional devices, making it the perfect all-in-one system for projects.

As the Raspberry Pi has been developed over the years, the computational power has grown significantly. The first iteration of the CPU was already faster than the Arduino boards that were available on the market, As the system grew in popularity, the development capabilities of the factory grew, allowing for faster CPUs to be integrated into the system. Now, the CPU integrated within them allows for complex calculations to be performed, very similar to a full size computer.

In line with the CPU growing stronger, this allowed for more peripherals to be added. The more that were added, the less additional functionality was desired, as it was already contained within the board from the factory.This caused the board to grow in popularity even more with the hobby community, due to the ease of use that was associated with implementing a Raspberry Pi into your system. Now, in

today's age, the Pi is considered to be the de facto standard when it comes to a project that needs lots of computational power in a small form factor.

3.2.2.1 Raspberry Pi Zero W

The Raspberry Pi Zero W is the original board Zero family. They come with a small form factor of 2.55 inches by 1.18 inches (65 millimeters by 30 millimeters). Although it is the older version of the Zero board family, it is just as functional as the newer version, albeit with a slower processing speed. The power supply on board utilizes a 5V circuit, but also has on board WiFi capabilities. The stock of the base board, priced at $10, is very slim, but there is stock limited to the ultimate starter kit, priced at $45.

3.2.2.2 Raspberry Pi Pico

The Raspberry Pi Pico is by far the cheapest board, $4 for a version without the headers pre-installed, or $6 for the version with headers installed. One of the biggest downfalls of this board is there is no on-board WiFi module, requiring an additional $20 WiFI module. There is a WiFi hat that has a built in 1 inch LCD screen, potentially bypassing the need to purchase a separate display. This system also has an on board temperature sensor, although it seems as though it reads ambient temperature only.

3.2.2.3 Raspberry Pi Zero 2 W

The Raspberry Pi Zero 2 W is the successor to the original Zero board. The Zero 2 has the same exact form factor as the original, with just a new faster processor. This new processor runs at an incredible 1 GHz and is also one of the only 64-bit microprocessors that was considered. The power supply on board utilizes a 5V circuit, but also has on board WiFi capabilities. The stock of the base board, priced at $15, seems to be non-existent, but there is limited stock of the ultimate starter kit, priced at $55.

3.2.3 ESP32

ESP32 is a set or series of low power microcontroller chips developed by Espressif Systems. Not very well known, Espressif Systems is a semiconductor manufacturing company established in 2008. Their headquarters are located in Shanghai and they are responsible for the ESP line of microcontrollers. One of the staples of ESP32 is that it is generally low cost compared to other microcontrollers with comparable specifications. ESP32 is also fairly new, as a successor to the ESP8266, it was released in 2016 and has been fairly popular since then. Also

since the release of ESP32, many variants have been created all with differing capabilities but still falling under the same SDK as the original ESP32.

3.2.3.1 ESP32 Feather Board - HUZZAH32

The HUZZAH32 is a Feather Board based on ESP32 boards and developed by Adafruit Industries. Similar to other boards, the HUZZAH32 commands seemingly comparable hardware and support while exceeding others in a lot of areas. it is a small form factor at 2.0 x 0.9 x 0.7 inches. It has an integrated WiFi transceiver as well as dual mode Bluetooth that can run classic Bluetooth and Bluetooth Low Energy. 4MB flash in its WROOM32 module and an onboard PCB antenna. It also allows for micro-USB input and has a Lithium-Ion charger built in that automatically steps voltage up or down. On top of all this, it can work with the Arduino IDE so we can take advantage of I2C, SPI and analog reading from the sensors we are using. To finish, it also boasts a price point of $21.95, which is a good price for the product.

3.2.4 Microcontroller comparison

Given our seven choices, we can then narrow the list down to a select few and compare them. Our project requires us to use a board that is able to handle many sensors and transfer the data via WiFi and Bluetooth. On top of this, choosing a board that is compatible with easily accessible and supported software is important as well. Arduino seems to be the clear choice when it comes to easily accessible and well-known and maintained software.

We will be narrowing this list down to the three best boards that meet these requirements at a minimum. We can eliminate the Arduino MKR1000 simply due to the existence of the Arduino MKR WiFi 1010. This board matches or exceeds the specifications and capabilities of the MKR1000 at a lower price point. We can also eliminate the Arduino Nano 33 IoT as this board, even with its good price to performance ratio, does not match up to the full capabilities of what we need.

Another board we can eliminate is the Raspberry Pi Pico due to the fact that it does not have an easy to configure charging circuit or WiFi module built in. Finally, we can eliminate the Raspberry Pi Zero W because of the existence of the Zero 2 W which has all of the same specifications but with a better processor. This means we narrowed the list down to the Arduino MKR WiFi 1010, the Raspberry Pi Zero 2 W and the HUZZAH32 Feather Board.

Constructing a table to compare the specifications of each board will allow us to more easily compare them and make a final decision.

Microprocessor	Arduino MKR WiFi 1010	Raspberry Pi Zero 2 W	HUZZAH32
Size	61.5mm x 25mm	65mm x 30mm	51mm x 23mm
Interface	32-bit	64-bit	32-bit
Pin Count	23	40	21
CPU Speed	240 MHz	1 GHz	240 MHz
Memory	256 KB	512 MB	4 MB
Communication	1 x UART, 1 x SPI, 1 x I2C, 7 x ADC, 1 x DAC, 8 x Digital I/O	2 x UART, 11 x SPI, 4 x I2C	3 x UART, 3 x SPI, 2 x I2C, 12 x ADC, 2 x I2S, 2 x DAC
Operating Voltage	3.3V	5V	3.3V
WiFi	Yes	Yes	Yes
Bluetooth (BLE)	Yes	Yes	Yes
Price	$35	$55	$21.95

Table 3 - Microcontroller Comparison Table

3.2.5 Microcontroller selection

The choice here ended up being the HUZZAH32. Although the others had comparable hardware and capabilities, if not better, this board had the best price to performance ratio and will allow us to take advantage of Arduino's software and connections on a cheaper comparable board. The biggest contender was the Raspberry Pi Zero 2 W but there is a large issue of lacking availability. We would need to purchase the full kit in order to get the board itself, and one of our considerations is manufacturability. Having a main component be hard to find means that this area is heavily restricted. On top of this, the Raspberry Pi Zero 2

W draws much more power than any of the other boards due to its build, which is entirely unnecessary. Also, although the Zero 2 W boasts such great specifications, we do not need specifications that high for the product we are creating, so going with a board half the price that will still work is the best option. This is a clear example of the impact of marketability. Price: $21.95

Figure 5 - ESP32 Feather Board HUZZAH32

3.3 Wireless Communication

This section works to break down the intricacies related to the wireless communication methods related to the project. Starting with a general overview of the options, followed by an in depth analysis of the possible means of communication, all major avenues will be discussed and analyzed in order for an effective decision to be made in regards to the most ideal choice.

3.3.1 Overview

Wireless communication is essential for products entering today's world. Almost everyone should have access to the Internet and wireless communication, as it is one of the most significant technological advancements in the last century. Transferring data via wire is still faster in many instances, but with the consistently expanding wireless technology, wireless speeds are going up considerably. Having to transfer data wired is a lot more cumbersome, however, especially when considering how small the device will be as well as the fact that it will be transferring data to a smart device. This device also needs to transfer data to mobile devices that are far away, which means that WiFi will be essential to the data transfer of this product.

However, what if the user is in their house while the sensor is in their backyard, not very far away? Well, Bluetooth is a better option for this short range. Looking

at Bluetooth, normal Bluetooth would not suit this product, as it is meant for large and fast data transfer. However, this device will only be transferring small packets of data from sensors on the board, meaning regular Bluetooth would be pointless since WiFi might as well be used for that purpose. One bit of technology, though, exists to make Bluetooth a bit more viable for small data transfer, and that is Bluetooth Low Energy. This version uses a lot less power and transfers small bits of data, mainly meant to be used in small IoT projects. This means that BLE is a contender for this device compared to WiFi, and they both need to be weighed against each other relative to this device.

3.3.2 Radio Frequency (RF) Communication

Radio Frequency (RF) is the measurement of oscillation of electromagnetic radio waves. RF is the integral base of nearly all wireless communication that we use today. The unit of measurement for RF is Hertz (Hz) which is the number of oscillations or cycles per second. Frequency in (Hz) is represented in an inverse relationship with the wavelength of the waves in the form f = c / λ where f is frequency in (Hz), c is the speed of light which is a constant value of roughly 3.0 x 10 m/s, and λ is wavelength in meters (m).

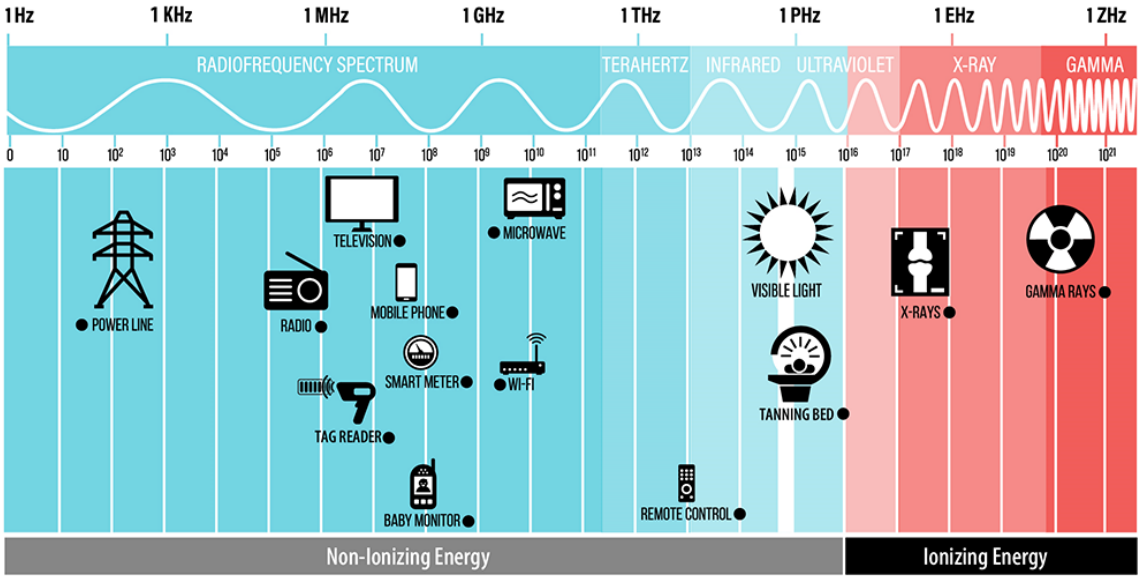

Figure 6 - Electromagnetic Spectrum Chart

The Radio Frequency spectrum falls under the electromagnetic spectrum in the range 100 kHz to 300 GHz. All of our electronics and appliances that use RF are within this range. The length of the wave has a lot to do with its capabilities in terms of distance as well. Longer wavelengths on the lower end of the spectrum are able to travel much further distances than the short length waves on the other end. The

chart below shows the electromagnetic spectrum and common electronics or appliances that run at specific frequencies.

The SmartPlant Hub will utilize both WiFi and Bluetooth, and both operate on this spectrum. Bluetooth, even Bluetooth Low Energy, works on a frequency of 2.4 GHz while WiFi works on a frequency of 2.4 GHz as well as 5 GHz. Following our rules for wavelength, 2.4 GHz tends to allow for a larger range connection compared to 5 GHz, but at a slower speed.

3.3.3 WiFi

It is fair to say that most people living in the United States and most of the world know what WiFi is at the base level. Based on the IEEE 802.11 standards, WiFi is essentially a group of network protocols that allows for connection between multiple devices on a single network. This connection allows multiple digital devices to connect with one another on a Local Area Network (LAN) and transfer data between each other utilizing radio waves. WiFi is nearly synonymous in society with the word Internet in basic terms nowadays. It is a crucial aspect of our wireless communication, as it helps connect all of our devices worldwide. WiFi routers are used in public or private places in order to act as an access point for wireless devices to connect to and transfer information.

The reason for choosing WiFi for this project is the fact that we want to be able to connect to a cloud server that receives data from the device. The mobile application we will develop will connect to that database and receive the data as well, sending it to the user on their mobile device. WiFi will allow this to happen even if the user is not nearby, and with WiFi being almost essential in many households nowadays, it was a clear choice for gathering and transmitting data from the device.

Our project will be utilizing WiFi, meaning the user will need to have an access point within range that the device can communicate with and connect to in order to transfer data to the server we will be using. There are common issues present with WiFi that exist for nearly all radio wave communication, mostly including interference. Anything between the access point and device connecting to that access point will cause interference including walls or other appliances. Even with this in mind, the range value of WiFi is still a large benefit.

Some of the benefits of utilizing WiFi to transfer data from our sensors/board in general is the range and speed. WiFi tends to be able to reach further distances and transfers data at a much faster rate than BLE. However, WiFi tends to have less privacy than BLE. On top of this, the part of the microprocessor that works with the WiFi utilizes considerably more power than BLE when it comes to transferring the data. Although, we will not need to have consistent use of this part, as it is possible to send pings at intervals that checks if the user would like to

connect to the device or not and then activate or sleep the module based on the response.

The biggest benefit for this device that WiFi provides is the distance aspect. A user can not connect via BLE to the device if they are away from home, however, using WiFi allows the user to get the data from anywhere they like. Although it is a bit more cumbersome in terms of power consumption, this feature is essential and WiFi is the only one that can do the job and is readily available in most places that a user would be utilizing this device.

3.2.4 Bluetooth

Bluetooth has been around for over two decades, since 1999 when a wireless voice headset was released that had bluetooth. Just like WiFi, Bluetooth transfers data via radio waves in the Ultra High Frequency (UHF) range. Specifically, Bluetooth operates at the 2.4 GHz frequency and is meant for short range transfer of data between devices. Usually, instead of using a router like WiFi uses for a hub, Bluetooth is meant for shorter range connection between devices. One of the most common examples that has existed for a long time is the connection between our devices like cell phones and either speakers or headphones to listen to the audio that would normally come out of the speakers on your device.

Bluetooth uses Personal Area Networks (PAN) as opposed to Local Area Networks (LAN) which is a network that connects wireless devices in an individual's space. Bluetooth is mainly used for small spaces to connect short range devices. This means that Bluetooth is a lot more energy efficient than WiFi, sacrificing range and data transfer rate.

Bluetooth has come a long way since its initial creation, now being much faster and gaining more connection range while still maintaining its lower power consumption. However, when it comes to devices that need to transmit very small packets of data like those from sensors, it becomes a bit over the top to utilize either WiFi or Bluetooth, especially when Bluetooth has a different subtype that is meant for small devices.

Our device will have a microprocessor that already has a built-in module that can use Bluetooth. In the instances where the connecting mobile device is within range of the device, it can use Bluetooth to transfer data so that it can run at a lower power consumption compared to WiFi. Lowering this cost even more will be a good idea as well, and can be achieved via Bluetooth Low Energy (BLE).

3.3.4.1 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) is a wireless personal area network available past Android 4.3 used to create short connections that can transfer small bursts of data. It was created in 2006 under the name Wibree by Nokia but became BLE in 2010 after being marketed as Bluetooth Smart. The reason for creation was specifically for small devices that could be powered with a button battery for even years at a time with such low power consumption while maintaining the transfer range. BLE is different from normal Bluetooth, Bluetooth is used to transfer a lot of data quickly but uses a lot more power, while BLE was developed for specifically small packet data transfer, a seemingly perfect solution for small IoT projects such as this one.

BLE, while being more secure, can also send small packets of data such as those from sensors a lot more efficiently than WiFi over smaller distances. BLE is also a lot more energy efficient and uses less power overall. This alone means our device will use less power when transferring data, which is arguably the most power-consuming aspect of the device as a whole.

3.3.5 WiFi vs BLE Selection

If this product were intended to send data locally to a device without consideration of speed, BLE would be the optimal choice for its security and low power consumption. However, this product almost requires the ability to send real time data quickly to a mobile device from anywhere. This means that WiFi is the clear option, considering the user should be able to access this data from anywhere, not just in a small radius around the device.

The final consideration made was the idea of swapping between the two. Utilizing WiFi for users who are physically further away from the device and BLE when they are close. This would be the most efficient option and would allow us to utilize the benefits of both, so this is our choice.

3.3.6 Structure of the System

The device will be in a casing with all sensors correctly placed. Temperature and humidity will be near outside the casing to monitor ambient conditions. The light sensor will have access to the ambient light outside the casing to get the same amount of light that the plant is getting. The moisture sensors will sit planted in the soil as close as possible to the plant to monitor water levels.

The microcontroller will be powered by a battery pack with an optional solar panel connection, which will also sit outside the casing with access to sunlight if needed. All sensors will be connected to our chosen microcontroller which will then transmit data via WiFi to a server if no bluetooth connection is found, or BLE if there is one possible. The user will then be able to view that data within the mobile application.

There will also be a few LEDs that display simple readings from the plant, and generally give the user an idea of if there is an issue with one of the readings. On top of this, there will be a bit more detailed information given on the LCD display. The LCD will be toggleable via a button on the device and will only activate for a short time to give the user a quick reading of the sensors without the use of wireless communication.

3.4 Software Technologies

This section will highlight possible avenues in regards to all portions of the technology execution. For the monitoring project, as 3D printing is planned to be used, 3D modeling software must be considered. In addition to this model, the team must decide on a model slicer, in order for the 3D model to be broken down so that the 3D printer knows what portions to print, in what orientations, and in what order. As a mobile application is desired, the platform of which the application will be released has to be considered, as to whether or not Android or iOS is the platform of choice. The mobile application is not the only code-related portion, however. In order for the sensor interface to be understood by the Arduino microcontroller, a code environment that is compatible with Arduino's must be decided.

3.4.1 3D Modeling

In order to 3D print the casing for our hardware, we must have a design realized in an STL file. STL files were first released in 1987 and developed by 3D Systems, a company which provides multiple services related to 3D printing, including printers, materials, and scanners. Unlike CAD models which contain attributes such as texture and color, an STL file only refers to the surface geometry of an object. Also STL can not represent circular edges and therefore does so by approximating with triangular planes. In order to construct the STL file there are multiple softwares which need to be used including AutoCAD/TinkerCAD and Ultimaker Cura. The workflow for 3D printing an object from scratch goes from designing a model, splicing the model, then printing the object.

3.4.1.1 AutoCAD

AutoCAD is an industry standard software and has varied uses outside of just 3D modeling for printing. Architects as well as mechanical engineers have been using AutoCAD in order to make precise and complex models for their designs for many years now.

3.4.1.2 TinkerCAD

A simpler alternative to AutoCAD is Tinkercad, a browser-based CAD which is free and is far easier to navigate. It includes multiple shapes/objects that are combined in order to create something much more complex. Considering that our design is not too complex and simply needs to house the necessary sensors/components, Tinkercad should meet it easily. The CAD software creates the model for our housing but that is not enough to actually 3D print from.

3.4.1.3 SolidWorks

Solidworks is a similar design to AutoCAD as it is a computer aided design which is more in depth than something such as TinkerCAD. This is the design program which we have had the most use with, as solidworks was used in previous courses at UCF and taught at length. It has sold over 3.5 million licenses since its release and utilizes a parametric feature based approach as opposed to AutoCAD. There is ease of access as we are provided access to solidworks as UCF students and the feature rich software makes it appealing to fulfill our requirements.

3.4.1.4 Ultimaker Cura

Ultimaker cura is a splicing software that is open source, beginner friendly, and allows for integration with our 3D printer. Due to the open source nature, there are a multitude of community made profiles that can help to simplify the setting choices for different filaments. There is a large amount of variation between the temps, print speed, and thickness. A thicker print will result in a stronger print but result in a drastically longer print time. Attempting to increase print speed will critically affect the accuracy and structural integrity of the final product. The printing temperature will vary depending on the filament type and will even vary depending on the manufacturer of that type of filament. The cura software works by taking the CAD model and allowing you to “splice” the model into a combination of triangular planes which are printable.

3.4.2 Mobile Application

There are multitudes of plant monitoring apps on the market as of today. The vast majority of these apps however are intended for the users to essentially make a calendar with their plant(s) and track its needs with reminders. Such apps include Planta, an app where you can add your own plants and get told through reminders when you should water/care depending on the plant and location you live. This process is good but lacks in comparison to what our project intends to do. Rather

than going off of averages for plants, we intend to create a device that lets users know the exact diagnosis of their plant. This eliminates any guess work or errors that can occur due to lack of lighting or irregularities in weather that something like Planta could not account for. Planta offers in-app purchases to their users and seeing that they have over 7 million plants monitored in their app, it is safe to assume they are seeing large profits over it.

Our project is relatively cheap compared to other devices that aim to deliver a monitoring device and the app would feature no in-app purchases, all functionality is available upon ownership of the device.

3.4.2.1 Mobile Application Operating System

The operating platform, when choosing to create a mobile application, can be daunting, but regardless, an important decision. When choosing how to develop the app there is a very important design decision to be made which is whether to support both iOS and Android or only one. Both have significant pros and cons when determining which platform we want to develop for.

3.4.2.1.1 iOS

To develop an iOS app the developers must be running on macOS, use Xcode which is the development environment for macOS, and have a developer account which costs $99 annually. Xcode was initially released in 2003 and has since been the staple for their app development, it supports Swift and Objective-C as their standard languages. Swift is a language which functions similar to that of python, where there is less syntax necessary and is considered to be a functional programming language. One large difference is the use of type inference, meaning that when you declare variables you do not need to determine its type as it will be inferred by the language depending on what is being read/written. With it being a less structured language, it is undoubtedly easier to code in than something like Objective-C which is merely an extension of C language. After finishing an app, it must be submitted for an extensive review by Apple and it is necessary for the program to run on the latest iOS version otherwise it can not be accepted.

3.4.2.1.2 Android

Android offers developers the ability to use the android software development kit and use languages such as Java and C++. The development environment offered is called Android Studio and can be run on any OS unlike Apple’s Xcode. Android studio also supports an emulator which allows multiple android phones to be tested in order to ensure the app is correctly adjusting to varying sizes of devices. Unlike the iOS counterpart, the version for which the app is supported does not have to

be the most recent, there is also no need to pay for a developer account when submitting an app to the android market.

3.4.2.1.3 Operating System Comparison

The two most prevalent operating systems when it comes to mobile application development are iOS and Android. Despite iOS being the most prevalent operating system throughout the world, Android tends to be the most popular. This popularity can most easily be seen when viewing the table below. The comparison shown compares the development language options that are available, the possible development environments, the end market, as well as the cost associated with developing an app.

Language options for iOS are limited to Swift, a language made specifically for iOS app development, whereas Android has more compatible languages such as Java and C++, allowing for a developer to easily migrate from other projects to creating an Android app. When considering development environments, a similar story arises. For iOS, Xcode is the exclusive app development environment, whereas Android has much more flexibility, although the most popular is the Android Studio due to the ease of integration with various phone types and sizes. Cost of development also comes into question when considering the operating system for development. Because iOS requires the use of Xcode for any development, a user is required to pay the $99 annual subscription fee to have full access to the development environment. Alternatively, Android has free, extremely popular options for development such as Android Studio. This free version allows for any individual regardless of financial status to begin to create apps for Android, most likely helping build its popularity for smaller app developers. Last but not least, the end marketplace needs to be considered. Although initially, it is free to upload an app to the Apple Store for iOS and the Google Play Store for Android, there is a large difference when it comes to the upload experience. iOS requires an extensive compatibility test, as well as app safety review to be performed by a 3rd party prior to release, whereas the Google Play Store does not have any of these requirements. The lack of requirements result in a bad, non-compatible app to be uploaded, but allows for any aspiring app developer to upload their app, with ease and speed.

Name	Language Options	Development Environments	Market	Cost of App Development
iOS	Swift	Xcode	Apple Store	$99 per year
Android	Java/ C++	Android Studio	Google Play Store	Free

Table 4 - Operating System Comparison Table

3.4.2.1.3 Operating System Choice

From this knowledge of app development in both environments, it becomes a question of which we are most comfortable in using for our project. Considering that iOS uses swift and objective-c, languages none of us have used in the past, the choice becomes much clearer. Our previous knowledge of using java in multitudes of classes makes the android development path much more desirable, especially with the ability to emulate multiple devices to ensure it works correctly. There is also no need to make a developer account and pay an annual fee should we choose to push the app to market. Taking all of this into consideration, the production team decides to move forward with making an application for Android, forgoing the decision to create an iOS application. This is due to the ease of emulation in terms of software testing, as well as the public release of the application for users on platforms such as Google Play Store.

3.4.2.2 Hardware Code Environment

In order to create a functioning device, we must use a compatible development environment for our microcontroller, the HUZZAH32. When considering possible development environments, we are going to focus primarily on environments that we have experience with prior.

3.4.2.2.1 Code Composer Studio

In previous projects at UCF, we have written code for physical boards using Code Composer Studio, this is because we have mostly worked with Texas Instrument boards. This environment only supports the C language, and would not be compatible with the HUZZAH32 considering that it is an Arduino board. Compared to our previous work in Code Composer Studio, we will not have to work in assembly code and lower level things such as registers and allocation will be irrelevant to our workflow. This makes our development process far simpler as there is much less to worry about in terms of lower level development and the most important thing is our logic.

3.4.2.2.2 Arduino IDE

The Arduino IDE is the standard development environment for all Arduino coding. The environment has built in support for hundreds of Arduino boards, with the ability to import additional libraries for more boards if the board being used is not natively supported. Arduino requires code to be written with a combination of

C/C++ languages. This means that for our development, we will be able to take advantage of object oriented programming which gives us much stronger capabilities. There is a large amount of crossover between that of java and C++, therefore it would make our development process simpler overall.

3.4.2.2.3 Hardware Code Comparison

There are many possible choices when considering possible software development environments. Two options were primarily chosen based on the prerequisite of having experience with these environments previously through school. The following table compares the two possible choices for development of the project code. The comparison criteria are native language support, Arduino compatibility, and OS support. These are the core differences between the two integrated development environments that we have previously had experience with.

Name	Language Support	Arduino Compatible	OS Support
Code Composer Studio	C	No	Windows, MacOS
Arduino IDE	C/ C++	Yes	Windows, MacOS, Linux

Table 5 - Hardware Code Comparison Table

3.4.2.2.4 Hardware Code Selection

In terms of hardware coding environments, there can be a multitude, however we limited our possible selections to options that are either required, or we have previous experience with. With this in mind, it is very easy to remove Code Composer Studio from the running due to the lack of compatibility with the Arduino board. This requires the selection of the Arduino IDE due to the native support of the program with the board we chose to use, the Huzzah32.

3.4.2.3 Databases

As the monitoring system plans to contain and house both sensor data, as well as possibly housing plant information in the future, allowing for the end customer to import data about the plant they are wanting to monitor. Having the information

regarding the plant information would allow the program to custom tailor the sensor levels to more accurately imitate the actual plants necessities.

3.4.2.3.1 MongoDB

MongoDB is considered to be a fairly easy to access database, allowing even a user not extremely familiar with databases to create a fully functioning database in a quick and efficient manner. MongoDB MongoDB is a database that is considered non-relational. Non-relational databases allow for flexibility in the data types that are input into the system. The lack of relational requirements help fuel the popularity and mass adoption of this database. Although MongoDB is considered to be non-relational, the database truly shows its flexibility when considering the ability to dynamically house relational, and non-relational data, on the fly, and simultaneously. Compared to other databases, this open fluidity in terms of data structure was something not before seen with databases that could scale to the level that MongoDB can.

3.4.2.3.2 MySQL

MySQL is a play on the base ideology of SQL, which stands for structured query language. This structure is one of the biggest limitations of the database, as it requires relationships to be formed between all data within the database. This relational requirement causes an issue when considering data that is unstructured in nature. The data that the plant monitor may contain most likely can be forced to have relationships, although in terms of development ease, using a database that does not have this limit may ensure a quick and effective development process.

MySQL truly shines in regards to performance, however. As the data is already indexed, and has pre-formed relationships, data access is quick and extremely efficient. When data is needed to be recalled from the database, the database systematically goes through the individual relationships, finding the appropriate data trace, as compared to say MongoDB which requires a “lookup” function to brute-force search the entire database to find a given data value. This lookup performance comes at a price, however. Entering values into a MySQL database requires entries to be entered row by row, as compared to a non-relational database that can input values without the need for input to be in a certain orientation or structure. This row by row entry style can cause the program to take longer to input the same data, although this time difference is fairly negligible to the end user.

3.4.2.3.3 Database Comparison

Databases come in a wide variety of options. When considering the different options of the databases, we chose to focus on the relation type, data entry type, as well as the flexibility of the data types within the database. Further expanding upon these criteria, relation type means whether the data within the database has to have a specified relationship or structure to it, or can the data be input freely regardless of possible relationships, or lack thereof. Data type entry consists of how the data is input into the system, once it is received. This means that there are multiple possible features that have to be checked and input at the time of data entry, regardless if that is the data being input, or alternatively, is there not a defined way that the data has to be input into the system. Lastly, the flexibility of the data types. This flexibility could possibly allow for a database to quickly and easily input new data types and data structures as needed, without the need to fully restructure the entire database.

Name	Relation Type	Data Entry	Data Type Flexibility
MySQL	Structured	Line by line	No
MongoDB	Unstructured	As received	Yes

Table 6 - Database Comparison Table

3.4.2.3.4 Database Choice

When considering the options available, MongoDB seems to be the better all around option. In terms of overall performance, scalability, and flexibility of data types, MongoDB objectively is the better choice. This decision comes from the viewpoint of future project progress. As it is early on in the design process, all factors may not have been fully considered or explored, possibly allowing another option to be considered in the future.

The dataset that the project team is aiming to work with has not been fully defined as of yet, meaning that the flexibility and acceptance of all data sets allows for the development to progress forward in any way that the team sees fit. This openness in terms of development allows for the team to not be tied down in any way, allowing for full creativity when designing the system and structure of the software.

3.4 Sensors

Sensors detect small changes and thus produce an output signal which is sent to a processor to make the allotted changes. Sensors make it easier for humans as

well as machines to distinguish small changes in which they can not see or make sense of. Sensors take data from the real world and convert that information into a stream that will be known to the machine. Advantages of sensors include but are not limited to accelerating certain processes as well as increasing accuracy, allowing access to data in real time, and increasing productivity for the user or machine.

The sensors are a key portion of the project, allowing the system to take in data, process that data, perform some key operations via the software, and then perform an output. The key sensors that were researched were temperature, humidity, moisture, and light. Focuses on accuracy, power consumption, and price were made when making final decisions. We considered a pressure sensor, however we felt as though this project would be better without one as most plants will not require such changes to pressure in this area. Sensors are important for our device because they are needed to make the measurements that humans can not make when it comes to plants. In this case the probe will be inserted into a potted plant or a garden and the sensors will measure various components for the user to then make a change or act upon it.

In order to get a final product that is cohesive and works as a unit, we will need to make sure that all components are compatible and work well together, especially when it comes to the sensors. Many sensors will not be compatible with the microprocessor that we have chosen, especially when it comes to operating voltages or general pin connections.

Some advantages of sensors in general are improving data capture and storage if needed. Another advantage is that because of the sensor we can have almost no transmission loss and continuously be sensing data during real time analysis because sensors can make that possible if that is needed. ANother advantage of sensor readings is that they will always be more accurate than a reading a human can take by interpreting data.

Some disadvantages to sensors can be limited or narrowed range due to some problem or a wrong choice in sensor for a specific project. Another disadvantage is that some sensors have a limited shelf life or time it can be used, while we would like our device to work forever unfortunately many sensors have less time they can be used. So the possibility of having to change out one component and not all the others is pretty high. Another disadvantage is that some sensors are that they use data from a chip that can be programmed and the decision making then comes from the sensor, this is only a problem in programmable chipped sensors. Another disadvantage is that these sensors can be sensitive to environmental changes if they are quite extreme over time.

3.4.1 Temperature Sensors

Utilizing a temperature sensor will allow us to ensure that specific species of plants are not placed in temperatures that are too hot or too cold. This will allow the user to adjust the location of the plants that are being monitored accordingly if needed. Temperature is an incredibly important aspect to a plant's environment, because all plants have a temperature range that they can live in.

Many plants, specifically those that bear fruit, that grow up north in colder areas might not grow as well in Florida due to the fact that plants in Florida do not get as many chill hours. Chill hours are defined as hours that the plant is in weather with a temperature between 32F and 45F. One example of this is apples. Apples need a fair amount of chill hours (roughly 1700) in order to grow fully. Some new varieties of apple do not need this many chill hours anymore, however, so they can grow in places that do not get as cold. One opposite example that this team knows well is oranges. Orange trees can withstand temperatures as low as 27F, however, any lower can risk damaging any fruit, which is the main reason it is not grown in places that get colder than that often.

Fruit trees are not the only examples of temperature impacting plant health, however. Temperature affects plants as a whole, because lower temperatures lead to the natural processes of the plants slowing down or stopping completely. Most plants that would be grown in a garden can not even survive sub-freezing temperatures. Freezes can cause many issues for plants, especially if they are unprepared. It can freeze the plant itself, stopping any processes keeping it alive and on top of this, sub-freezing temperatures means that much of the water keeping the plant alive will also freeze.

For some plants in which temperature is not a heavily relied on issue, the user could use this portion as an extra data entry for their new found plant knowledge. Many houseplants do not need to be in a specific temperature range, however it could become a sort of experiment to see whether a plant is able to bloom better in certain temperatures versus others. Our temperature sensor should be able to take in that data entry and store it for our user.

The importance of a temperature sensor in regards to a plant monitor is because plants ultimately use temperature as a seasonal cue. If a user has a flowering plant, and for some reason has this plant in an environment where temperature is opposite of ideal, the user will be able to see at all times what temperature surrounds their plant and thus be aware of any possible flowering situations when they are not prepared or during non flowering months.

Some advantages of temperature sensors are that they are extremely low in cost which can be important for bulk buying. Another advantage is that they can be reliable for being used continuously and also have a fast response time due to low thermal mass. Some disadvantages of temperature sensors are that they can sometimes be non linear or even less stable (thermocouples). Another disadvantage is that a reference is often needed and they tend to be less sensitive

(thermocouples). Another disadvantage is that they have low resistance and need a current source to work which can also result in a longer response time (resistive temperature sensor).

3.4.1.1 Thermocouple Sensor

The most common type of temperature sensor is the thermocouple. These are thermoelectric sensors which include two different metals bound together in some way at one end in a junction. One metal is kept at a constant temperature while the other varies, the voltage output that is read is the temperature difference between the metals voltages. Due to the variability in performance depending on the metals used, there have been standards implemented in order to simplify the distribution and production of thermocouples.

Thermocouples are important because they can be used to maintain specific temperatures in this case for our plants environment. Thermocouples are mainly used in the industries, however we plan to use potentially a thermocouple as a temperature sensor for our plant. We may be able to use a thermocouple sensor to not only measure the temperature of the plant's environment, but help regulate the temperature around the plant as well.

3.4.1.1.1 Temperature Sensor (TSYS01) Peripheral Module

The performance of this sensor includes measurement from -40°C to 125°C with an accuracy of 0.5°C. It activates on very low power consumption and operates from 2.2V to 3.6V. This sensor requires I2C communication and includes fast response time. This sensor is calibrated during manufacturing. Its features include a 12 pin connector and 24/16 bit resolution for temperature. Price : $16

3.4.1.1.2 Low Voltage Digital Temperature Sensor (N34TS108)

The performance of this sensor includes measurement from -40°C to 125°C with an accuracy of 0.75°C from -20°C to 85°C and 1°C from -40°C to 125°C. This sensor works on low current, 6µA, and a supply range of 1.4V to 3.6V. This sensor has a resolution of 12 bits. Price : $0.60

3.4.1.2 Resistive Temperature Sensor

A resistance temperature detector utilizes a wire of pure metal that surrounds a cylindrical shape. Pure metals allow for an accurate measurement of resistance vs temperature as there is a defined relationship between the two. Compared to a thermocouple, the resistance temperature sensor has a much slower response time, often taking seconds rather than milliseconds. The importance of this type of temperature sensor is that it is calibrated among a resistance that is at a known standard. Because of this these sensors can have increased precision and stability for a given extra length of time. This can be important for our plant life because although this type of sensor has a slower reaction time, it can give a more accurate reading for hopefully a longer period of time. It is important for our plants to get an accurate reading of temperature rather than a faster but also wrong reading and thus having the user make wrong accommodations for their plant.

3.4.1.2.1 Precision Temperature Sensor (LM335)

The performance of this sensor includes measurement from -40°C to 125°C with an accuracy of under 1°C under 100°C. This sensor operates as a 2 terminal zener and has a breakdown voltage directly proportional to the absolute temperature at 10mV/°K with a voltage range of 2.92V to 3.04V. This sensor has linear output and a dynamic impedance of less than 1Ω and operates within the range of 450μA to 5mA. This sensor can detect temperature during continuous mode as well as an intermittent mode. The response time of this sensor is about 2 μsec. Price : $0.62

3.4.2 Temperature Sensor Selection

Of the above temperature sensors, the one we have chosen to use is the LM335 because of its accuracy, wide voltage and current range. We will be able to use this temperature sensor all the time. This sensor is a zener diode whose reverse breakdown voltage will be proportional to absolute temperature. This sensor determines the temperature by measuring the electrical signals. It can also be easily calibrated which is good for us because this may need to be different for certain plants. Its price is better than TSYS01 and will allow us to buy multiple for all our systems. The reason we did not choose the TSYS01 is simply due to its price and bulkyness.

Figure 7 - LM335 Temperature Sensor pin

3.4.3 Temperature Sensor Comparison

This section compares the sensors on the basis of the measurement range, accuracy of the measurements, range of the required supply voltage, as well as price.

Name	Measurement (°C)	Accuracy (°C)	Supply Range (V)	Price ($)
TSYS01	-40 to 125	0.5	2.2 to 3.6	16.00
LM335	-40 to 125	1	2.92 to 3.04	0.62
N34TS108	-40 to 125	0.75	1.4 to 3.6	0.60

Table 7 - Temperature Sensor Table

3.4.4 Humidity Sensors

Humidity is a critical factor in plant survival; if humidity is too high, it can cause a plant to develop fungal or bacterial diseases and most likely die. On top of this, high humidity environments can encourage mold or bacteria as well as encourage pests that can harm the plant as a whole. If humidity is too low, on the other hand, it can stunt a plant's growth which can cause serious health issues in the short and long term. Humidity sensors will be used to monitor this in order for the user to know whether to move the plants home to a new area or not. Most of the time, temperature sensors have built-in humidity sensors, so there is a good chance we will be using one that has both in order to save space and cost.

Humidity sensors can be used in plant health to promote plant growth as well as reduce the overall labor for the plant to try and work hard for their food or water. Humidity sensors can also help decrease insect problems for the plant. Humidity sensors are important because it can be a super important measurement to get right, especially when trying to create a perfect environment for your plant to thrive and succeed in. Instead of doing this measurement by hand, we will have a humidity sensor do it for the user.

Humidity sensors work by reporting moisture and temperature, and the ratio from these two of moisture at that particular temperature to the maximum moisture at

that temperature creates what we know as relative humidity. There are three main types of humidity sensors: capacitive, resistive and thermal.

Some advantages of humidity sensors are that they are overall easy to clean and cheap to buy in bulk. Another advantage is that it is able to be used in certain already contaminated environments, in this case a soiled potted plant. Some disadvantages are that they may have some limited accuracy or even limited range for measurement and can even be slow to measure overall.

3.4.4.1 Capacitive-type Humidity Sensor

Capacitive sensors work by using a capacitor to measure the relative humidity. The capacitor is a small hygroscopic dielectric material that is between two electrodes. Most manufacturers choose to use plastic or polymer as the dielectric and their dielectric constant ranges from 2 to 15. This constant along with the sensor geometry determine what the capacitance value will be when there is no moisture present in the air. Because the dielectric constant of water vapor is 80, an increase in the capacitance value will denote an increase in humidity in the environment. This means that the dielectric constant is directly related to the value of humidity. Because the capacitor simply functions to fluctuate depending on the humidity, there must be other electronics included in order to make it a functional sensor which can output data.

3.4.4.1.1 Capacitive Humidity Sensor (HS1100)

The performance of this sensor includes a humidity operating range from 0 to 100 %RH and has an operating temperature from -40°C to 100°C. The supply voltage is high at 10V. This sensor has a high reliability and long term stability. This sensor is suitable for voltage and frequency output circuitry. Price : $8.99

3.4.4.1.2 Relative Humidity and Temperature Sensor (SEN-18364)

The performance of this sensor includes a humidity operating range of 1 to 100 %RH with a typical accuracy of 3 %RH and a temperature range of -40°C to 80°C. This sensor has a wide voltage support of 2.2V to 5.5V and excellent long term stability as well as a fast response time of less than 8 seconds. The resolution is typical for this type of sensor at about 0.024 %RH. Price : $5.95

3.4.4.2 Resistive-type Humidity Sensor

Resistive sensors use a conductive polymer, salt or substrate in order to determine a change in electrical impedance. There are typically electrodes that are deposited on the substrate or electrodes are wire-wound on a plastic. The sensor will absorb water vapor and the ionic functional groups are dissociated. Once the groups are dissociated there will be a measured increase in the electrical conductivity. The result is a nonlinear response to varying humidity but using analog or digital methods will result in a linearization of the response.

One major advantage of resistive sensors for humidity is the fact that it can be calibrated by a resistor at a fixed relative humidity point. Doing so bypasses the need for calibrating a standard humidity. These sensors also boast an extended life period of over 5 years in residential and commercial environments, which meets our requirements.

3.4.4.2.1 Resistive Humidity Sensor (HR202L)

This sensor has a performance including a humidity detection range of 20 to 95%RH. The supply range is 1.5V and has a power rating of 0.2mW at maximum sine. The accuracy of this resistive sensor is about 5%RH with a response time of less than 10 seconds. This sensor is relatively new to market and is made from organic macromolecule materials. Price : $2.11

3.4.4.3 Thermal Conductivity-type Humidity Sensor

Thermal conductivity sensors differ from the previous methods as it measures absolute humidity. Absolute humidity ignores the air temperature and will only determine the amount of moisture in the air. It does so by suspending two NTC thermistors by thin wires with the sensor, one is housed in an enclosed space with ventilation holes while the other is housed in a hermetically sealed chamber. Due to the fact one thermistor is in a sealed chamber, it will act as the control and the thermistor with ventilation will be what determines the humidity. Current is sent through both wires and the conductivity difference between the two thermistors will give the absolute humidity. These are thought to be the most accurate type of humidity sensor but due to the nature of its measuring, it may not be the most useful for our particular application.

3.4.5 Humidity Sensor Comparison

This section compares the different humidity sensors on a variety of criteria. The different criteria are the measurement ranges (read in % of relative humidity), accuracy of the overall system (read in % of relative humidity), the required supply voltage range for the sensor, as well as the pricing of the sensors themselves.

Name	Measurement (% RH)	Accuracy (% RH)	Supply Range (V)	Price ($)
SEN-18364	1 to 100	3	2.2 to 5.5	5.95
HS1100	0 to 100	5	10	8.99
HR202L	20 to 90	5	1.5	2.11

Table 8 - Humidity Sensor Table

3.4.6 Humidity Sensor Selection

Of the humidity sensors above, we have chosen the SEN-18364 because it has a lower and wider range of voltage support, as well as long term stability and a cheaper price to buy multiple. This sensor also has I2C capabilities which makes implementing it a lot easier for the board we intend on using. The sensor also has a temperature sensor built in, however we do not plan to use this. We may change our minds later and use it as a back up. This sensor can be easily calibrated to the specifics that we need for each plant type. This sensor detects changes that alter electrical currents in the air, thus monitoring these small changes is how we calculate the humidity. The reason we did not choose the HS1100 is because of higher price as well as higher voltage, this sensor also had a slower reaction rate.

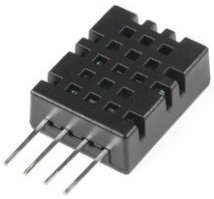

Figure 8 - SEN-18364 Humidity Sensor pin

3.4.7 Moisture Sensors

It is absolutely necessary to ensure, especially with plants that are potted, that the moisture levels in the soil are just right. Our moisture sensor will be inserted straight into the pot and will consistently check on the moisture level so that the user knows what the soil is looking like when they are not able to test it themselves. Many plant growers stick a single finger in the soil to see if it is dry enough to water

their plant, however this is only good for the top of the soil where the heat or light is mostly hitting.

By having a moisture sensor that is inserted all the way to the bottom of the pot, the user will know for certain whether the soil is moist or not. Too little moisture will hurt the plant's ability to grow and possibly kill it while too much moisture can essentially drown the plant. It is also important to note that for some plants, too much water could harm them. It is not only important to note the "normal" moisture level, but be able to change that when you have a desert plant or succulent versus an herb.

Traditional methods to measure the moisture in soil include the use of gravimetric measurement. In order to do so it requires removing, drying and weighing of a soil sample. This is obviously not the optimal method for our application and therefore the soil moisture sensor is far more appropriate. Moisture sensors do not directly measure the water content of the soil, rather it indirectly does so by use of electrical resistance, or neutron interactions.

When water is drawn from the soil and the moisture level decreases, the resistance through the sensor goes up and the sensor can monitor that change. The same works in the opposite direction, if there is more water added to soil, then the resistance goes down. This is referred to as estimating volumetric water content but sensors can also measure the water potential. Rather than measuring the resistance, sensors that measure water potential do so with tensiometers and gypsum blocks.

One of the biggest things to consider in terms of a moisture sensor is its ability to last a long time. Being used to monitor moisture means that the sensor will most likely be taking a lot more of a beating in terms of wear and tear compared to other sensors that will simply sit in open air.

Some advantages of a moisture sensor is that the sensor can ultimately save water in the long run by allowing our user to not over water and thus waste water. Another advantage is that this sensor can allow you to save on energy costs and fertilizer costs. Some disadvantages of the moisture sensor is that the probe must be inserted into the soil and it can be hard to find long term probes. Another disadvantage is that some sensors may require initial evaluation of the soil and or conditions of the plant before inserting the probe.

3.4.7.1 Tensiometer Moisture Sensor

The tensiometric sensor is the most common of moisture sensors that measure the moisture level in soil. This sensor is a water filled probe that ultimately measures the tension in the soil which implies the potential soil moisture in a pot. This is done by measuring the amount of water a certain soil will retain, or

otherwise how much water the plant has drunk or not. This is important for our device because we need a sensor that is able to be submerged into the pot with the plant and able to measure the amount of water in the soil without the user having to constantly check themselves.

3.4.7.1.1 Soil Moisture Sensor (VH400)

This sensor includes a probe that is inserted into soil with a detection depth of 93mm. The power consumption is low at less than 13mA and a supply voltage of 3.5V to 20V. This sensor has a rapid response time that reacts to constant changes of water moisture. This sensor is made to be waterproof and able to hold up to the soil for long time use. This sensor is compatible with Arduino, and is also easily paired with other sensors of its kind. Price : $41.95

3.4.7.2 Volumetric Moisture Sensor

The volumetric sensor, similar to tensiometric, is another of the most common of moisture sensors that can measure the moisture level in soil. This sensor is a probe that can be inserted into the pot with the plant. The way this sensor works versus the previous is that instead of measuring the tension of moisture, this sensor measures the actual volume of water that is left in the soil. Again it is needed for this device that the probe be able to be in the pot with the plant, but also instead of measuring the tension, it makes more sense for this project to be measuring the volume of water in the soil so that this measurement can be different for each plant and can be easy to note.

We are able to make more sense of water volume than water pressure for different types of plants. This sensor can also be easily linked to an irrigation system if we so choose to input one in the future. We plan on adding a water reservoir which can work side by side with this type of sensor. It is also good to note that this probe does not have water in it and therefore is less likely to break being a single probe rather than a tubbed probe.

3.4.7.2.1 Soil Moisture Sensor (STM32)

The performance of this sensor includes a detection depth of 38mm and an analog output. The operating voltage is from 2V to 5V. This comes with a module application we can apply such as an auto watering system, which we may or may not use. This sensor has a 3 pin custom connector jumper wire. Compatible with Arduino. The way this is designed will make it easier to insert it into the pot directly with the plant. Price : $2.99

3.4.7.2.2 Moisture Sensor (SEN0114)

The performance of this sensor includes a value range of 0-300 dry soil, 300-700 humid soil, 700-950 in water. The power supply required is 3.3V or 5V and the current is 35mA with an output voltage of 0-4.2V. Price : $2.70

3.4.8 Moisture Sensor Comparison

This section compares the various moisture sensors based on the output type, depth of detection, necessary supply voltages, and pricing.

Name	Output	Detection Depth (mm)	Supply Range (V)	Price ($)
VH400	Analog	93	3.5 to 20	41.95
SEN0114	Analog	20	3.3 to 5	2.70
STM32	Analog	38	2 to 5	2.99

Table 9 - Moisture Sensor Table

3.4.9 Moisture Sensor Selection

Of the two moisture sensors listed above, we have chosen the STM32 because it will be easier to program how we would like since there is no set range and already an application connected. The operating voltage is not set like the other choice so this will be easier to connect to our system. This sensor works off of electrical charge output, which means when there is less water in the plants soil the resistance increases and vice versa. The reason we did not choose the SEN0114 is because of the premade ranges as well as the detection depth, in case we had a taller pot it is better to have a longer sensor rather than too short in case a pot is hoarding water at the bottom. Due to the nature of how plant roots work though this is thought to be a negligible complication due to the length of the sensor probe.

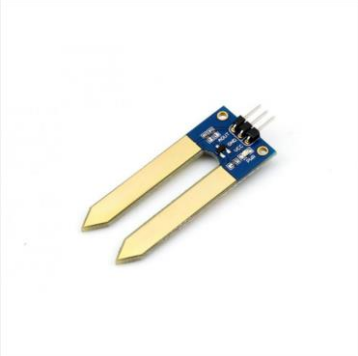

Figure 9 - STM32 Soil Moisture Sensor

3.4.10 Light Sensors

Alongside moisture, light is arguably the most important factor to consider when growing/maintaining plants. Light is crucial to a plant's health for one obvious reason, it needs light to photosynthesize and produce food for itself. Too little light will lead to a quick wilting and death of most plants. There are three different light levels for plants which include low light, medium light, and high light. In addition to this we have house plants versus outdoor plants which can be at each light level. If we have a houseplant that needs medium light but it is getting high outdoors light then this plant is not going to be okay. That is specifically how important not only lighting but the right lighting is important for overall plant health. It can be the difference of a wilted beautiful flower to a fully bloomed beautiful flower.

3.4.10.1 Types of Light Sensors

Light sensors are often referred to as photodetectors or photosensors and there are varying types of detection. Photoemission refers to sensors that detect photons and particles which cause current flow, these are considered to be producing light and thus the size of the current detected determines how much light is being emitted. The current flow is determined by how wide the holes which allow electrons to flow, the width of these holes is solely dependent on the amount of light hitting the optical filters.

There is a slight variation of these light sensors called phototransistors, which operate in the same way but have added amplification which makes them even more light-sensitive. Alternatively there are photoresistors, which are also referred to as light-dependent resistors. These are made of high resistance semiconductor material, which is sensitive to visible and near-infrared light. They operate by changing the resistance value depending on how much light is being applied. Such an implementation is used in street lights, the photoresistors will enable lights turning on if the resistance meets a particular value.

3.4.10.2 How Light is Determined

There are a multitude of terms that are used to determine the strength of light, often including candela which is the strength of light to our eyes. Lumen is a term often used for lightbulbs or any other lightsource and is a standard unit for determining how much light is emitted. Lux however takes into account the surface area of the light, namely how much of the surface is affected by the light. Understanding these terms will help in choosing which light sensors are best for our current application as the manufacturers will often vary what unit of measurement is used when testing their products and being able to distinguish their meanings will be greatly beneficial in our part selection. If we were to choose a light sensor that measures the lux, it would mean that the sensor is determining how much light is falling onto our plant which would be quite useful when trying to notify the user on actions.

One thing we need to consider in our design is only monitoring light levels during the times that the plant should be getting light, meaning it should not run at night time and take into account weather patterns like cloudiness. This means we will want a sensor that has specific modes for all types of weather and times of day.

Some advantages of light sensors are that using such can make it easier on the ones using it to implement an automatic lighting system. Another advantage is that it can be used to control brightness. Another advantage is that they are often decently cheap to buy and available in many sizes. Some disadvantages are that the sensors can be sensitive to temperature changes if they are of ultrasonic type. Another disadvantage is that these sensors may have a slow response rate even though they may have low power consumption.

3.4.10.3 Light Sensor (SEN0097)

This breakout board has a 16 bit AD converter built-in meaning it will output a digital signal directly so it forgoes the need to calculate anything as the data is already in Lux.
The performance of this sensor includes multiple modes such as night, moonlight, cloudy indoor, cloudy outdoor, sunny indoor, sunlight afternoon, etc. This sensor has a supply voltage of 3V to 5V and a wide range of high resolution. Price : $4.50

3.4.10.4 Light Detection Sensor (LM393)

The performance of this sensor includes a photoresistor module that detects light intensity. Depending on the amount of light detected by the photoresistor module, the resistance will change, typically the resistance decreases with increasing light

intensity. The working voltage for this sensor is 3.3V to 5V with an analog output. This sensor outputs a clean waveform, with an operating current of 15mA. This sensor can also detect ambient lighting. The response time of this sensor is 1 µsec. Price : $8.99

3.4.10.5 Ambient Light Sensor (HW5P-1)

This sensor uses a built-in optical filter that detects ambient lighting by linear output. This sensor has low dark current as well as low working Lux. The performance of this sensor includes a supply voltage of 3V to 15V and a stable working temperature. This sensor has multiple Electrical and Optical characteristics which include which type of ambient lighting it is currently reading. However because of this module the sensor has quite a slow reaction time. Price : $0.95

3.4.11 Light Sensor Comparison

This section showcases the comparison between the different light sensors based on the mode of light testing, the necessary supply voltage range, as well as pricing of the sensors.

Name	Modes	Supply Range (V)	Price ($)
SEN0097	multiple	3 to 5	4.50
LM393	intensity	3.3 to 5	8.99
HW5P-1	multiple	3 to 15	0.95

Table 10 - Light Sensor Table

3.4.12 Light Sensor Selection

Of the three above light sensors, we have chosen the LM393 because since there are no modes we can program it how we want based on what the sensor reads in an analog output. That is important because certain plants are specifically low light or high light plants and these need to be accounted for as to not exceed the specific amount of light a plant needs to thrive. This sensor is a photoresistor which essentially works as a switch, or comparator circuit, set to low or high on its output.

Sensitivity can be easily adjusted to the desired output by changing the potentiometer. The reason we did not choose the other two light sensors is because of their multiple pre-programmed modes, while we just wanted a simple ambient light sensor to program how we want to use it.

Figure 10 - LM393 Light Sensor

3.4.13 pH Sensor

PH can be argued as one of the most important things to look at for plant health. Not only does the plant need their water to be at optimum pH, but their soil as well for nutrients to thrive and survive. A pH sensor can be crucial for plant monitoring because certain plants need to maintain a specific pH range to be healthy and happy. PH is important to a plants overall health because pH is what influences nutrients to be available or grow in the plants soil. When pH is in the right range for certain plants, it is easier on the plant to take in the nutrients available to them. Many plants can maintain a healthy pH for about 3 to 4 months, while others can maintain for less that or more. This is more common in outdoor versus indoor plants, being able to hold a healthy pH for some time, while mostly hydroponic or aquaponic environments make it harder to maintain.

A pH sensor can help our device in the sense that the user can be aware of the overall health of their plant. The overall importance of a pH sensor will be to show that their plant soil is at optimum pH for the nutrients to thrive and the plant to ultimately eat. If a plant is not effectively and actively eating or taking in nutrients the plant will stop drinking water and other nutrients to survive and thus die. If pH is neglected to be taken care of for a plant, the plant will suffer deficiencies in nutrients as well as reducing harvest if it is a flowering or harvester plant, and of course a decreased seed growth rate.

Some factors that affect plant health include climate, soil texture and mineral content. While climate and soil texture can not always be changed, the user can

aid the plant in giving or taking away minerals in their soil. Some plants need to live off a decreased pH which means more aluminum in their soil. When the aluminum levels become too high they can potentially become toxic and harmful to the plants health. This mainly affects the availability of nutrients for a healthy plant.

Optimum soil pH for a typical home garden is about 6.5. Most plants thrive from a range of 6.0 to 7.0, which is in the slightly acidic range. A pH level of 5.0 or below is considered too low for these plants and thus heavy metals will become toxic for plant growth and survival. A pH level of 7.0 or above is considered too high for these plants and thus the nutrients are unable to be broken down enough which again in turn slows down the plants growth and survival.

To reduce a plant's pH if it is too basic and needs to be reduced to a slightly acidic level, the user can add elemental sulfur to the soil. By adding a pH sensor to our device we can help the user to know when to add this sort of fertilizer and be able to see what is happening to their plants pH level by adding a little at a time. By being able to see the pH lowering the user can determine how much of this fertilizer they put into the soil and regulate the pH over time. To raise pH if the plant soil is remaining acidic and we need to bring it up to slightly acidic, the user can add baking soda or lime to the soil. Again the pH sensor will allow the user to see where the pH level is at and see after adding a bit at a time where their pH level rises to.

Some advantages of pH sensors are that the sensors are highly accurate and provide mostly precise pH values to the user. Another is that they are used in many varieties of applications that would not be first thought about, such as cheese making, or stain removing. Another advantage being it is a simple and fast way to measure any pH and of course easy to calibrate by using a buffer solution when needed. Some unfortunate disadvantages are that the pH sensors are mostly very expensive and have a high chance of breaking the glass electrodes due to their being so delicate. Another disadvantage is that the probes do need to be calibrated quite often however the process is simple.

3.4.13.1 Combination pH Sensor

Similar to the other sensors that are used in our device, there are multiple types of pH sensors that are available. There are combination sensors which are the most commonly used and act as the foundation for all other types of pH sensors. The way they operate is quite simple and has two electrodes, one which acts as a reference and the other which records any fluctuations in pH. The reason these types of pH sensors are so common is because they are mainly used in a laboratory setting.

3.4.13.2 Differential pH Sensor

Building on the previous type of sensor, differential pH sensors have three electrodes, with the third electrode acting as a metal ground which accounts for any error or misreading due to an external factor. The extra importance of this third electrode is so that the probes do not become contaminated due to random pH changes, in other words the third electrode serves as a buffer to the rest of the probe.

3.4.13.3 Laboratory pH Sensor

The most light weight sensor which is used in pools, aquariums, and hydroponics is a laboratory pH sensor. The laboratory pH sensor is very similar to the combination but uses a plastic body and a small glass tube. These sensors are extremely versatile and often considered to be basic in nature which are perfect for our application as it only needs to determine the pH of the soil and can be inserted straight into our pot with soil and water and plant. The way this type of sensor works is that there is a single electrode encapsulated in a test tube solution with a glass bulb surrounding it. Activity from hydrogen ions around the glass bulb in the soil creates signals which are then sent through to the electrode. A cool thing about this type of pH sensor is that they can be customizable for certain types of application, in this case our pH sensor is made specifically for soil pH detection. The three categories these sensors can come in are basic, advanced and research type. In our case we will be looking at laboratory basic type sensors.

3.4.13.3.1 Analog pH Sensor / Meter (GRV-pH)

This sensor includes a small screw probe and a jumper cable to the meter board. The screws are inserted into the soil for a continuous analog reading. The accuracy of this device is about 0.2. The range of pH this sensor reads is from 0.1 to 14.0. This sensor is compatible with arduino but not ras pi. The supply range is from 3.3V to 5.5V. Price : $28.99

3.4.13.3.2 Lab Grade pH Probe (ENV-40-PH)

This sensor is a probe in which can be inserted into the soil of the plant for about 2.5 years with recalibration required after 1 year. The response time of this sensor is 95% in 1 second. The maximum depth that the probe can be inserted is 70m while the probe is 150mm by 12mm. This sensor reads a pH range of 0 to 14.0. The accuracy of this sensor is 0.002. This device is completely submersible in soil or even water. The supply voltage of this sensor is 5V. Price : $85.99

3.4.13.3.3 Consumer Grade pH Probe (ENV-30-PH)

This sensor is also a probe in which can be inserted into the soil of the plant for about 12 to 18 months with recalibration required after 3 months. The response time of this sensor is 95% in 4 seconds. The maximum depth that the probe can be inserted is 35m while the probe is 150mm by 12mm. This sensor reads a pH range of 2.0 to 13.0. The accuracy of this sensor is 0.1. This device is also fully submersible in soil or even water. The supply range of this sensor is 5V. Price : $48.99

3.4.13.3.4 Soil pH Sensor (RS-PH-*-TR-1)

This sensor is a two pronged probe that can be inserted into the soil of the plant and has long term stability which is about 5 percent of a year. This probe can measure a pH range of 3.0 to 9.0 ph. This sensor has an accuracy of 0.3 and a response time of less than 10 seconds. The probe is 123mm and a power supply voltage of 5V to 30V. The shell of this sensor is completely sealed which makes it resistant to corrosion and waterproof as well. Price : $40.00

3.4.13.4 Process pH Sensor

Another type of pH sensor is the process sensor which uses a higher type of sensor technology combination and mainly measures just water. These types of sensors are extremely durable which allow for them to be inserted into a pipe or mass body of water. This sensor is mainly used for industrial or wastewater monitoring due to this sensor's heavy duty coating and possible self cleaning design allows for constant or continuous water pH sensing.

3.4.14 pH Sensor Comparison

Name	Accuracy	pH Range	Supply Voltage (V)	Price ($)
GRV-pH	0.2	0 to 14.0	3.3 to 5.5	28.99
ENV-40-PH	0.002	0 to 14.0	5	85.99
ENV-30-PH	0.1	2.0 to 13.0	5	48.99

Name	Accuracy	pH Range	Supply Voltage (V)	Price ($)
RS-PH-*-TR-1	0.3	3.0 to 9.0	5 to 30	40.00

Table 11 - pH Sensor Table

3.4.15 pH Sensor Selection

Of the three above sensors, it was only a race between the two ENVs. The sensor and meter were a bit too cheap and we want to stay away from the analog output of their meter when converting it to our microcontroller and screen. The two ENVs are very similar but we have chosen the ENV-40-PH. This one was chosen because of the lifespan it holds. After buying two of the others you are at a higher price than one of these for twice the time. This sensor has a better accuracy level and the range is better at 0 to 14.0 rather than 2.0 to 14.0. We can potentially make the pH sensor an add-on when creating multiple devices simply due to the price of this sensor, however for our prototype it is the perfect choice. This sensor will be good for our project and for the user to have the probe in use for a long time and since it is waterproof it will be able to hold up to moisture standards for any plant the user chooses to grow.

Figure 11 - ENV-40-PH pH Sensor

3.5 Power Source

Selecting a power source for any powered device is one of the most important things to consider, because taking into consideration every factor of the product is essential. Normally, most powered devices nowadays are either rechargeable or have a direct connection to a power source via a cord and power supply, mostly

being our outlets we have in the house. Our device needs to be portable, and most portable devices nowadays use rechargeable batteries, and this is what will be used here as well.

When considering options for powering a unit such as this, going with a low-cost, low power, and rechargeable source were the most important factors. Having to replace batteries is mundane, but if our system is already low power, there is no need to rule out the obvious option of solar charging. This is especially true because this unit will be placed outside or in sunlight. Therefore, solar charging fits all three requirements listed above and is the choice for this system.

It is important to note that in order for this system to run on lower power, the unit will not be on at all times and will enter sleep mode especially during the night ime. It is not necessary to have a solar cell, but it will be an option that will extend the time that the batteries can run.

3.5.1 Battery

The best option for our choice will be the use of a battery. This is because there is no guarantee that there will be an available outlet within reach of the device, and batteries are easily attachable to the HUZZAH32. We also will want to use rechargeable batteries instead of non rechargeable, as this will allow the user to keep the same battery for a longer time if they wish to utilize a method to recharge the batteries.

Electric batteries like the ones we will be using have an electrochemical cell within them as well as an anode and cathode on either end, the former being negatively charged and the latter positively charged. When a battery is connected to an external load, a reaction in the chemical cell converts reactants to products, the former having high energy and the latter having low energy. This conversion releases electrical energy that can go to the external load.

There are many types of batteries, but the main two are known as primary and secondary batteries. Primary batteries are single-use batteries that are not rechargeable. The most common primary battery that most people probably use every day are alkaline batteries. Examples of alkaline batteries are the very common AA and AAA batteries that are used in some of our electronic devices like remote controls or toys. On the other hand, secondary batteries are those that are rechargeable from whatever source would allow for it. In this project, we will be using secondary batteries as this will allow us to offer the option to recharge the batteries with an external solar panel. The most common secondary batteries are lithium-ion batteries that are used in handheld devices like our phones.

3.5.1.1 Alkaline Battery Pack (AA Batteries)

Alkaline batteries are the most common battery in the world aside from being the most common primary battery. This battery utilizes a chemical reaction between zinc and manganese dioxide. This battery is not rechargeable as there is only a set amount of each chemical and will run out eventually. Due to environmental concerns as well as general user convenience, utilizing alkaline batteries is out of the question simply because they are single use and could possibly leak.

3.5.1.2 Lithium-Ion Battery

Lithium-ion batteries are secondary batteries, meaning they are rechargeable. In a Li-Ion battery, lithium ions move across the battery from the anode to cathode when discharging and the other way when recharging. Li-Ion batteries use electrolytes as a medium for the lithium ions to pass through. These batteries have their share of good and bad aspects. Generally, Li-Ion batteries have a higher density as well as they tend to have a longer lifespan in terms of how efficient recharging the battery ends up being over a long period of time.

Li-Ion batteries have existed for a few decades and are generally pretty cheap in terms of batteries as a whole. Even though these batteries have a lot of pros, the biggest downside is the safety of the batteries. Li-Ion batteries are able to leak as well as combust. They are generally unstable, however, due to the fact that these batteries have existed for a long period of time, high quality and low risk options exist where these negative factors are lowered in risk.

3.5.1.3 Lithium-Ion Polymer Battery

Similar to Lithium-Ion batteries, Li-Po batteries are also fairly common and utilize the movement of lithium ions across a medium. However, instead of an electrolyte which is a liquid, Li-Po batteries use a polymer electrolyte, which is gel. These batteries have a good bit of advantages over Li-Ion such as being lighter, more flexible and safer due to their low chance of leaking in comparison. On top of this, they are much less likely to combust as well.

The biggest downsides to Li-Po batteries lie in their high cost, lower power storage and shorter life compared to Li-Ion batteries. Therefore, although Li-Po batteries seem to be generally better in terms of safety and reliability, they fall short in many important areas.

3.5.2 Battery Comparison

This section serves to compare the various battery choices on the criteria of output voltage, battery life (measured in mAh), as well as overall pricing of the battery packs.

Name	Voltage	mAh	Price ($)
AA Battery Pack	6V	8000	2.95
Li-Ion Battery Pack	3.7V	6500	24.95
Lithium Ion Polymer Battery	3.7V	2500	15

Table 12 - Battery Table

3.5.3 Battery Choice - 18650 Li-Ion Battery Pack 6600mAh 3.7V

The choice here was fairly obvious for our project. The ESP32 Feather Board being used has an on-board charging port that matches this battery pack. This 3-pack of 18650 batteries will supply plenty of power to the system and be rechargeable, if desired. Using power saving methods on the board, this pack should be able to last a few hundred hours of use without needing to be recharged. However, if desired, a micro-USB charger can be used, with the solar panel being one option. Compared to the other two, the AA batteries were immediately out because of their lack of rechargeability and their voltage. The Li-Po battery was not chosen because it is more expensive in terms of price to mAh and it would not last as long as the Li-Ion battery pack that we have chosen for the device.

3.5.4 5V 250mA Solar Cell

Although the battery pack would be sufficient for maintaining the system for a considerable number of hours, one option for recharging via the board would be a solar panel connected via micro-USB. This panel, relative to the size of the rest of the product, is fairly large which is why it will be included as optional and can be connected/disconnected at will. The 250mA is low, however, a higher amperage is not necessary since we do not need the charging to be as fast as possible, and the price vs. charge rate was worth it for this system. There were many other panels that bolstered a higher amperage but they were a lot larger and tended to

cost upwards of 5-6x the price of this panel, and since it is not a necessary part of the device, this was a good place to cut down.

3.5.5 5V Step-Up Voltage Regulator (DC-DC Converter)

Utilizing a 3.7V battery pack on a board that runs at a 3.3V on the Li-Ion charging circuit means that regulating voltage is essential to the running capability of the board. Although we could add a regulator, it was decided that it was ultimately unnecessary. The board that has been chosen has a built-in regulator so that the voltage is already monitored and connecting a 3.7V Li-Ion battery will not be an issue at all. On top of this, even though the solar cell is 5V, the micro-USB connection already has a step-down regulator to bring that down to the operating voltage as well. This lowers the need to have additional components on the board, meaning we can keep it at an even smaller form factor as we had wanted to do.

3.5.6 Power Source Selection

The decision for the power source was the 18650 pack with a solar cell connected to the board to charge the batteries if needed. The solar cell is not particularly necessary for this project, as the power saving options of the board as well as the large capacity in the battery pack means that even without charging, the pack would last a long time. So, this means we can make the solar cells optional. The user can connect the cell to the board if they would like to charge the batteries, otherwise any micro-USB charging source would work as well. The solar cell is fairly large in the grand scheme of our design, so leaving this as optional is a good idea for those who would like a more compact product at the cost of consistent charging.

Figure 12 - 18650 Li-Ion Battery Pack and Solar Cell

3.7 LEDs

Light emitting diodes were invented in 1907 but only first produced in 1962 and utilized on expensive equipment and laboratory equipment. Electrons and electoral holes in a semiconductor are excited with photons in order to invoke electroluminescence. LEDs are available in many different types such as miniature, high-power, and AC power. Miniature LEDs are often a singular color and are sold as packs of multiple lights. They are also strung together onto a strip and sold as an LED strip, very common and found in almost any store that sells any home appliances. These come mostly from 2-10mm in size and are rated for 1mA to 20mA, while also being rated for 2-5V power supplies.

High-power LEDs are incredibly strong and have a strength of over a thousand lumens, despite the large output they come in a relatively small form factor. The large output means that they can be mounted onto a heatsink in order to prevent damage due to overheating. Most LEDs function with DC power but Seoul Semiconductor developed an AC-driven LED where parts of the LED are on or off depending on the cycle. The efficiency of these lights is very high as they can switch on in under a microsecond and emit more lumens/watt than incandescent bulbs. There is an estimated lifetime of 35,000 to 50,000 hours of being on and therefore for this application it is more than enough to meet our needs.

Choosing LEDs for use in our project is relatively simple as they will be used to indicate whether the plant needs attention from the user in some way. Considering the system will be modular and used on plants placed outside, it was
important to have the lights be bright enough for visibility. Being that the system itself will be smaller in size the LEDs could not be a strip and it is much more preferable to use individual lights.

3.7.1 100F5T-YT-RGB-CC Common Cathode LED

These are 5mm LEDs which have very long legs for the anode/cathode. Upon initial research it would seem that these were an easy and workable option for our device but the length of the legs left the need for more nimble solutions. It requires 3V for green and blue but 2V for red. Price: $7.96 (100 pcs)

3.7.2 0603 SMD LED Diode Lights

The 0603 are DC powered, 0.06 x 0.03 inch LED chips that are incredibly easy to put on any part of the plant system and therefore superior to the 100F5T-YT-RGB-CC which include very long legs. The voltage is similar to the rest as it will need 3V for the green, blue, and white while red/yellow require 2V. Price: $5.99 (100 pcs)

3.7.3 WS2812BLEDB Individually Addressable Smart RGB

Rather than being a single color like the other options presented, these smart pixel LEDs are connected to our microcontroller and the color can be chosen there. These also offer differing brightness levels which can be chosen using the microcontroller, although the brightest options can consume up to 160mA per LED. Price: $15.99 ($0.16 / Count)

3.7.5 LED Comparison

The table below showcases a visual comparison between the aforementioned LED options. This table compares the various choices based on pricing, voltages needed, as well as the wavelength possible by the different light values.

Name	Wavelength (nm)	Supply Range (V)	Price ($)
100F5T-YT-RGB-CC	Red: 620-625 Blue: 450-455	Red: 3 Blue: 2	7.96
0603 SMD	Red: 620-625 Blue: 460-465	Red: 3 Blue: 2	5.99
WS2812BLEDB	Red:620-625 Blue: 465-467	Red: 3 Blue: 2	15.99

Table 13 - LED Table

3.7.4 LED Selection

Figure 13 - Individual Smart LEDs

The LED that will be used for our product will be the WS2812BLEDB. This is preferable to the previous options as it can change colors depending on real-time data rather than having us just turn one off and another on. Without a doubt, it is the more expensive option, but the addressability of the individual LEDs makes this the ideal choice.

3.8 Display

For our device, the display will be used as a way to directly show the data from our sensors to the user. There is the intent for users to be able to remotely turn on the device and monitor the plants diagnostics from there; should they not have their phone near them, they can simply turn the monitoring device on and see all the sensor data from an attached display. It is a necessary consideration for the display to be able to show all data while also having the text be large enough to reasonably read.

When attempting to choose the particular display for this project there are many design considerations we must consider. There are many types of displays available on the market today, electroluminescent, QLED, OLED/AMOLED, and LCD are common. Electroluminescent displays excite atoms in order to pass an electric current that emits photons. These displays are mostly used in cars for the dash as they are incredibly rugged compared to an LCD or OLED display. A TFEL (Thin Film Electroluminescent Display) can run from -60 to 105 °C as well as 100k hours without burn-in. It is clear that the electroluminescent displays have phenomenal performance but when it comes to size and price, they are seemingly outperformed.

3.8.1 Display Data Transmission

The way in which data is transmitted from the microcontroller to the display is considered to be communication protocols. These can be split into two types, inter bus system or intra bus system. For this project, intra system communication protocols are relevant as they are the way we establish communication between components on the board. One such protocol is CAN (controller area network) that operates with message-based protocol and has a priority linked to the devices to determine which device transmits first. This is mostly designed for automotive parts and other vehicle related electronics.

3.8.1.1 I2C

One of the most common serial communication bus' is I2C (Inter-Integrated Circuit), which is a synchronous, packet switched, single-ended, bus developed by Philips Semiconductors in 1982. It has become so popular that multiple companies including Siemens, TI, Motorola, and Intersil produced I2C compatible products since their invention. The I2C operates by having a serial data line and serial clock line with both having a pull-up resistor, if any line is driving the line low, another line attempting to transmit data will detect this and wait to transmit.

Prior to I2C, there was Serial Peripheral Interface (SPI), where there is full duplex communication and a master-slave organization between devices. This organization allows for a singular master which connects to one or multiple slave devices. This configuration has been considered the standard for interfaces as it is incredibly simple and allows for fast communication in embedded systems.

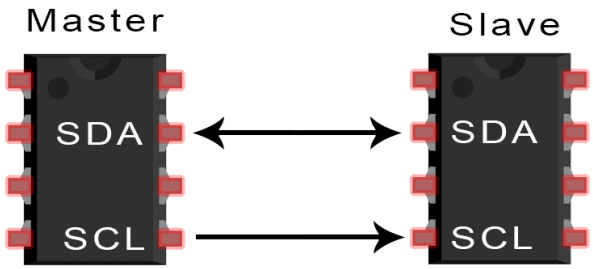

Figure 14 - i2C communication

For the particular use of our project a display is needed to simply show the user the relevant data while also being able to fit all the data in a single frame. Due to the sleep/wakeup implementation the display will not be on for an extended period of time and therefore most of the mentioned display types will be acceptable as it will not suffer from burn in. When considering cost and size, the most relevant types would be either an LCD or OLED display that operates with either i2C or SPI protocol.

3.8.1.2 SPI

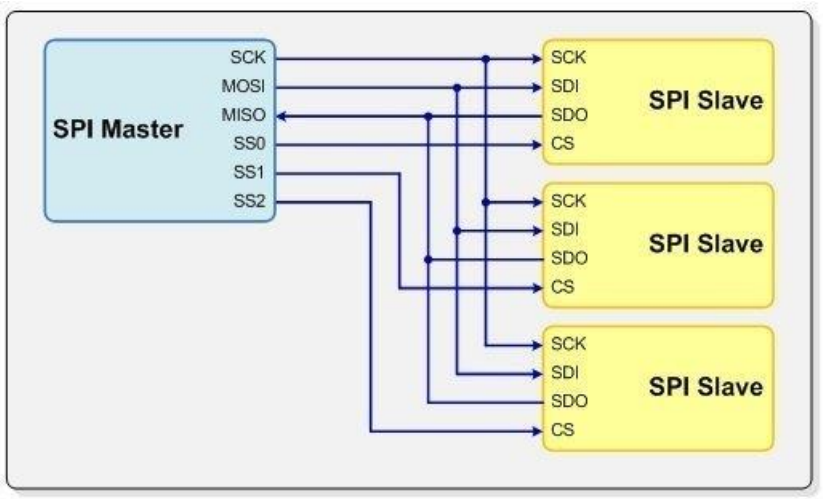

Figure 15 - SPI communication

This mode of communication is not only applicable to the LED component but is used as an interface between nearly any peripheral device and the microcontroller. As stated before, serial peripheral interface is synchronous full duplex master-slave based interface. The data is synchronized from the falling or rising edge of the clock cycle and allows for both channels, the master and the slave to be transmitting data.

In contrast to the I2C protocol, SPI typically has 4-wires which correspond to the clock, master out slave in, master in slave out, and chip select lines. Similar to any other protocol, the clock line refers to the master device that all other peripherals are based off of. This configuration limits the master to only being one device while there are multiple slaves which can communicate with it. Master out slave in refers to the channel where the slave receives data from the master, the device generating the clock. Alternatively the master in slave out refers to the lines where the devices relay data back to the master. Each device has a respective line in these wires. Chip select is how the master communicates with an individual slave, often disconnecting or connecting them to the bus.

In order to start communication between devices in SPI, the master device generates a clock signal from which the corresponding slave is enabled with chip select. Subsequently they are synchronized with the clock edge and data is transmitted to and from the master by the slave device. When writing the respective code for your devices, it is possible to determine whether you want the data to be shifted on the falling/rising edge of the clock, as well as what specific modes you want the interface to operate in.

The ability of full duplex communication compared to the half duplex that I2C offers means that there can be data transmitted to and from at the same time. The half duplex of I2C necessitates that a device must be switched from sending data to

receiving data and that state change leads to slower response times. Overall SPI, although more complex/expensive, often leads to faster data transfer rates and better performance than I2C.

3.8.2 QLED

QLED are often referred to as electro-emissive or electroluminescent quantum dot displays and make use of quantum-dot light-emitting diodes. LED panels that are active-matrix are similar in that they apply current to inorganic layers of particles in order to produce light. These displays are capable of being far brighter than the other options but commercial production of QLED displays are relatively recent and therefore a high price can be expected. Quantum technology has been utilized in order to give LCDs color by printing a layer of QD film and placing it over the LED backlight; these subpixels of the QD film will line up and give color.

3.8.3 OLED

The first OLED was created in 1987 and the successive variations function with the same foundation of layers of organic materials between an anode and cathode. An OLED emits the images by applying voltage across the anode/cathode, furthermore the interaction between positive/negative charges emit light. The ability for OLEDs to be incredibly thin make them desirable when compared to LCDs which can be thick due to the need of backlighting, as OLEDs do not need backlighting since the light is emitted directly on the display's layer. Despite having better image quality, weight, and response time they are more costly than the LCD as there is a limited amount of producers for the substrate.

3.8.3.1 Monochrome 128x64 OLED

The 128x64 OLED display has a readable display of 128 x 64 controllable white pixels. This array allows for any number of patterns, or combinations of activated pixels, to be turned on or off. This flexibility enables this display to be utilized for multiple purposes, such as displaying logos, and pictures, to create mesmerizing art. This display measures 2.42 inches across diagonally, allowing large for display needs to be met.

3.8.3.2 Waveshare 1.5inch RGB OLED Module

This LCD display is 1.5 inches and offers RGB capability while operating only at 3.3V/5V. It can be operated using 4 wire SPI or 3 wire SPI depending on implementation and the display is higher than that of the HiLetgo with 128x128

pixels. Despite being RGB and having a better resolution, it still only consumes a maximum of 41.8mA or 0.0418w. Price: $18.59

3.8.4 LCD

Liquid-crystal displays (LCD) operate by using light-modulating properties of liquid crystals and polarizers. Rather than directly emitting light, LCDs reflect light back to the viewer or use a backlight in order to produce the image. The individual pixels of the LCD screen have a layer of molecules between electrodes as well as polarizing filters. These polarizing filters are what allow light to pass through to the other polarizer without being blocked. Similarly to the polarizing filters, color filters can be applied which allow the use of RGB subpixels that can create complex images. These displays are commonly used in televisions, monitors, smartphones, calculators and more.

LCDs have become the most common type of display used today, this is due to the varying size at which LCD panels can be produced making them applicable to almost anything that needs a screen. LCDs can mainly be categorized as being passive-matrix or active-matrix.

3.8.4.1 Passive-Matrix

Passive-matrix technologies preceded active-matrix and make use of super-twisted nematic (STN), often they are black and white or use color filters as described above in order to create low complexity imagery. Due to the inexpensive cost to produce and reliability, passive-matrix was most common for small devices with little information being displayed.

3.8.4.2 Active-Matrix

Active-matrix utilizes thin film transistors arranged on a glass surface, where the voltage at each pixel is varied in order to turn it on or off. Comparatively to passive-matrix it features a much higher response response time, better viewing angles, and higher refresh rate but at a higher cost.

3.8.4.3 HiLetgo 1.3" IIC I2C Serial 128x64

This 1.3 inch display requires 3-5V DC power and has very low power consumption, only using 0.08w when fully lit. The size is perfect for our use as it can display all the relevant data we need in 1.3 inches while also being large enough to read from a short distance. In order to connect this display it uses a 4 pin connector and installing it is relatively trivial. Price: $8.99

3.8.5 LCD Comparison

LCD	Monochrome 128x64 OLED	Waveshare RGB OLED	HiLetGo 1.3” 128x64 LCD
Size	26.6mm x 19mm	26.8mm x 26.8mm	33mm x 33mm
Resolution	128 x 64	128 x 128	128 x 64
Thickness	6.2mm	6mm	6mm
Weight	4.5g	4g	5g
Connection	I2C	SPI	SPI
Voltage	3-5V	3.3-5V	3-5V

Table 14 - Display Comparison Table

3.8.6 LCD Selection

We have chosen to go with the HiLetgo display as it has the easier install and fits the requirements for our use case. In addition to the ease of installation, this also comes in significantly cheaper than the alternative. It was also determined that the size of the screen is larger than the rest and will enable us to display all of the relevant data to the user in a readable way. The SPI connection is also considered to be adequate enough for our microcontroller and the thickness should not conflict with the design of our housing for the hardware in any large capacity. The improved color variability that OLED provides was ultimately not necessary for our application, should we have chosen to implement a camera to take periodic pictures of the plant an OLED or higher quality picture would be necessary but for displaying sensor data the LCD will suffice.

Figure 16 - HiLetGo LCD

3.9 PCB Design

The addition of a PCB is an integral part of the hardware of the device, as it will be in control of the power regulation and connect everything together.

3.9.1 PCB Design Software

This section serves as to detail potential software choices when it comes to designing PCBs. Multiple factors will be taken into consideration, such as pricing, and familiarity with the software.

3.9.1.1 Eagle

Eagle is a software that allows designers to electronically design a printed circuit board (PCB). Eagle allows users to create basic schematic drawings and place them seamlessly onto a board where they can create traces that will electrically connect components of the schematic together. Eagle is extremely user friendly and fortunately for our team we are all very familiar with it. Eagle is a basic two step process that helps the user along the way to make sure their design is wired and reasonably made. First the user can create their schematic, and then the user turns that schematic into a board layout. The way Eagle makes this easy is their editors and their online libraries as well as components they have ready for you to use.

Eagle has two editors which work hand in hand together making board design super easy for users. These editors are the schematic editor and the PCB board layout editor. Once the board is created online the user can go ahead and order the board and have it shipped to them to solder whatever extra is needed and thus program any parts that need further programming. Eagle also has a PCB library

content which consists of online libraries to browse, and complete components you can also order with your board.

The Eagle schematic editor is equipped with a SPICE simulator, modular design blocks that are easy to drag and drop, and also rule checking to make sure all your electronic components are wired correctly. The Eagle PCB board layout editor is equipped with design synchronization, obstacle avoidance routing, and again design rule checking which helps make sure all your board constraints are chosen and reasonable.

3.9.1.1.1 Importing Libraries to Eagle

Since the team decided to use Eagle as our PCB design software, it is important to describe the process we used to import the microchips we wanted to use into Eagle since they were not already in the library provided. To upload a microchip that is not available in the Eagle library, we must first do a search for the exact chip we need and find a website with the chip's footprint and symbol. The symbol is needed for the schematic design of Eagle, and the footprint is needed for the board design of Eagle. Once we find a correct version of that we can download it to our computer and then upload it to our Eagle library.

Once this is added to our Eagle library we can activate it by making the bubble next to the name green and then we will be able to import it into our schematic design through the Eagle library. This is a simple process which we did twice in our design process for two microchips we wanted that were not shown in the Eagle library. The libraries and their being active for us to use is shown in the figure below.

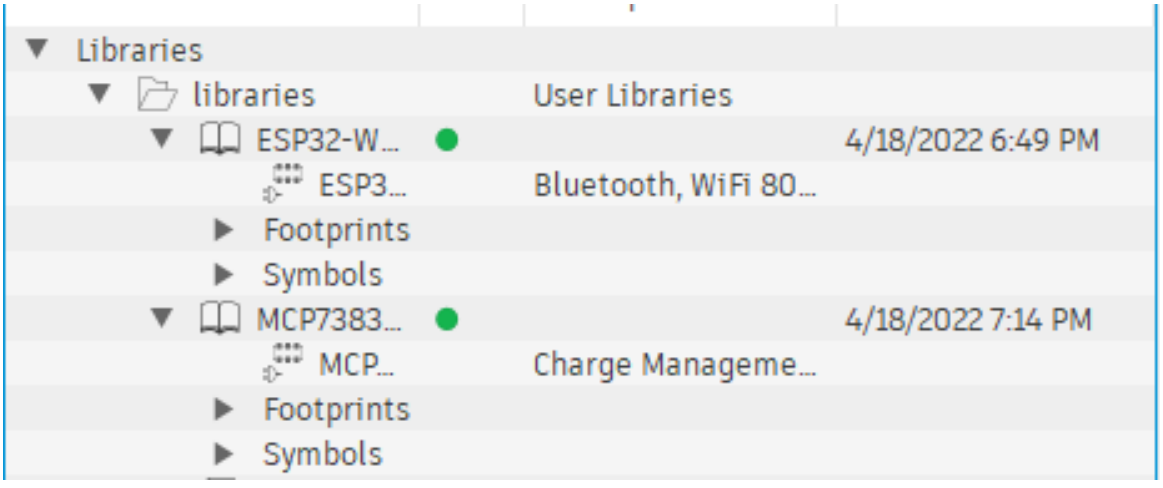

Figure 17 - PCB Imported Libraries

The figure shown above shows the two libraries we imported to our Eagle design for our PCB layout. These libraries were imported because they were not given in the Eagle main library. Luckily through searching online we were able to find a footprint and symbol of the microchips we wanted to include in our design. One problem we came into was that the green bubble shown in the photo was first gray, if you do not activate the library then it will not import into your project. To activate the bubble all you need to do is click it and wait and it will turn green which will allow you to use the library. Having the footprint and symbol of the microchips will allow us to create the PCB design in an easier and more efficient fashion. It is important that the footprint and the symbol is accurate as if we were to manufacture it ourselves, there is the possibility of some connections not being accurately done so when soldering our microchip, it could have missing or even incorrect connections. An example of a footprint and symbol is shown below.

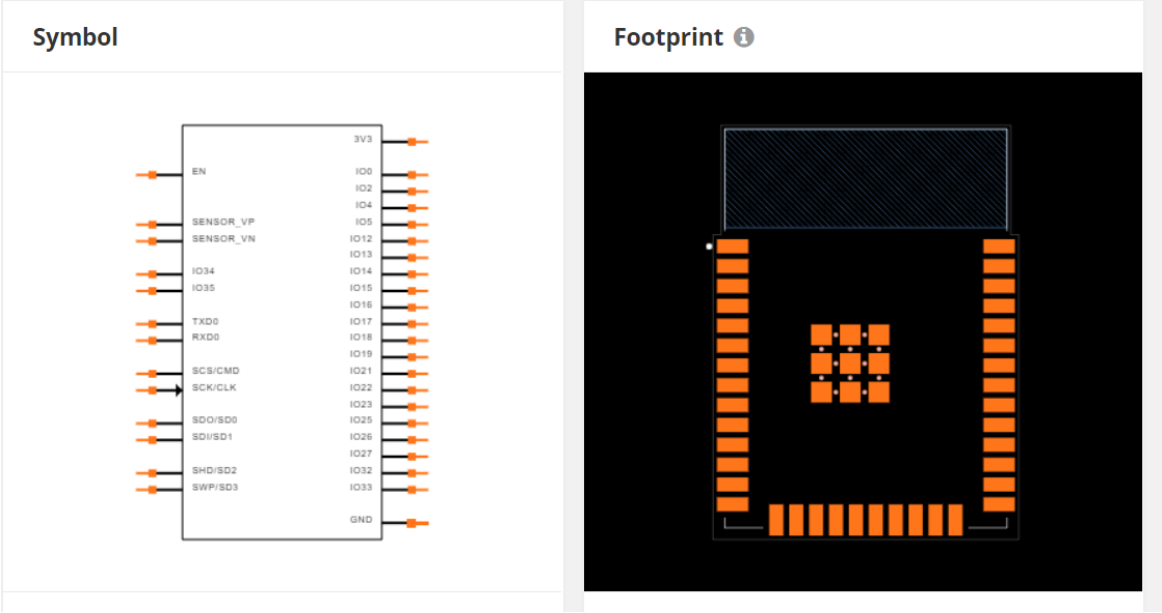

Figure 18 - ESP32-WROOM Footprint and Symbol

The figure shown above shows the footprint and symbol of the microchip we specifically wanted to include in our PCB design. The symbol is used for the schematic portion of Eagle, and the footprint is used for the board layout portion of Eagle. It is important to import both of these to our Eagle library because this allows Eagle to make sure our connections in schematic and board layouts are all there and correct specifically to the board. This will also allow Eagle to put correct measurements on the board where you choose your micro chip to go.

The problem with importing the libraries is that when Eagle helps you set up your bill of materials, it will search for the name of the device which unfortunately limits the search down to only what Eagle has in their libraries. So for this ESP chip and our MCP chip, we will need to do some searching online to find the exact chips we

need and then put them on the board ourselves. This is okay, however putting the chip on the board yourself could be potentially scarring to the board if soldered on wrong. You only have one chance to solder a part onto a board so we need to be extremely careful when putting the piece onto the board.

3.9.1.3 Altium Designer

Similar to Eagle, Altium is another editor for PCB design which includes schematic and board layout. This software allows the user to create their schematic and combine it to its board design. This software includes schematic capture, multi-channel design, library management, simulation, board layout, interactive routing, data management, and fabrication drawings. To use this software, the user needs to subscribe to one of their plans, this goes from $405 per month. This software also allows the user to connect with other users for help or even to work together or just see other works.

3.9.1.4 SolidWorks

Similar to Eagle, SolidWorks is another editor for PCB design which again includes schematic and board layout. This software also includes 3D CAD functionality which can shorten development time and thus improving quality. This software includes 3D CAD, simulations, electrical design, visualization and validation. This software was recently updated so that the capabilities are enhanced to allow the user to work faster and more efficiently. This software is about $9.99 a month which is significantly cheaper than Altium.

3.9.1.5 PCB Artist

Similar to Eagle, PCB Artist is another editor for PCB design which again includes schematic and board layout. This software includes a library of parts, up to 28 layers of board, a controlled autorouter, multiple pages for schematic, and Eagle import. This software is free to use, which makes it better for us and less money to spend. This software also comes with a user's guide and video tutorials.

3.9.1.6 KiCad

Similar to Eagle, KiCad is another editor for PCB design which includes a schematic capture and board layout. This software allows the user to create their own symbols, and has a library with thousands of parts to choose from for their design. This software includes an integrated SPICE simulator as well as a 3D viewer for the board to preview your model. This software is free to download.

3.9.1.7 PCB Software Choice

The reason we chose not to go with SolidWorks is because the team is not familiar with this software which will take up a lot of time to go over. Again there is a membership pricing to use the software so we would be wasting money on something extra for this project. Again, having to learn how the software works will take lots of time out of our design process and thus maybe mess us up in the long run. It is better for us to go with the free version of Eagle and of course we are already familiar with Eagle. It is important to note that our team will be presenting our project in July rather than the term after, so time is not on our side. So being able to cut out learning a new software is a plus for us.

The reason we chose not to go with PCB Artist is because we as a team again are not familiar with this software which will take up a lot of time to learn the ins and outs of the design software. This software is free which is a big plus for us, however since we already know how to use and have downloaded the free version of Eagle, it makes no sense to try and learn this software just because it is free and will take up a lot of our time.

The reason we chose not to go with Altium Designer or another software similar is because you need to pay for their services. Another reason we chose not to use this software is because we had never used this software. Being that we were not familiar with this software made it easy for us to go with the one we were familiar with. Having to pay for software we were not familiar with would have been extra money to spend on this project and could have caused our PCB design to be much longer than this one did. Although this software probably could have been very similar to use compared to one we know how to use, we chose to not spend over $400 on a software we were not sure would even work for us.

The reason the team chose not to go with KiCad, was a difficult decision because this one was certainly one we would have chosen to use if we were somewhat familiar with it beforehand. Unfortunately we decided against it because we knew Eagle and already had it downloaded. This software was free so that would have been a plus for us, however since it was new and we would have to teach ourselves how to work the software we decided against it. Again taking extra time to learn new software when we already have limited time for this project is just not an option for us. Maybe in the future we will learn to use this product as it may have more options than Eagle to buy from.

The reason the team decided to go with Eagle as our PCB building software is because we had all used it before and were somewhat familiar with how to create what we wanted using this software. We had never used any other software for PCB building so we felt we needed to go with what we have used just in case we needed extra help from each other. Eagle also has a free version which is the one we used, so that aspect also made our decision that much easier. If we had to

learn from scratch how to use this software, like the others we would be wasting a lot of time trying to learn the basics and then how to show what we wanted in a schematic and a board.

Since we all learned how to use the Eagle PCB design software from Junior Design class, it helped us all a lot to not only be able to import footprints and symbols we needed, but also show exact circuits and parts and where to even find the libraries or parts that we can use. Eagle is really finicky when it comes to grabbing parts and moving them or renaming, so learning this ahead of time was a big plus for us to know ahead of time instead of again wasting a lot of time learning the software. This also made it easy for us when implementing our bill of materials to add to our cost workup. It was important to not only show the components but how much they were when buying them through Eagles billing software.

3.9.2 PCB Manufacturer

While the design is incredibly important, finding a company that will actually print the PCB and send it to us is also an important part of the process to consider, especially when there are so many in existence. Making sure we balance reviews of the manufacturer, price, shipping times and capabilities is important for all aspects of the project.

3.9.2.1 PCBWay

One very popular PCB manufacturer that's cheap is PCBWay. They are located in Shenzhen China and are known for low volume, low cost PCB printing and shipping. When it comes to the capabilities, we can only order in multiples of 5. Shipping times depend on what we choose, cheap or free shipping means we will be waiting 3 weeks up to a month for our board. Spending a bit more for shipping will get the board to us a lot faster, possibly a week or lower. There's a good reason that PCBWay is widely used when designing and paying a manufacturer to print the PCB, but there are plenty of other options.

3.9.2.2 OSHPark

OSHPark is a US manufacturer of printed circuit boards, and because they are a US company, there is a lot of draw to this company for people developing in the United States. Shipping cost and times are lower, but actual cost is a bit higher for the actual manufacturing part of the process. The team would like to keep manufacturing in the states due to all the problems from COVID when students were unable to receive any parts from overseas. This is an extra precaution we would like to take if possible, but overall OSHPark is definitely a good competitor.

3.9.2.3 JLCPCB

JLCPCB is another Chinese PCB manufacturer that is well known. JLCPCB definitely comes back in terms of the cost for shipping, as it follows a similar pattern to PCBWay. Free shipping means the order will most likely take a month while fast shipping can arrive in a week. Generally, it seems that JLCPCB is also noticeably cheaper than its competitors which is a large plus for this choice of manufacturer.

3.9.2.4 PCB Manufacturer Choice

All three of these are great choices for what we want to do, they are generally cheap and allow us to quickly acquire multiples of our PCB design just in case. Shipping costs and times are fairly comparable and so are the actual costs of the PCB design and creation itself. However, one seems to be a good middle ground and that’s JLCPCB. With low actual costs as well as cheap and fast shipping on top of a good reputation, it was definitely the best choice for us and what we want to do with this project. It is important for us to choose the option right now that has cheaper and faster times because there is a possibility that our first PCB that we end up getting does not work. If this does happen we need to be able to order a new one super fast and get it in adequate timing.

4.0 Design Constraints and Standards

Design constraints relative to the SmartPlant Hub will be discussed in this section. On top of this, the standards relative to the components and general technology used in the SmartPlant Hub will also be discussed.

4.1 Constraints

Constraints relative to the software and hardware design are discussed in this section and taken into account in the design process. Constraints come from every aspect that affects our lives when dealing with any product.

4.1.1 Economic Constraints

The biggest constraint in terms of economic constraints is the general funding of this project. The group has not received any outside funding from companies or others, rather, we will have to fully fund the parts and testing for this project on our own. The reason this constraint has a big impact on the design process is that we will be required a lot more to test sensors and parts sparingly, and try to narrow down what will work as best we can before ordering the parts. This means more time needs to go into research instead of actually building the device itself.

Another constraint we need to consider is the idea of marketability. This product is not meant for businesses or companies, but the average consumer. If we would like this product to eventually be marketable, we need to make the device as cheap as possible while maintaining the specifications we want. If having the device be a bit more expensive will allow us to make a more stable and reliable product, then that will be worth it so long as the final product is affordable for consumers who would want it.

We ran into economic constraints numerous times as price was an important factor when picking our components. This is especially true when economic constraints cross paths with the manufacturing constraints. If we want this product to eventually be marketable, we will need to make sure that the components are available in bulk at a decent price. When deciding on the microcontroller, we already made a decision to go with a cheaper option that did what we wanted it to do without unnecessary aspects to it that would end up not making much of a difference in the end anyway.

4.1.2 Time Constraints

Senior Design students are expected to finish documentation and prototyping for the entire project in the span of two back to back semesters. One of the biggest time constraints for this project is the fact that we are going to be entering Senior Design 2 in the summer. This means that we will actually have a bit less time compared to if we were to finish Senior Design 2 in the Spring or Fall semesters. Although it is not a huge amount of time difference, it means we will have to make sure we are taking it into consideration while designing a project which is both functional and considerable in what it offers.

There must be a consideration of parts that are both in stock and with reasonable shipping time to receive them as well as test/implement them. Considering the short time period of summer term, should we fall behind on our design schedule, it will have large consequences for the second semester. In doing so we must also try to take into account what is reasonably possible in the given time-frame.

Time is the most important constraint to work around, in all honesty. Working on building up from a basic prototype into adding additional features is almost entirely based on how much time we have left after the initial design is completed. It also is constantly taken into consideration as we need to make sure we meet deadlines at every turn.

Another specific aspect of this project where time constraints need to be taken into consideration is the development and purchase of the PCB. Due to the fact that we will have to send the design to a manufacturing company and have them make and ship it, we need to get this done as early as we can. This is especially important because there will most likely be something we need to change with the PCB, and we need to make sure that we leave enough time for everything else while also considering the fairly long wait times for PCB shipping, which can take a week or more.

4.1.3 Environmental Constraints

Working with the environment in mind is a critical aspect of engineering nowadays. We are more invested in saving the environment than ever, and for good reason. The biggest consideration we can make in terms of the environment is our energy usage. Implementing ways to include renewable energy will be an important constraint to work with. On top of this, reducing waste is an important environmental constraint as well. If the user has to discard batteries or any component, it will be a strain on the environment that otherwise would not have occurred. We will strive to work within these two big environmental constraints and create a design that works with them.

The ways in which we will be working towards these constraints involve the power system setup we are using as a whole. Utilizing a rechargeable battery means we will not have as much waste in terms of batteries. We will also be using a solar

panel option to help improve this as well by increasing the usage time of the batteries as a whole, giving them a much longer lifespan.

4.1.4 Social Constraints

The biggest aspect of this project that involves social constraints is working on the design with the idea of being readily affordable and available to everyone. Even though this project involves a marketable product, working towards making it available for people is important. We will be working on the design with this in mind. One of the biggest aspects is making it affordable, but making it available to those who do not have a smartphone is important as well. Although this would mean they have less features and freedom to view readings, they would still be able to utilize the base functions.

4.1.5 Political Constraints

Thorough research into any possible consideration relative to political constraints has led us to believe that there are no political considerations for this particular project.

4.1.6 Ethical Constraints

Especially due to the fact that this product will attempt to be marketable, the ethical constraints surrounding lower quality components are taken into consideration in the design. Quality and longevity will not be sacrificed to cut corners in cost, unless it is decided that the cut will not significantly impact the final product. On top of this, we will make sure to research concepts or designs similar to ours and give credit where it is due in order to not breach any protected intellectual property.

4.1.7 Health and Safety Constraints

The biggest health and safety constraint that needs to be taken into consideration during design is relative to general electronics and power safety. Ensuring that the device is well put together and we use batteries and a power connection that are secure are important when designing and creating. Ensuring the power supply has the lowest possible chance of malfunctioning is essential as well, due to the slight danger posed by batteries exploding.

The other constraint that needs to be considered is the quality and composition of every component including the casing. We will make sure to use non-toxic and safe to handle parts in order to lower the risk of a user having any type of problem

relative to their health while setting up or using the device. This includes the material we will be using for the housing of the device.

Another health and safety constraint that needs to be considered is that at any moment in time one of the team members could get sick. Nowadays that can range from just a cold to the COVID virus. We are lucky that for this paper we were able to do the majority of the work online together, however when it comes to creating our project we will need to meet in person. If one person gets sick we may all need to get tested depending on if anyone was exposed during the time of the last meeting. Luckily now the time for sickness if you have COVID is shorter, however that is still a week or so that one or more members of our team could potentially be out. We hope it does not come to this, however it is definitely something that needs to be taken into consideration during this project.

4.1.8 Manufacturability Constraints

The biggest manufacturing constraint that the group needs to take into consideration is involving the component selection process. Ensuring that the components selected are readily available and in production is important in ensuring the manufacturability of the SmartPlant Hub as a whole. This comes into account especially when considering the fact that we would like to make sure this product is able to be mass manufactured. If a component is rare to find, it will not allow the final product to be fully manufactured commercially.

The other aspect involves the casing intended in the design. Materials for commercial casing tend to be a bit more expensive for individuals to obtain. Due to the ease of use and cheapness of 3D printing, we will be using 3D printed parts for our casing. These 3D printed parts will be made from the PETG filament, as it is readily available and fits our needs for this device as a whole. Due to the fact that the group already has easy access to a 3D printer, it means that there is no delivery limitation to parts used in the case and prototypes can be produced fairly early on.

One large consideration is the printed circuit board which must be ordered from a manufacturer. There are time constraints regarding its manufacture which must be taken into account such as the country of origin, namely the US or China. Should the chosen PCB manufacturer be in China, there will be a lower cost but increased shipping time and possibly production time. Comparatively, a manufacturer based in the US would likely see faster production times but at a heavily increased price. Even between the manufacturers in the same country there can be varying pricing and delivery times. The price for manufacturing is highly dependent on the size of the printed circuit board and the amount of layers necessary. Such manufacturing constraints involving the circuit board production is incredibly important considering our emphasis on being low cost.

We already saw an impact of this when it came to selecting a microcontroller. Due to the fact that we want this project to be potentially commercialized, selecting a microcontroller that can be bought in bulk and that is available for us at a decent price is important. Raspberry Pi boards tend to be more for individual purchase for small projects, and while this is good for our base purpose, the ESP32 was simply the better option at a better price and also can eventually be used in mass production.

Another thing to consider when choosing an overseas manufacturer is the possibility that another “COVID event” will happen. We saw it in 2020 when students were unable to receive any parts or PCB in a timely manner due to this pandemic closing everything down. If we choose to use an overseas manufacturer we will make sure to order as soon as possible and if anything goes wrong we will cancel and go with our US manufacturer. It is unsettling that we even may have to consider this due to a national pandemic but unfortunately this is what times have come to.

4.1.9 Sustainability Constraints

Sustainability is an important consideration when creating a product, especially one that will be exposed to the elements for any period of time, as the SmartPlant Hub will. We will need to consider the constraints involved, such as how it will hold up against the wide range of temperatures from the northern to southern United States. On top of temperature, another concern is weather, especially rain and wind. When designing the SmartPlant Hub, specifically the casing, we will take both of these constraints into consideration and make sure the components are not damaged by these conditions.

Short term component protection is only one aspect of sustainability. Another large constraint involves the long term sustainability of the product. The team will ensure that the components selected are able to withstand the short term conditions discussed for a long period of time. The biggest parts that this constraint impacts are the casing and any sensors that are outside of the casing such as the light and moisture sensors.

When taking these constraints into consideration, the biggest thing that we need to consider is housing. The case needs to be able to protect the components at risk from the weather mentioned. We will be using PETG filament and waterproofing the casing to ensure that the product does not succumb to the elements short or long term. This filament will also need to withstand different temperatures and maintain its protection regardless of the outside conditions surrounding the device.

4.1.10 Travel Constraints

Although the team all attend classes at UCF, not all of us live in Orlando. Travel is a constraint to our project due to the fact that not all work can be done over the internet. We are able to finish this paper while working together online, however when it comes to building the actual device we will all need to make the drive to campus to meet up and work on the device.

Due to the travel concern, it is important to also take into account that our travels will be paid for out of pocket. For example, one team member lives 42 miles or 50 minutes away from campus. That is almost 100 miles round trip per every meetup. This means after three meetings, there will need to be a gas stop which also adds up money wise especially today. This also does not take into account that we will be driving other places in between these meetings. Unfortunately due to the fact we need to drive far to meet each other, and gas prices are extremely high right now, this may be a reason for less meetings. Because of travel constraints our team will need to be cautious of when and how much work we get done together at each meeting so as to not waste valuable time and money.

4.1.11 Scheduling Constraints

It is important to note that scheduling meetings in person after taking into account the travel constraints will also be difficult when considering each team member may have a job or family to also take into account. Not only making a long trip to work together is a day event, but also taking into consideration the fact that we all have jobs in order to pay our bills, some of us have two jobs in fact. We all live off campus therefore bills come out of pocket and this adds up when we also add the fact that we live on our own and not with our parents anymore.

We will do our best to accommodate everyone and again be extremely cautious of when and how much work we get done during our meetings so as to not waste any valuable time and of course lose money by potentially having to miss work and thus not be able to make car payments, or phone bills, or even worse rent.

4.1.12 Personal Constraints

Due to the fact that this is not a single person project, we have to consider that there is a possibility something could happen in one person’s personal life to cause them to miss a meeting or need to take a day off from working on the project. We understand that things happen and personal things are important and sometimes that person may not want to talk about what is going on. The team will do our best to accommodate everyone and try to be as understanding as possible depending on the situation per that person. We hope that no one has any issues throughout

the next few months of work, however we must still take into consideration that anything is possible.

4.2 Standards

Standards created by governing bodies are essential to consider when creating any project, especially one that may be marketable in the future. There are set standards involved when it comes to electronics and power supplies as well as wireless communication. All three are key components of the SmartPlant Hub so their standards need to be discussed and considered before and during the design process.

Some of the standards we will be using will be the standards developed by the IEEE, which is the Institute of Electrical and Electronics Engineers. The IEEE is a leader in developing standards for technology. Other standards will come from the Bluetooth Social Interest Group, the IPC and the National Institute of Standards and Technology.

4.2.1 Power Supply Standards

We will be utilizing a few set components within our system in terms of powering the device. We will utilize a Lithium-Ion battery pack with an optional connected solar panel that will be used to passively recharge the battery pack if the user decides to do so. Therefore, we will need to not only consider the standards involved with powering a small system with batteries, but also with the use of a photovoltaic system to recharge those batteries.

4.2.1.1 1725-2021 - IEEE Standard for Rechargeable Batteries for Mobile Phones

Our power supply will consist of a rechargeable Lithium-Ion battery pack, and the IEEE has developed standards for non-stationary battery packs. The document 1725-2021 - IEEE Standard for Rechargeable Batteries for Mobile Phones covers rechargeable Li-Ion battery packs. Although this document states it is for mobile phones, it also covers any non-stationary battery packs as well, which is what we will be using. When discussing the rechargeable batteries, it is noted in the standard that lithium batteries are sensitive to overcurrent and over temperature conditions caused by shorting or runaway charging.

Such relevant situations can result in overheating of the battery and ultimately a system failure. The standard details how accidental short circuits can occur when metal objects bridge the terminals of the battery pack, therefore it is a necessity

for the battery pack of our device to be enclosed and secure such that we do not short our printed circuit board which it is connected to. Sections 6.4-6.8 of the IEEE 1725 details the battery pack safety considerations

When considering these standards, we will need to take into account any part that advises on safely utilizing this pack. We will ensure that the pack is kept secure without the risk of unintended damage and also make sure that all connections are secured and wires are insulated.

4.2.1.2 IEEE 1526 Recommended Practice for Testing the Performance of Stand-Alone Photovoltaic Systems

Although optional, we will need to take into account the standards involved in including a solar panel in the system. This standard developed by the IEEE covers the testing and performance when using a stand-alone photovoltaic system, which includes utilizing any number of solar panels to charge a battery that powers a system. We will strive to take these standards into account when creating and connecting the system to power.

4.2.2 IEEE 802.15.1-2 and abridged BLE Standards

The IEEE 802.15 working group covers Wireless Personal Area Networks (WPAN). There are 10 major areas within this working group, some are active and some are not. Our project will involve this in some capacity with Bluetooth Low Energy.

Specifically, 802.15.1 (task group one) contains the standards for general Bluetooth technology. It covers and defines the physical layer as well as Medium/Media Access Control (MAC) which is a sublayer responsible for controlling the hardware that interacts with wired/wireless transmission. 802.15.2, on top of this, covers the coexistence of WPAN with other wireless devices such as the other aspect of our project, the Wireless Local Area Network (WLAN). Due to the fact that we will be using WiFi and BLE, this set of standards works perfectly with what we intend to design.

The IEEE 802.15.1 also has an abridgement added more recently that handles the standards relevant to Bluetooth Low Energy being used in Internet of Things devices. Our device will include this technology, so this abridgement on top of the original Bluetooth standards should be sufficient for the purposes of our project, at least for the use of BLE.

4.2.3 Bluetooth SIG Standards

Although the IEEE 802.15.1 was the standard for Bluetooth during its inception, the Bluetooth Special Interest Group (Bluetooth SIG) has since taken over the standardization of Bluetooth devices. The IEEE has not maintained the 801.15.1, and therefore today that is replaced by the SIG standards. This will be taken into consideration, and although the SIG standards are the basis for Bluetooth devices today, the IEEE 802.15.1 still has roots in the inception of Bluetooth as a whole.

4.2.4 IEEE 802.11 Standards

On top of the previous standards involving WPAN, the IEEE 802.11 covers the physical layer protocols and Medium/Media Access Control (MAC) for Wireless Local Area Network (WLAN) communication. Although we're using BLE for close-range access to the device, we will also be using a WiFi connection for long-distance readings. This means we'll need to consider these standards as well. IEEE 802.11 uses various frequencies including the well known 2.4GHz and 5GHz. The plant monitoring device is advertised as featuring a mobile app which pulls relevant plant information from a database, it will be updated using WiFi when the range requires it. From following the standards outlined in IEEE 802.11, we will be able to deliver a fully adequate product that is sellable/marketable.

This standard was created in 1999 as standard 802.11a but only related to standards with a wireless network bearer operating at 5GHz in ISM band with 54Mbps data rate. There were successive iterations over the following decades, such as 802.11b that took from the foundation of 802.11a but was able to make it much more commercially viable. Namely, 802.11b was using the same method for access as 802.11a but delivered a much higher throughput as well as a far lower price. Due to this, the new iteration became the standard for wireless LAN technology.

The successor to 802.11b was the 802.11g which utilized OFDM technology, orthogonal frequency division multiplexing, in order to deliver a 22 Mbit/s throughput. The hardware was even made to be backwards compatible with 802.11b but due to this will experience issues from previous iterations which reduced the throughput. The throughput which is described in the standard amendments are not reflective of the actual throughput a user would experience.

Due to the transfer of packets from a wireless source to often a wired connection (ethernet), there is a difference of header lengths and therefore the throughput would be limited by the speed at which the packets can be converted and unpacked. There is an added layer on top of this as the speed at which the application supplies the data to the connection will also result in varying

throughputs, often the throughputs which are described in the standard are ideals of what can be achieved but not reflective of reality.

Although 802.11 utilizes and covers various frequencies and devices, we can focus on one extension specifically. Our chosen microcontroller actually runs on the 802.11b standard specifically. This means it runs at up to a throughput of 11Mb/s at the standard 2.4GHz.

4.2.5 IPC-2221 PCB Standards

Since we will be utilizing a PCB for this device, we need to consider the PCB standards that IPC has put forward. The group IPC-2221 created standards for PCB design, at least in the generic sense. Although this group develops standards for generic physical PCB design, there are supplements that add to the set of standards as well. These supplements include IPC-2222 through IPC-2227, specifically IPC-2222 covers rigid physical PCB design and IPC-2223 covers flexible physical PCB design. The standard goes into details regarding bus layouts, component mounting, etc. for each type of PCB. Therefore, this set of standards is necessary to take into consideration during the design of the PCB itself including mounting and materials.

4.2.6 NISTIR 8059 Standards

NISTIR 8059 is the Materials Testing Standards for Additive Manufacturing of Polymer Materials by the National Institute of Standards and Technology. Additive Manufacturing covers 3D printing, which is necessary for our project as the casing manufacturing process. We will need to keep these standards in mind when creating the casing, as we will be using a polymer material. One specific standard that the NISTIR 8059 covers is the mechanical properties of parts used in Additive Manufacturing. According to the standard, 56% of the materials used in additive manufacturing are photopolymers while 40% are thermoplastics, the remaining fraction is cut between powdered materials. It also aptly points out that manufacturers do not follow a convention when it comes to the mechanical properties of their materials. Often the technical data sheets for these materials are missing key components, such as optimal printing parameters for the material. Other properties of the materials which are not normally provided include molecular weight and distribution, dispersed phase volume concentration, viscosity, crystallinity, and void content. The standard also concludes that by ultimately not providing the full scope of the materials to consumers, subpar performance will be yielded. So, we will be using materials and a printer that follow standards and will also keep in mind the standards involved in the process of design as a whole.

Taking this standard into consideration relative to what we've chosen as our material, we've chosen PETG for our material. This material is actually considered the safest material in terms of 3D printing filament as its fumes are harmless. PETG is actually used in the manufacturing of water bottles and is considered food safe. This filament follows the standards necessary and this works out in our favor, as it is also the best choice for our design due to its properties.

4.2.7 Software Standards

The ISO/IEC/IEEE 29119 software testing standards is intended to create an agreed upon set of standards for any organization to use when doing software testing. The standards are separated into multiple parts that describe the process for which software needs to be tested. In part 1 of the document the vocabulary is defined for further use in the standard and gives adequate examples for organizations to understand. This allows members of a given organization to be on the same page when referring to the software and clears up any confusion in the development process.

4.2.7.1 Test Processes

Due to the varying types of software development that are used by organizations, creating a set of standards poses a challenge. In part 2 the ISO puts forth a general process for which software testing can be done. It does so by outlining multiple methodologies for testing software that differ in their process. One such test process that is detailed in the standard refers to the Organizational Test Process, its purpose is to develop and maintain organizational test specifications. The reason for this process is in order to meet stakeholder requirements and expectations while making sure that they have access to the specification throughout the software testing.

Another such concept that is discussed in this part of the standard is test monitoring and control. This refers to how project managers are meant to overlook the testing process and the steps they take to ensure it is done successfully. In regards to our project, this test processes section is important in order to understand how we should approach the testing/debugging phases of our various sensors and systems. By following this we can hopefully reach a software standard which is in accordance with the industry standard.

4.2.7.2 Test Documentation

This portion of the document refers to how an organization should be recording their software test documentation and includes templates/examples of what should be produced while testing the software. The standard divides the documentation

into Organizational Test Process, Test Management, and Dynamic Test process. These are the overarching process levels which were detailed in part 2 of the standard and refers to what data, specifications, and results are necessary for proper documentation.

4.2.7.3 Test Techniques

While part 2 refers to the overarching processes, part 4 gives adequate examples and definitions for particular design techniques that can be used to test software. One such technique is called specification-based test design, where the software tests the specifications of the system. In our case such a specification-based test would be checking whether the device wakes up in specified time from our requirements. Another could be how fast the user is displayed data from the database upon opening the app.

4.2.8 Group Coding Standards

In order to make the code readable and easily debuggable, it was agreed upon by all team members that it would be partitioned. By doing so we intend to isolate each of the sensors/systems in order to make the debugging process simple as by having all of them in a singular function could make the process tedious. It is also important that we follow a similar standard when writing our programs as we will all be working on separate parts which must be compatible in order to implement a functioning product. There is an agreed upon standard for what formatting, commenting, and naming conventions. By doing this it will make our code readable for other group members should they need to debug or help in the process of development.

4.2.9 C Language Standard

The ISO/IEC 9899 details the form and interpretation of programs that are written in C or expanded language of C. This standard is essentially the definition for creators on how to compile your code. This standard is incredibly useful as our microcontroller will use the Arduino Software IDE which uses C/C++. It goes into detail of the syntax and constraints of the language, how data is input and output, as well as the restrictions and limitations that are the result of using C language. The standard is similar to that of the software testing standard in that it includes an entire section dedicated to creating definitions for the rest of the document. It then describes multiple environments that are used to execute C programs and the important aspects of them, namely the Translation environment and Execution environment.

Being that learning C code is similar to learning a new language, one can argue that this standard is like a teacher. What this standard does is basically teach a student the ins and outs of how to write, understand, and of course compile a C code. The importance of C code in specific is that every other coding language out there today borrows rules or tricks from C language coding standards. It is important for a student to learn and know C coding language and its standards before moving on to other languages, which is what we have done in school ourselves. Since we have this standard to go off of, a student could use it as a tips and tricks guide to help them master C language and thus possibly create more and more coding environments.

Further in the document there is an entire section detailed on the multiple expressions used in C and gives proper examples of how each operates. Perhaps the most useful section of the standard is that which describes the C libraries. This standard goes in detail about floating point extensions, coding rules, embedded C code, bounds checking interfaces, allocation for dynamic functions, decimal floating point, mathematical special functions, and new character types. It is very important to understand the C standard code when coding in C because the C standard basically explains why the compiler does something, or why it is not compiling code.

This standard also goes into detail about new things added which can also be helpful to us. Since this standard goes into such detail about all of C code, this section serves as a more understandable documentation on what is included in the library and how you could use them. This standard can help our team put together a useful code and help debug any problems we may come across.

Since our team is only just finishing college, this standard goes into so much detail we are not taught by professors or through book or school work. It could be very helpful to look through online resources that talk about this standard and how C language works for us to better understand the code we are writing and why our code acts the way it does, and in return maybe learn a few new tools from the standard that can help our code. One of the most important things to learn from this standard is the way the compiler decides to sequence the compiling. Especially if we come up with a length code, it could be extremely important for us to go back and make sure we do not miss any marks. It also goes into syntax which can sometimes be overlooked when coding for hours on end. Overall this standard could be arguably one of the most important for us to follow.

4.2.10 Standard For Sensor Performance

The IEEE provides a common standard for sensor performance which comes in units, limits, terminology and conditions. This is mainly for digital I/O interfaces, which in our case will include the humidity sensor, temperature sensor and light sensor. This standard goes into depth about performance parameters or each

sensor which allows them each to go into market. These performance parameters must be included for buyers to see, so for our project this should not be too much of a problem since it is more on the marketing and creation of the sensors end. However, it may be helpful for us to see some of these performance standards of some of the sensors we have purchased.

5.0 Design

Following the extensive research done in our part selection and technology research, we have been able to narrow down our project into more definable subsystems. This section serves to define these subsystems and even go into detail on how they are to operate and what technologies we plan on employing to meet our requirements. The descriptions described in this section are not the final version, as it may change due to unforeseen circumstances as a result of testing and prototyping. There will be a section further in the document which will describe the final design structure once all testing has been done.

5.1 Hardware

The system's hardware components comprise of an array of sensors including but not limited to temperature, humidity, light, and moisture, as well as various LEDs which will serve as current status updates to the end user, buttons that trigger on and off states, as well as a self contained solar cell power system, with a battery to hold and maintain power to the overall system.

5.1.1 Breadboard testing

After all parts arrived from the respective suppliers, testing of the individual components for compatibility, as well as state of the parts, was necessary to ensure that all portions of the hardware side would work, as well as work together. This section will cover the breadboard testing of the individual components, along with any notes or comments regarding the parts and their function that was noticed during testing.

5.1.1.1 Microcontroller Testing

As seen above in section 3.2.5, Figure 5, the microcontroller chosen was the ESP32 Feather Board named the Huzzah32. The board was set into the breadboard just for testing purposes. The board is capable of standing on its own, but was easier to push into the breadboard for the rest of the testing. A USB Micro-B to A was plugged into the Huzzah32 and into the testing computer for powering, as well as data transmission between both systems, as seen in Figure 15. The software development environment that was utilized was Arduino IDE. A sample code was used to test the on-board LED. This code was to test the functionality of the board itself, as well as the capability of data transmission between the host computer and the microcontroller. A simple timer was implemented, every 1000 milliseconds the on board LED was flashed on, after another duration, the LED was turned off. The code was uploaded to the microcontroller for execution.

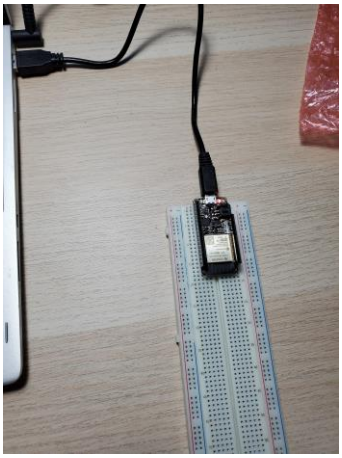

Figure 19 - Microcontroller Breadboard Test

After successfully flashing the code to the Huzzah32, the code executed as planned. After the successful execution, the microcontroller was deemed in perfect working order, fit for all future integrations with all possible sensors moving forward.

5.1.1.2 Battery Testing

As seen above in section 3.4.17, Figure 12, the battery chosen was the 18650 Lithium-Ion Battery Pack. The battery pack contains a JST connector from the positive and negative leads from the cells of the battery. The connector on the battery was the male version of the JST. There is a female JST connector built into the Huzzah32. The Huzzah32 has a built-in charging circuit, allowing for the battery to be charged while the board was plugged in via USB to a host computer, or to a power adapter into the wall. Once the battery was plugged in, the battery was charged. There is a signal LED on the board that flashes while the board does not have a battery connected, and that same LED will maintain the color once a battery is connected, as shown in Figure 16.

After successfully connecting the battery to the microcontroller, the on board LED that signals whether a battery is connected was triggered and maintained the red color. A short amount of time was required to charge the battery then the USB connecting the board to the host computer, which typically fully powers the board, was removed. The on board LED was lit, signaling that the battery was supplying enough power to fully power the system, without the need for external power. With this observation, the battery was deemed in perfect working order, being more than capable of supplying the entire system with power without having additional sources attached.

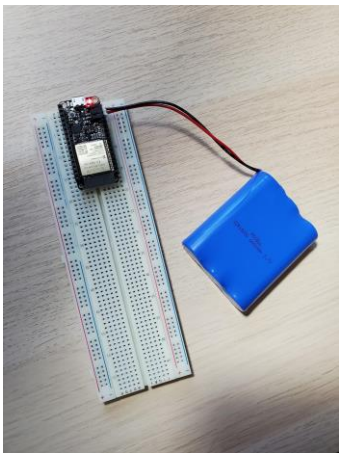

Figure 20 - Battery Test

5.1.1.3 LCD Testing

As seen above in section 3.4.23, Figure 15, the screen chosen was the HiLetgo 128x64 OLED screen. The HiLetgo screen was plugged into the breadboard, and connected to the microcontroller via jumper wires. The HiLetgo has 4 pins, a VCC, GND, SCL, and SDA, which need to be connected to the proper pins on the microcontroller. All of these pins have an exact pin on the Huzzah32 to connect to, as seen in Figure 17. Once connected to the proper pins on both the microcontroller as well as the screen pins, a sample code was pushed onto the Huzzah32 which was meant to clear the display, and print out a very simple test message. The color was set to be white, and the size of the font was set to be 1, the smallest of the possible fonts. The location of the start of the text was set to be the top left of the screen. This test message was set to display for 1000 milliseconds, then the screen would clear, and restart the loop over again.

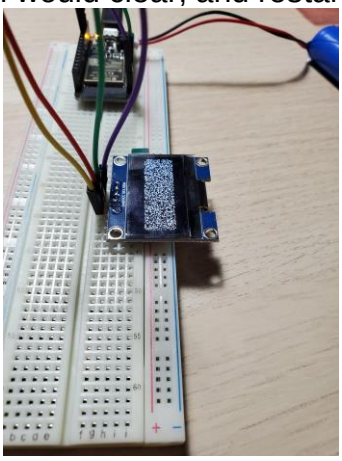

Figure 21 - LCD Test

When the screen was connected to the Huzzah32, the first thing that was noticed was the screen did not clear when it was meant to. The code was modified to only clear the display, and still the screen did not clear. The entire screen was covered in an array of white, unclearable pixels. At this point, it was deemed that the OLED screen that was delivered was faulty. A new screen will have to be ordered, and it is unsure if a different model will be procured.

5.1.1.4 Moisture Sensor Testing

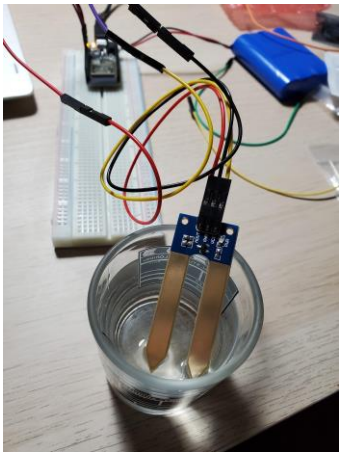

Figure 22 - Moisture Sensor Test

As seen above in section 3.4.8, Figure 9, the moisture sensor that was chosen was the STM32 soil moisture sensor. The moisture sensor was not directly plugged into the breadboard, instead the three leads from the sensor were connected to the Huzzah32. These 3 pins, VCC, GND, and AOUT, had three female pin connectors that a jumper wire was connected to. These wires were then connected to the microcontroller, and then transmitted all of the sensor data back to the program, as seen in Figure 18. The sample program that was utilized showed the numerical output of the moisture level.This moisture level was polled every 500 milliseconds for testing purposes.

After successful connection of the jumpers to the sensors, the data was transmitted from the sensor to the Huzzah32. The moisture level was output to the system as a numerical value. As the moisture level increases, the value increases as well. To test the validity of the results, the sensor was tested in dry conditions, then by slowly dropping the sensors in a small glass of water. As the sensor was dropped in the water, the value increased proportionately. The output of the results proved that the sensor was working as expected..

5.1.1.5 Light Sensor Test

As seen above in section 3.4.12, Figure 10, the LM393 light sensor was the sensor of choice. This sensor came on a board with 3 output pins, a VCC, GND, and a DO. These pins had to be connected to the breadboard, then connected to the Huzzah32 via jumper cables, as shown in Figure 19. A sample code was written and flashed to the microcontroller that read the data from the sensor and provided a very quick numerical output of the data. This data would increase or decrease based on the amount of light that was present in the environment of the light sensor. This sensor data capture was delayed by 1000 milliseconds so as to not overwhelm the Huzzah32 with data.

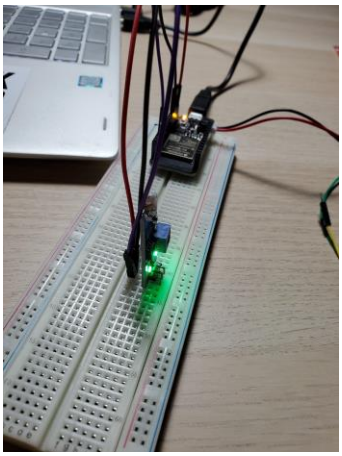

Figure 23 - Light Sensor Test

When the light sensor code was executed, the sensor picked up on the amount of ambient light within the test environment, and sent the data to the microcontroller. This data was then output by the system in proportion to any changes in the environment's light level. A high output light was placed in close proximity to the light sensor and as the light neared the sensor, the value increased as expected. This result showed that the light sensor chosen works as intended, and will suffice when it comes to implementation into the entire project.

5.1.1.6 Humidity Sensor Testing

As seen above in section 3.4.6, Figure 8, the SEN-18364 humidity sensor was the sensor selected. The SEN-18364 has 4 output pins, a VCC, GND, SDA, and a SCL. The SDA is the serial data pin that must be connected to a pin that can input and output data. The SCL pin is the serial clock pin. This pin helps ensure proper communication timing between the microcontroller and the sensor itself. Due to the nature of the pins, the sensor had to be connected to a breadboard, then connected to the Huzzah32 via jumper wires, as seen in Figure 20. Once

connected, the SEN-18364 was told to send data to the microcontroller via a small code that was flashed to the board itself. This code was written so that the board would poll the SEN-18364 every 1000 milliseconds, and receive the data from the sensor. Once the board had the data, the data would then be output to the console terminal via a simple print statement, ensuring that the data was valid and was readable both by the microcontroller, as well as the user. This is overall similar to the end design as the board upon receiving the data would simply send it to our database and update the relevant table before being sent to our mobile app for the user to view.

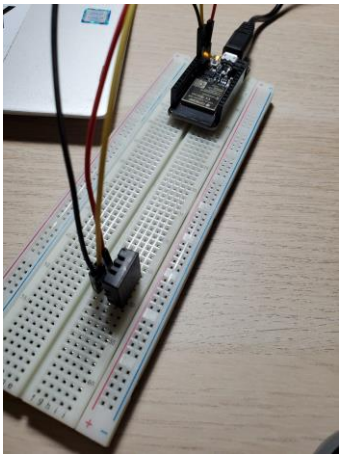

Figure 24 - Humidity Sensor Test

Making the initial connection to the board's pins was tricky, but after small trial and error, a successful connection was made between the two. When the code was executed, the system was able to read the data from the sensor, and output this data to the user. Following this successful test, it was deemed that the sensor was in perfect working order, and would be more than capable to fulfill all future sensor needs as the project progressed.

5.1.1.7 3D Printed Housing testing

Following the printing of our housing, the team has set up a plan to test the durability and practicality of the design. In order to ensure that it will meet all requirements we have, it should be able to house all parts in their respective places while allowing their probes/prongs to adequately get readings from the soil or from the environment surrounding the plant without negatively impacting their accuracy. In order to ensure that the case has no bearing on the accuracy of our data, it is intended to run all sensors outside of the housing and then within the housing, having placed all the sensors in the same spots they should yield nearly exact readings. Should the readings differ in accuracy, it would be necessary to redesign the housing with these considerations in mind.

5.1.1.8 PCB testing

Upon delivery of our circuit board once we have finished the PCB design portion, it will be necessary for us to make sure it serves the intended purpose. Namely the two sensors which are to be placed upon the board must be independently tested to make sure they are operational as well as ensure that the full power integration is correctly implemented. First we would have to connect the battery to our printed circuit board and allow it to connect to the 3.3V regulator we have placed on it as well as the LIPO charging circuit, the current at the end of these particular connections would be tested to ensure that the PCB is accurately performing its job. The chip on the board would have to be flashed with our code to test the two sensors on the board as well as test the pin out from our chip to the microcontroller to ensure that it is receiving the correct amount of power from our board.

5.1.2 Sensor System Flowchart

The flowchart below shows the basic flow of how the sensors will be operating. No matter where you start in this system, a sensor is reading. When a sensor reads a value that is not wanted, for example moisture level is too high, then the system will notify the user by pinging them through the app or lighting up one of the leds. The system will then go again and when the answer is perfect moisture level, then the system will move on to the next.

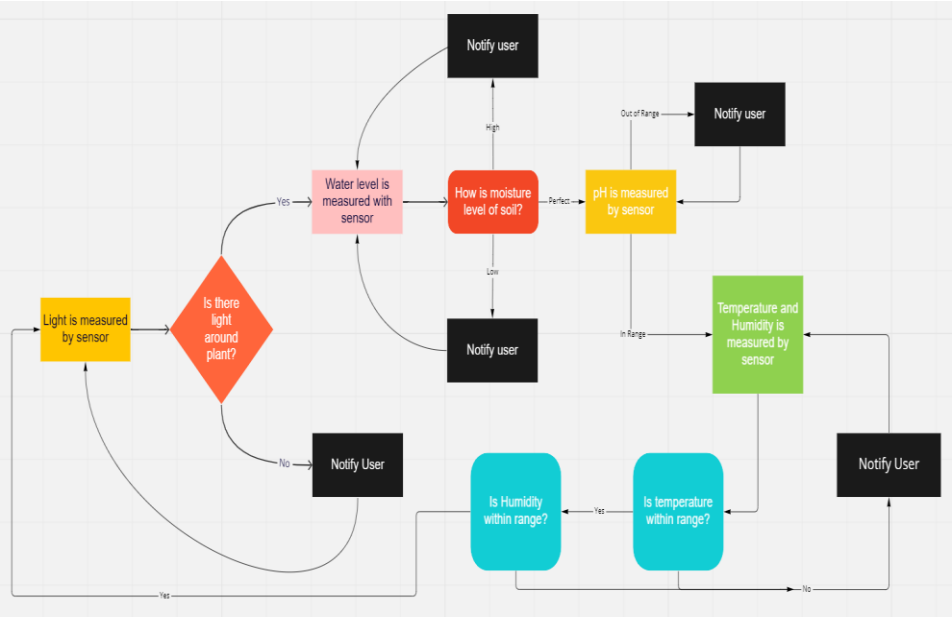

Figure 25 - Sensor Flowchart

The point of the flow being like this is because we want to keep the user notified in almost every aspect of their plant life cycle. This monitor will not only monitor the plant but help and alert the user when something is off. The reason for continuous checking is because at any time, the temperature could have just been checked and maybe five or so minutes go by and now the temperature is a bit too high.

This flow chart does not take into account the times where the LCD or LEDs are used, only the sensor flow. This flowchart also does not take into account the intermittent sleep modes we plan on having to save battery life as well as sensor life. This flowchart assumes that throughout each cycle, each sensor will be consulted as to if their piece is being met, when it is it moves on, when it is not it notifies the user to make a change and checks again.

5.1.2.1 Light Sensor Flowchart

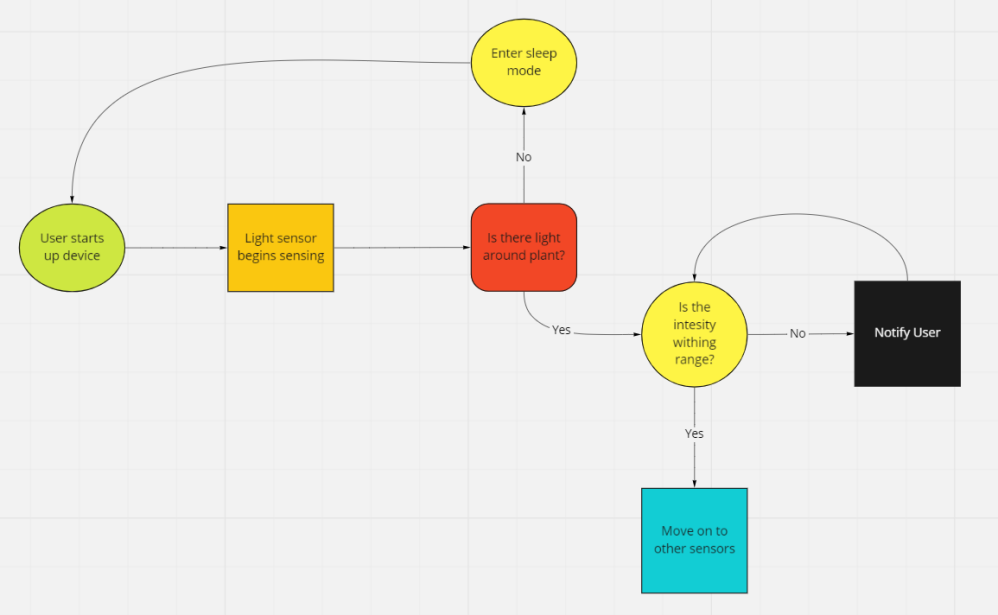

Figure 26 - Light Sensor Flowchart

The flowchart above shows a quick flow of how our light sensor will be working. Monitoring the amount/level of light that a plant receives is vital to its continued growth. The light a plant receives will affect the way in which it is able to produce chlorophyll, which is needed to keep it healthy. Different plants require a varying level of light, typically house plants require a lower amount of light while something such as a fruit requires a great amount of light. Should the fruit/house plant not get the light that is necessary, it will have stunted growth and may even die. The user will start up the device or activate a check on their plant. The light sensor will begin sensing and if there is no light around the plant, the sensor will go into sleep mode.

If there is light around the plant the sensor will then make sure the light range is okay for the plant. If the plant does not have a good light range then the user will be notified and the sensor will then check again. If there is adequate light range for the plant then the device will move on to other sensors.

The light sensor is able to detect light by measuring the intensity around the plant, and does so in lux. Therefore it is important that when reading this data, we take this into consideration before pushing it to our database. This process will start over when the device is stayed on or when the user just turns on the device or activates a check in.

5.1.2.2 Moisture Sensor Flowchart

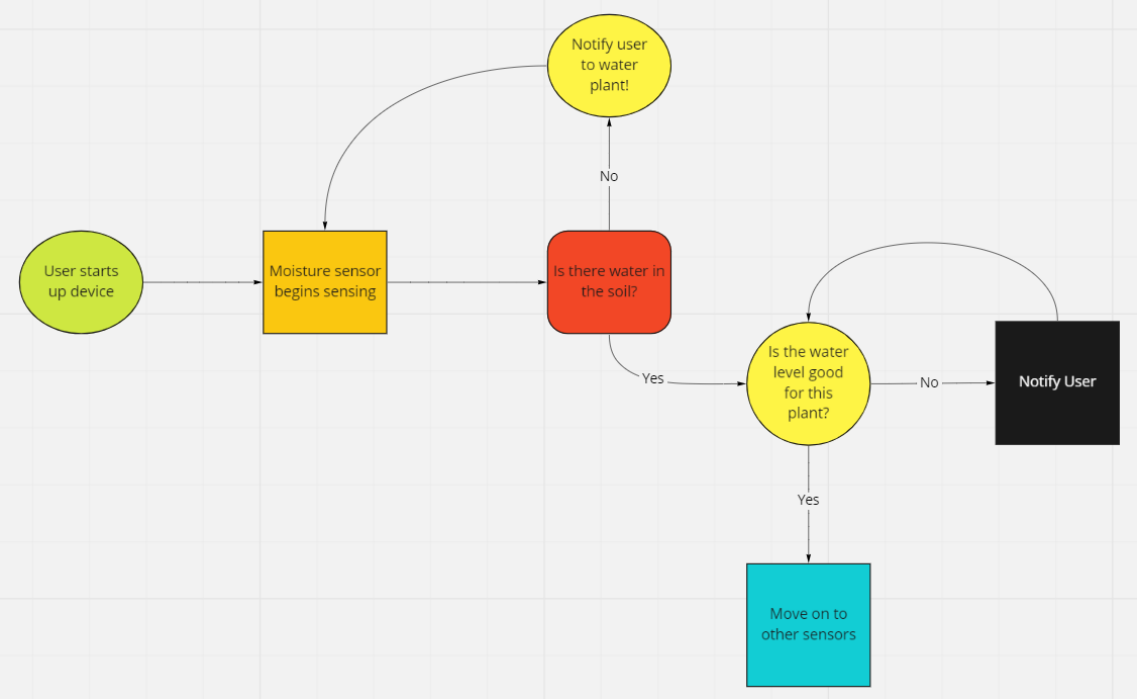

Figure 27 - Moisture Sensor Flowchart

The flowchart above shows the basic flow of our moisture sensor. The user will start up the device or activate a check in and the moisture sensor will begin sensing. If there is no water in the soil the user will be notified to water their plant. If there is water in the soil the sensor will then check if the moisture level of the soil is adequate for the plant. If the levels are not good then the user will be notified, otherwise the device will move along to another sensor. This flowchart does not take into account that the moisture sensor can sense constantly.

5.1.2.3 pH Sensor Flowchart

The flowchart above shows the basic flow of our pH sensor. The user will start up the device or activate a check in and the pH sensor will begin sensing. If the pH is not within range the user will be notified, and if the pH is in range then the device will move on to other sensors. This flowchart does not take into account the fact that the pH sensor can stay on and sense constantly. Our initial design does not ensure that the sensor will stay on and sense constantly however, due to the sleep/wake up functions that will be placed upon all of our hardware. Should the pH sensor be compromised in any way there are currently no considerations taken place as the probes tend to be incredibly durable and sturdy, meaning any misuse for our application would not negatively impact its performance.

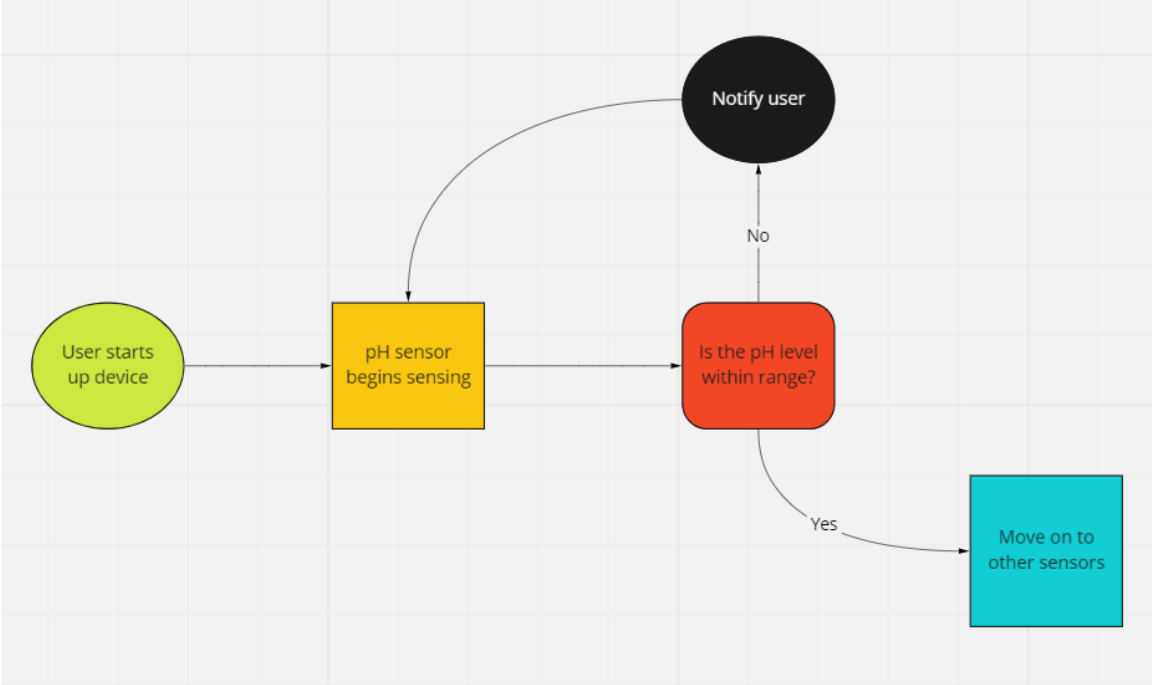

Figure 28 - pH Sensor Flowchart

5.1.2.4 Temperature and Humidity Flowchart

The flowchart above shows the basic flow of our humidity and temperature sensors. Being able to monitor a plant's temperature/humidity is vital in order to ensure that it is growing optimally. By having these two sensors, we are able to ensure that the client is making the right choices such as whether the temperature of the plant's environment should be raised or lowered. The user will start up the device or activate a check in and the temperature sensor will begin sensing. If the

temperature is not in a healthy range then the user will be notified, and upon the temperature of the plant lowering into its target range, the notification will be removed. Alternatively if the temperature is too low, the user will be notified that it should be raised and upon raising to the target, it will remove the notification. If the humidity is within a good range then the device will move on to another sensor.

It is worth noting that for our design, the device is planned to be a small device which gets planted into the plants soil, it is important that these two sensors remain above the soil and unobstructed by our case. Should the sensor be obstructed or within the case, it is possible that there would be incorrect readings which could negatively affect plant growth. This flowchart does not take into account that the temperature and humidity sensors can stay on and detect small changes intermittently.

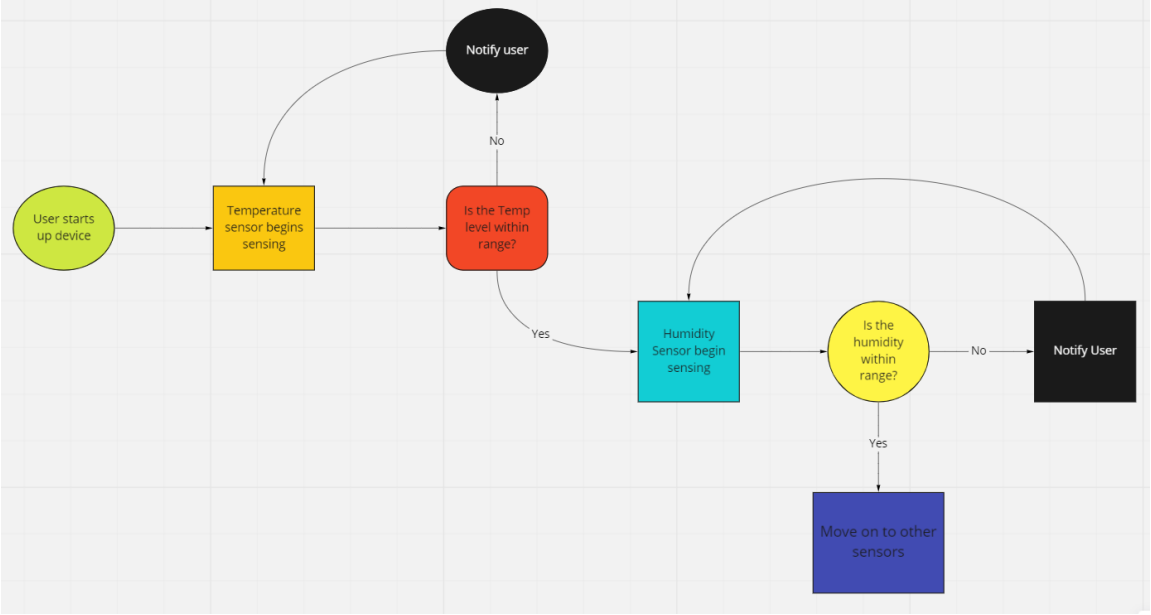

Figure 29 - Temperature Sensor and Humidity Sensor Flowchart

5.1.3 Display System

The display subsystem is shown in the chart below and explained in detail furthermore.

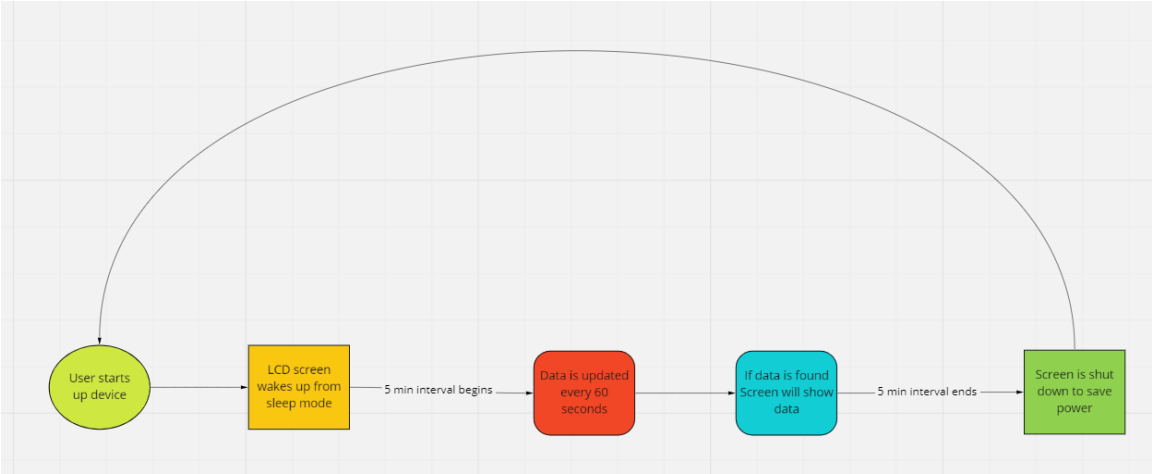

Figure 30 - LCD Display Flowchart

One such subsystem of the device is the LCD display. Although relatively straightforward, certain requirements need to be taken into account such as placement, response time, and interactions between the button, microcontroller, and display. When initially approaching a workable design, the concept of a physical display to forgo the necessity of a smartphone was clear as it is inexpensive and extends usability. Upon a user pressing the physical button, the microcontroller will be woken up from a sleep state, the general consensus is that the device will remain on for a period of 5 minutes and update the data on the lcd every 60 seconds.

In order to achieve this, we must use interrupts and timers using the master signal, microcontroller, clock signal and then displaying the relevant data on the screen. There is already open source code that is available online for the LCD selected and the use of such should make this implementation simple. Upon the end of the 5 minute interval, it is intended for the device to power off entirely. Should the user turn the device on using the app remotely, the LCD should not turn on as to save power. This choice was made after considering that the user is already on their phone to power the device, therefore the need to physically see the screen is redundant.

5.1.4 Overall schematic

This section will provide a full schematic for our board design as well as the schematic for the casing. It will also cover the two together.

5.1.4.1 PCB Design

A printed circuit board (PCB) uses conductive pathways and electronic circuits to connect components. We use a layout software, in this case we will use Eagle, and combine all components in specific placements and then route the electrical connectivity which will be printed on a manufactured circuit board. These electrical traces are etched into layers of sheets that are contained in this board.

We will be basically creating our own PCB for this project to help us with a main goal. Designing the perfect PCB will help us reduce the errors or possibilities of a short circuit in our overall project. This is extremely important because we will have various components on top of our PCB all connected and performing different tasks.

The most common PCB is a single sided PCB which contains a single copper layer in which the components are soldered onto one side and the circuit is visible from the other. The advantage to this type of board is it is not only easy to manufacture but easy to repair as well, and of course easy to design.

Another type of board is the double sided PCB which contains the same layer of copper but not only added to the top but the bottom as well. Because there is substrate in the middle of this board, through hole technology is used to connect electrical circuits through drilled holes to connect metal parts. This gives the board more space to be used for a reduced size of board which can make this a compact project. The advantage of this is that it can be relatively cheaper and more flexible.

The next type of PCB is the multi layer PCB which can include multiple layers of copper. This means multiple holes to connect multiple layers to the other layers. The advantage of this is again the ability to compact the board and also adding much design flexibility.

After choosing the type of PCB that is needed for the goal, we need to go through designing the PCB which includes understanding the electrical parameters needed for the job. This includes building an overall schematic of the board as well as how the overall layout of the PCB is going to look like. The design process of a PCB then includes placing all the components needed in certain spaces and creating traces to and from components, drilling holes or vias where they are needed if there are multiple layers on the PCB.

The overall design of our PCB is crucial to come up with a perfect schematic so that our PCB design and connectivity can be simple and easy which is important for overall improved reliability from the PCB itself. This is important for our project because we hope to use the device for a long time as well as eventually have multiple designs and devices.

The importance of a PCB for this project is to essentially have a second microcontroller without having the same board. We hope to use this PCB to do most of the work for the microcontroller and thus regulate all the power for our

device as a whole. That includes the microcontroller as well as sensors, and of course battery power so that not too much power is being used all the time when the device should be asleep.

Once a decision is made by the team we will begin choosing the correct type of PCB, and thus come up with a plan as to how the PCB will be designed. Our rough thought as to how the PCB will be implemented is that it should control absolutely all power for our device. This should be done by a voltage regulator. Most of our sensors will take in 3.3V power supply, and the enable pin for our microchip we chose needs 3.3V to turn on so we believe that a 3.3V regulator will do the trick in this regard.

We also would like to possibly implement a charging circuit for battery power as well. This should be done similar to the microcontroller we have chosen which uses a lipo charging micro chip inside their board. And lastly we are thinking of keeping our two sensors that need to be housed away from the others on the PCB so that it can also be power regulated and not over crowd our microcontroller.

First we will make a rough schematic and then go over it together, then we will create a final schematic that covers all of our ideas. Then we will convert our schematic to a board layout and then find a perfect place for each component and route traces. We will then make sure everything is correct before placing the order with a vendor for the PCB.

5.1.4.2 PCB Schematic

As a team we decided to create our PCB so that it is able to handle all the power regulation for our device, as well as take care of our two sensors that will need to be towards the bottom of our device due to what it is sensing. We also decided to add a charging circuit for battery power as well. Shown above is our PCB design schematic that we built in Eagle. As you can see we have inserted the ESP32 WROOM chip onto our PCB board to use for our two sensors, moisture and humidity.

Below is a photo of our schematic for our PCB design that was created in Eagle for this project. Fortunately for us, we knew how Eagle worked and did not have to deal with any constraints of how to do things.

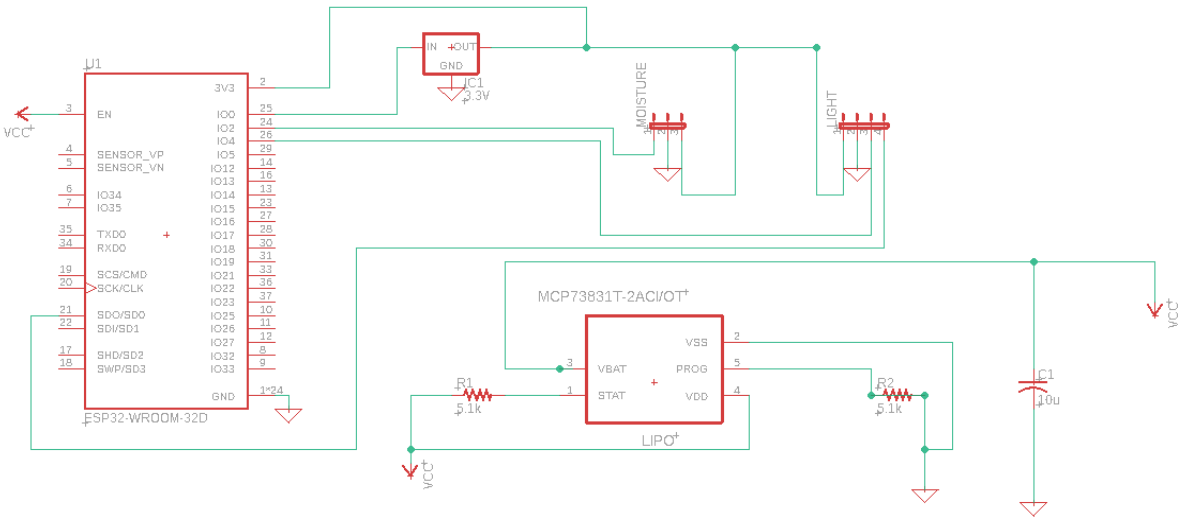

Figure 31 - PCB Design Schematic

We wanted to have the same chip as our microcontroller so that our system can be easily compatible and grouped together. We also have a 3.3V voltage regulator that will control the power that is used on the board as well as the power given to the sensors and when. We have the jumper for the moisture sensor which includes 3 pins, voltage, ground and analog output which is connected to the microchip to be easily programmed. We have the jumper for the light sensor which includes 4 pins, voltage, ground, analog and digital output which is connected to the microchip to be easily programmed.

At the bottom of the board we created a charging circuit for our battery power to charge our boards and system as a whole. This includes an MCP chip that is also used on the microprocessor that we are using as well, again to be easily compatible and grouped together. This circuit includes a voltage input, a battery input, some resistors are used and a capacitor. Again the way this is supposed to help our project is to be able to regulate all power our device will be using, as well as be in charge of our two sensors, light and moisture, who need to be in the soil or outside our housing.

When looking at our schematic, you can see the moisture sensor jumper pin has three pinouts which will be connected to the jumper wire and the sensor itself will be put into the pot with the plant. Pin one is the analog output which is connected to a pinout on the microchip, IO2. Pin two is to be connected to ground. Pin three is the voltage pin which we have connected to our 3.3V regulator. The way that this is connected will allow our sensor to perform at peak performance.

When looking at our schematic, you can also see the light sensor jumper pin has four pinouts which is where our light sensor will be inserted. Pin one is the voltage

pin which we have connected to our 3.3V regulator. Pin two is to be connected to ground. Pin three is the analog output which is connected to a pinout on the microchip, IO4. Pin four is the digital output which is connected to a pinout on the microchip, SD0. The way that this is connected will allow our sensor to perform at peak performance.

When looking at our schematic, you can see the 3.3V regulator chip has three pin outputs. Pin one is the “IN” pin which is connected to a pinout on the microchip, IO0. Pin two is the “GND” pin which needs to be connected to ground. Pin three is the “OUT" pin which is connected to all sensors to regulate the voltage they receive as well as the enable pin on the microchip which enables the board to work, EN. In other words it will regulate the amount of power the board will receive.

Not pictured in our schematic will be a jumper connected to the 3.3V regulator out. It will be a jumper port for our microcontroller so that our PCB can handle all power and regulate the power for not only our microcontroller but our whole system. The reason this is not shown in the photo above is because we were limited on time to get this paper out before being able to finish the PCB schematic as a whole since we have only one team member working on it during this time. Right now the paper was our first priority so we had only one person doing the schematic for now. All decisions were made by all team mates and screen shots were sent to keep everyone in the loop on progress that was made during the process of importing libraries and of course what parts we wanted to use for each component.

The purpose of our PCB is to not only charge our battery for power, but to regulate power for the whole system. This includes the two sensors that will be jumped from this PCB, and the microcontroller that will have the rest of our sensors and display. These choices were made to reduce the cost of the printed PCB by keeping it relatively compact as well as having the power system be fully integrated.

5.2 Software

In order to deliver the data to our user, it is imperative that we have working and efficient software implementations of our hardware as well as a mobile app and database. For all of our sensors, there must be a software implementation with the microcontroller in order to ensure that the data is being correctly read and therefore in the proper format before updating our database.

5.2.1 Software flowcharts

This section will serve as the home for the flowcharts for development, as well as the overall structure of how the project will work together.

5.2.1.1 General Software flowchart

Below is the general software design flowchart for our device. This is a very simplified explanation of the big picture interactions between the larger systems of our device. What is not captured is the fact that the user will turn on the device either with the physical button or with the mobile app. In the case that the user turns on the device using our physical button, the process will still be the same in that the database will be updated, however they will not see the relevant notifications due to them not having the mobile app open.

It is possible to implement it in a way that once the sensor data is read, it will compare to the baseline on the microcontroller rather than the mobile app and alert the user on the LCD display as well. This however seems to be redundant as considering the user will be at home or on a trip when checking their plant, they would always have their phone on them therefore the mobile app is sufficient. The users will also be able to turn off the device using the mobile app, which is not captured but is an important software interaction with the device.

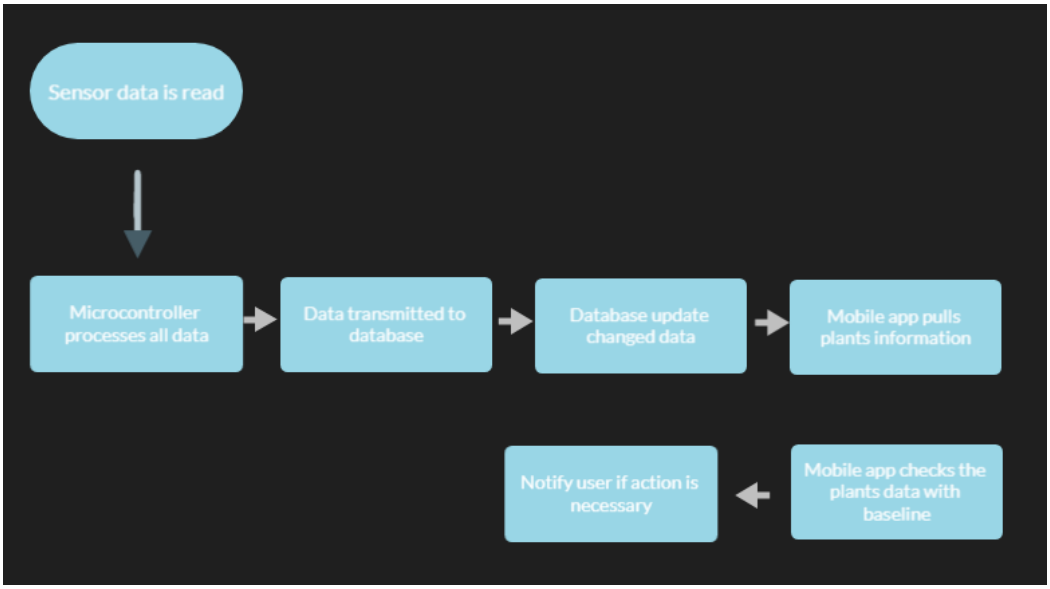

Figure 32 - General Software Flowchart

5.2.1.2 Mobile App Software flowchart

Below is the mobile app software flowchart which outlines the ways in which the mobile app interacts with our database to deliver the data our microcontroller gathers. In our general sensor flowchart, at every point in which the sensors are meant to notify the user, it is actually just sending that sensor data to the database and when it is displayed to the user within the mobile app, that is when

the user is notified of necessary action. One interaction that is not shown on this flowchart is that when the user starts the app, it sends the wake up call to the device, from there the MongoDB tables are updated.

A more detailed explanation of the mobile app itself and the components included within it are detailed in the software interface section. The most important interaction of the mobile app is waking up the device upon logging in, without this action there would be an issue of incorrect data being displayed to the user which would negatively impact plant growth/health (unless they physically start the device with the button, enabling an update to occur).

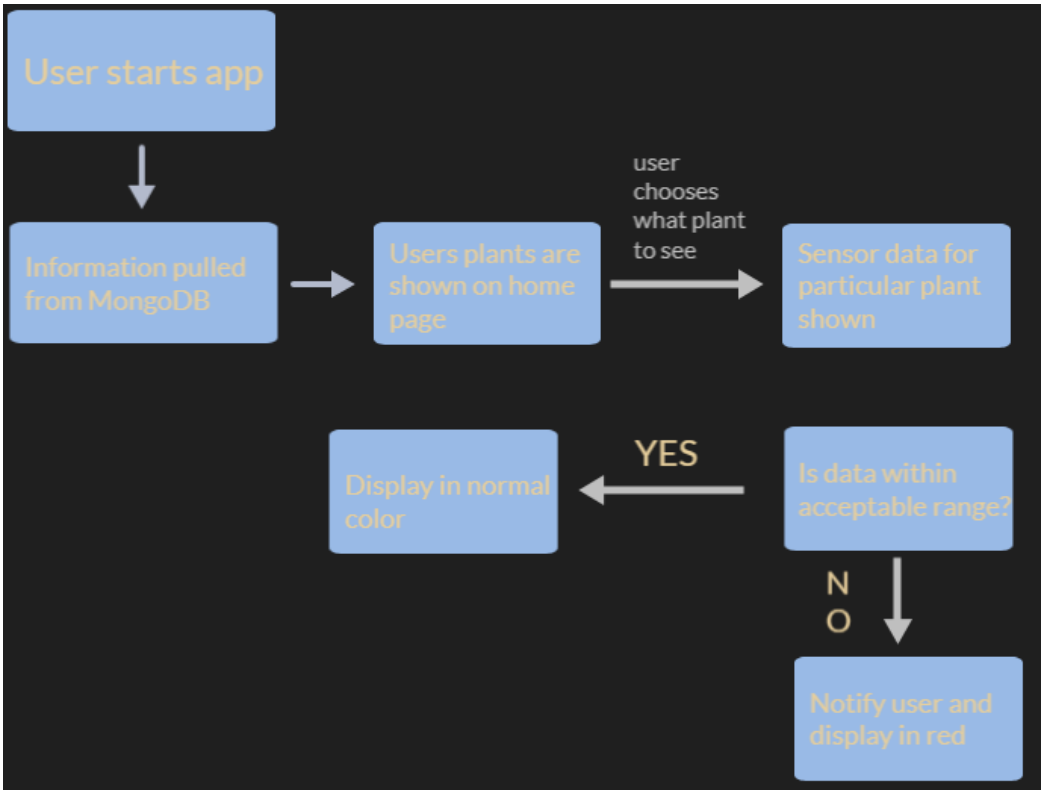

Figure 33 - Mobile App Software Flowchart

5.2.2 Software interface

In this section we detail the layout of the mobile app deliverable and go into the specific design choices we made after the testing of our hardware.

5.2.2.1 Mobile App Home Page

In figure 28 we show the mobile apps home GUI design. In order to implement this we plan on using React Native, an open-source mobile application framework which was designed by Facebook. This is an open source development environment that takes from React, an already popular and familiar library which works with JavaScript and allows us to take a similar approach to designing the mobile app. Seeing as this framework is being used by Facebook itself as well as Microsoft, and Uber it will be more than adequate for our application.

Due to the already simple framework of React, React Native operates in a similar capacity by allowing us to take a component based approach. This would mean that we are able to test the home screen itself, the user information itself, as well as the interaction between the mobile app and the microcontroller in order to wake up the device all separately. This would be meeting our coding standard as we are aiming to create code which is easily testable and verifiable.

Figure 34 - Mobile App Home

Previously in our software technologies, it was decided that we would create the app for android devices as it would be more difficult and costly to develop in the iOS environment. Upon further research it was discovered that for one year you can get a free development license for iOS app store meaning we would not need to purchase a license, as well as the fact that React Native will allow us to do cross development, making for seamless deliverables in both Android and iOS.

5.2.2.2 Mobile App Plant Page

In figure 29 the page of the mobile app which corresponds to the particular plant being tracked is shown. In this layout we show the name of the plant being tracked and the most recently updated sensor data that was read from the device. As shown in the general software flowchart, it is ensured that this should be the most recent data as upon wakeup of the device the sensor data will be recorded and then the database updated prior to the mobile app pulling the sensor data from the database. On this page is where you would get an indicator that one of the sensors is reading values that are outside of the plant's optimal range. This would be indicated by the text being in red and a small text box instructing the user on what should be done in order to bring the plant into optimal growing ranges.

Figure 35 - Mobile App Plant page

It is intended that this page would be in a separate view/route from the home page and therefore it would be easily verifiable/testable. Within the API for this page, we would have a query for our MongoDB database where we would pull all the relative information from the plant, including its recorded name, temperature, moisture, light intensity, and humidity. Under this page is also where you would view the battery life for the device, as it is intended to be scalable for multiple devices under one account and therefore the battery life should only be associated with the plant being monitored as the amount of plants a single device can handle is one.

5.2.3 Database Design

We will be utilizing MongoDB for our database in order to store data. We will be utilizing a database to perform two tasks: store information and data about different plants and their preferred environmental factors as well as storing live data when needed from the device's sensors. Therefore, one database will be mostly static unless we decide to add more plants and data into the database, while another is dynamic and will read sensor data. The user will be able to access the static data in one section of the app and the dynamic database also through the app but in their personal section where they can view their plants.

The static database itself will have a fairly simple design in terms of how the data is going to be stored. We will have a database with a table that stores plant names and under each: temperature, humidity, light level, moisture and pH levels. These values will be different for each plant based on research done. The user will be able to select from a list of plants and view the data held in the database for that plant.

The dynamic database for reading sensor input will be a bit harder to implement in terms of integration, but the actual structure will be fairly simple. We will have a table that contains cells for each of the data points that the device will be measuring, similar to the cells for each plant in the static database. We will have multiple readings over time, with the most recent one being shown to the user. Not only will the database be dynamically updated, but the app itself will also be updated to read from the database on set intervals. MongoDB is fast and will allow for quick readings from the sensor and a quick output to the user so that we get the most real time experience we can.

While we have the database, we will also need to host it somehow and allow for a program to connect to that database. This will be done using PHP in XAMPP, which will host our server that stores that database. PHP has a MongoDB driver that can be used to take variable information and query the database to store that data where we want it to be stored. This query will be done within our program in the Arduino IDE that connects to the sensors on the device and transmits data.

6.0 Integration

While we have covered the hardware and software design portions of the project as a whole, integrating the two is the most important part as it will allow the device to come together. The way to do this will be utilizing the Arduino IDE and installing the ESP32 board so that it can be used to program reading from the HUZZAH32 that we chose.

6.1 Integration via Arduino IDE

Due to the fact that the board we chose is not specifically an Arduino board, it is not entirely apparent that the Arduino IDE would be utilized for this project. However, the ESP32 boards can be integrated and used in the Arduino IDE, they just need to be installed first as a package. This is fairly simple and allows us a useful IDE with many features that we can use to gather data and utilize it.

Essentially, we will be writing code within the IDE that not only connects to the board and gathers data from the sensors when needed, but also connects to the webpage hosting the database and sends that information to the database.

6.2 Integration via Mobile Application

On top of this, we also need to allow the data to be read from the database. This will be done through the app, as the code for the app will connect to a web page/service that is hosted by us and set up via XAMPP that holds the sensor data in a database and gathers the information needed so it can display the data live to the user on their phone. Other than live readings, the user will also be allowed to view the plant database that we have created in order to see what the optimal environmental conditions are for the plant they choose.

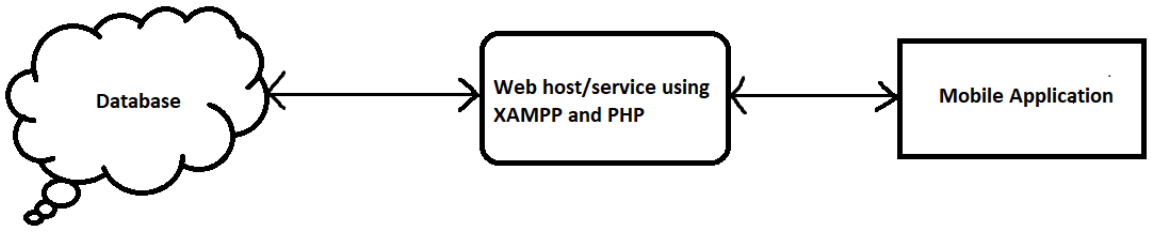

Figure 36 - Integration of data transfer

6.3 Effects of Integration

These two parts of the integration not only need to act individually, but they should be able to interact as well. Utilizing a web page will not only aid in helping with security, but it will also help us with this. When a user selects a plant within our static database, they will be able to see how their current readings compare to the plant choice that they selected.

Choosing a plant and having their selection compared to their plant readings will be fairly simple to implement, as we will simply have to allow for a comparison to be made between the two cells of the tables. This outcome will then be output to the user and the user will get a notification if the values are far enough off of the optimal for their chosen plant, so they can go make a change to correct the problem.

7.0 Administrative

This section serves to contain all managerial portions of the project such as budget, meeting dates, as well as project milestones.

7.1 Estimated Budget

Below you will find the Estimated Project Budgeting Chart for this project. The reason for “TBD” in the Quantity column is due to the possibility of multiple systems versus one. We are not sure if we will be creating one major system or multiple smaller systems for each plant. This is dependent upon our time frame given. We will achieve a working-in full prototype first and then branch out to making more systems either better or all the same. As we decide whether to add more sub systems, this section will be added to by changing “TBD” to numbers such as 2 or 3.

This project will be self-funded.

Item	Quantity	Price Estimate
Moisture Sensor	TBD	~ $10
Humidity Sensor	TBD	~ $15
Light Sensor	TBD	~ $10
Temperature Sensor	TBD	~ $5
pH Sensor	TBD	~ $50
Batteries	TBD	~ $15
Custom PCB	TBD	>= $30
Miscellaneous Plants	TBD	N/A
Possible Enclosure (plant &/or device)	TBD	TBD

Item	Quantity	Price Estimate
Possible LCD Display	TBD	~ $15
Breadboard	TBD	~ $10
Extra Components (LED lights, etc.)	TBD	<= $50
Total Estimated Cost		**~ $205**

Table 15 - Estimated Project Budget

7.1.1 PCB Bill of Materials

Below is the bill of materials for our PCB if we buy everything shown on our schematic. The bill of materials below shows the price of our board assuming we buy one. That includes one 10uF capacitor at the price of $0.21, one 3.3V regulator chip for $0.76, one 4 pin jumper for $1.10, one 3 pin jumper for $1.20, and two 5.1kOhms resistors for $0.03 each.

The MCP chip for charging is not available through Eagle so unfortunately we will need to find another way to purchase that. The ESP32-WROOM micro chip is also not available through Eagle so unfortunately that will also have to be found some other way. If we are unable to purchase these chips from another source then we will have to remake our PCB design with another microchip and charging chip module.

Quantity	Value	Package	Manufacturer	Price (from)
1	10u	C025-024X044	TAIYO YUDEN	0.21
1	3.3V	78XXS	ONSEMI	0.763
1		JP4	WEIDMULLER	1.1
1	MCP73831 T-2ACI/OT	SOT95P280X145-5N	**MOUSER**	0.77

Quantity	Value	Package	Manufacturer	Price (from)
1		JP2	WAGO	1.2
2	5.1k	0204/7	PANASONIC	0.025
1	ESP32-WROOM-32D	MODULE_ESP32-WROOM-32D	**AMAZON**	29.98
TOTAL				34.073

Table 16 - PCB Bill of Materials

It is important to take into account that the price of the PCB being manufactured has not been considered or added to this list yet because we are not ready to print our circuit board so we are not able to see what the pricing is. The pricing of a PCB is based on size as well as how many you buy at the same time. We ended up choosing a manufacturer that had the cheapest shipping time and price, as well as faster shipping and faster time that it takes them to create the board and actually prepare it for us which means putting the components on that we purchase through them.

As you can see from above, after searching for the two chips we were missing from Eagle we found one on Amazon and one on Mouser. The price of the ESP32-WROOM is $29.98 which we will be ordering from amazon.com and the MCP is $0.77 which we will be ordering from mouser.com. We needed to search far and wide for these components as one website we found was not shipping the chip for months and of course we do not have months to wait for these main parts since we are short on time. Luckily amazon and mouser are in stock and able to send us our parts as soon as possible, and because of this we will be ordering these parts right away even though our PCB is not ready to be ordered yet. Right now we plan on ordering our PCB overseas, however if we find out they will be taking time to get it to us then we will consider other options, such as the US manufacturer.

Fortunately for us we are now able to move forward with creating our board layout. The only problem we will face from finding these parts externally rather than from Eagle is that now rather than ordering the PCB already finished by the manufacturer, we will need to solder our parts on by hand when we receive the board.

7.2 Actual Project Cost

Within this section, you will find the cost breakdown of the parts that were purchased for the project. These tables were developed as the products were compiled, using Google Sheets, so the entire team had full transparency when it came to the cost of the project, to the individual parts.Table 8 showcases the subtotal cost of the project, including the name of the item, the quantity of the item purchased, and a subtotal of the products, including a cost per-unit.

Item	Package Cost	Quantity in package	Price
Battery	$24.50	1	$24.50
Microcontroller	$24.95	1	$24.95
Button	$2.50	10	$0.25
USB micro-b	$4.75	5	$0.95
LED	$6.99	100	$0.07
LCD	$8.99	1	$8.99
Light	$8.99	10	$0.90
Humidity	$5.95	1	$5.95
Moisture	$2.99	1	$2.99
Temp	$1.86	3	$0.62
Total (Pre-tax)	**$98.46**	**Total (Per-unit)**	**$76.16**

Table 17 - Project Subtotal Cost

Table 9, shown below, has a per-vendor breakdown of all products ordered including the cost of the entire product such as shipping, taxes, possible applicable tariffs. A grand total was achieved by adding all of the totals together, and the total per person value was calculated by taking the grand total value divided by the total number of team members, 4.

	Adafruit	Amazon	Mouser	Robotshop	Digikey	Xump
Subtotal	$53.70	$24.97	$5.95	$2.99	$1.86	$5.99
Promotion	-$5.37	$0.00	$0.00	$0.00	$0.00	$0.00
Shipping	$12.97	$0.00	$7.99	$2.99	$4.99	$5.99
Taxes	$3.14	$1.61	$0.39	$0.19	$0.13	$0.00
Tariff	$0.00	$0.00	$0.00	$0.00	$0.19	$0.00
Total	$64.44	$26.58	$14.33	$6.17	$7.17	$11.98
Grand Total	**$130.67**					
Total/ Person	**$32.67**					

Table 18 - Project Actual Cost

7.3 Project Milestones

Throughout research conduction, we will be able to note our prototype goals and then be able to add prototyping milestones to the “Build Prototype” portion. This timeline was developed initially, taken strictly from the timeline provided to us from professors. This is not a fully inclusive, or a fully accurate timeline. This only serves to give us a rough time table for when we need to expect things to be done and completed. We plan on working together every step of the way and helping each other reach our individual goals for the best outcome and output of our final project.

Milestone	Duration	Approximate Completion Date
Divide and Conquer		February 4
First Meeting With Dr. Wei		February 7
Initial Research	~ 8 weeks	February 7 - March 25

Milestone	Duration	Approximate Completion Date
First Rough Draft Document		March 25
First Rough Draft Meeting		Week of March 28
Second Rough Draft Document		April 8
Second Rough Draft Meeting		TBD
Final Draft Document		April 26
Build Prototype	~ 1 month	May - June (first week)
Testing	~ 1 week	June (second week)
Possible Redesign	~ 1 week	June (third week)
Final Prototype	~ 1 week	July (first week)
Presentation	TBD	July (first week) - TBD
Final Report	TBD	July (first week) - TBD

Table 19 - Project Milestones

7.4 Meeting Technology

In order to keep in contact with each group member and keep an organization among the varying aspects of our project, we intend to employ a multitude of apps and technologies available. By using these, we plan to eliminate clutter, improve communication among group members, and keep track of our progress as well as version control.

7.4.1 Github

Github is probably the largest hosting service for software development and version control and uses git. Git itself is a software that you can download which will monitor a particular set of files, often all referring to the same larger project, and allow you to track what files have changes made and even the amount of lines. It is a free, open source software, and was originally made in 2005 for the intended use on linux kernel.

There are similar alternatives to git, including bitkeeper and monotone which offer the same type of version control but as of 2022, bitkeeper is no longer being updated which makes it inferior to git and monotone does not have the same ease of use. Github is merely the website from which the version control and issue tracking can be facilitated throughout our development process and Git is how we upload the files in the project to be managed.

Using this combination of software is clearly advantageous for a multitude of reasons. When attempting to upload a new version of the code, you can view what exactly is changing before doing so and even leave comments for when fellow developers review the changes on github. Once the files are uploaded to github, it even displays what files have changed in a particular merge and in a section there is even a tracker for how many lines of code were added or deleted in a particular merge. Possibly the largest reason we are using github is that should the break upon a new merge or act in a way that was not anticipated, the previous version of the project which was prior to the error can be downloaded to revert it.

7.4.2 Google Drive

In order to create all of our documents and keep track of certain aspects of our project we have chosen to utilize google drive. After using Microsoft’s suite in the past, it was clear that Google Drive would be the better option as we have a substantially large main design document which requires constant change and multiple people are allowed to edit the document at once. This will allow us to write at a much faster rate and completely forego the need to combine different documents into a single one, it also makes the figure and table tracking much easier as it can be updated as they are added and in order.

7.4.3 Google Sheets

Aside from google docs, which was used to develop our divide and conquer documents as well as the larger report, we are utilizing google sheets. Google sheets is a simple to use spreadsheet, nearly identical to Microsoft Excel, which makes it intuitive to keep track of parts, prices, budget, and much more. Once the tables are created in google sheets they can just be copied and pasted into our design document making it a seamless process.

Aside from the design documents and parts spreadsheets, we have also used Google Docs to have a meeting document which has notes related to each meeting with the professor and has goals/guidelines for us to meet before the next meeting or submission.

7.4.4 Discord

To facilitate general communication between group members, have meetings online, and share relevant documents/links we have decided to utilize discord. There is an individual server for the project and within it we have separate channels which correspond to any large announcements, links to parts, and links to any relevant documentation including microcontroller datasheets, libraries relating to sensors we are using and a general message channel for communication between members.

Discord also allows for files to be sent in channels, meaning members can send pictures of particular parts being tested or show an error that is being experienced by a member. One of the best features of discord, is the screen share ability, with screen share we can have our schematic creation done online and forego the needs to meet physically, which gives us more flexibility and get more work done overall. When files that are detected to be code are sent on the application, it includes the formatting in the actual file which is particularly useful for reviewing another member's code without actually having to download it. Although usually used for social communication between friends, Discord is a free, very simple to use option for our project.

7.4.5 Trello

Trello is a much more formal application when compared to Discord in that it allows us to make a detailed list of tasks for our project. Once the tasks are created on trello, we can allow people to join a particular task to denote that they are working on it, as well as issue specific time constraints on said tasks. Although initially used by software programmers in the agile development cycle, Trello is incredibly useful for us as we can create concise and straightforward tasks that are meant to be easily identifiable and testable.

Upon initial glance it can seem fairly simple, merely a list of tasks with dates similar to a calendar, but it offers much more. There is the ability to have multiple lists (referred to as boards in Trello) which can divide the software/hardware needs, and each task within these lists can have tags which relate to the particular system it is a part of or even a description which gives a general understanding of what is to be completed. Trello will help greatly in the development/testing process by

allowing the members to keep track of what has been done and what is needed in the future.

7.5 Meeting Dates

When the project timeline, from Table 10, was compiled, the team decided to plan on meeting at least every Monday to talk specifics on what needs to be done next and by what person, as well as talk about the overall project. It did allow the team a time to schedule to meet regularly to discuss progress and issues that may have arisen since the last meeting. This, in execution, was not perfect, however. Life events occurred, team members could not attend a particular meeting and had to be filled in via the meeting notes, as well as collaboration with other team members. Certain Monday's, as a group, it was decided against having a meeting on those days, for one reason or another.

As a whole, however, the project team has stuck to a fairly regular meeting schedule, as seen in Table 11 below. As a note, this does not include any times that collaboration was made in between set meeting dates, only marking the agreed upon group meeting dates and times.

Meeting	Date
Initial Team Meet & Greet	1/13
First Project Meeting	1/24
Meeting Prior to Wei Meeting	1/31
First Meeting With Dr. Wei	2/7
D&C Meeting	2/14
Meeting Before Spring Break	2/28
Meeting Post Spring Break	3/14
60 page Check-in	3/23

Meeting	Date
Second Meeting with Dr. Wei	3/28
Weekly Meeting	4/10
Software Design	4/16
PCB Design	4/20
Weekend Meeting	4/24
Final Paper Check-in	4/25
Development Meeting	5/2

Table 20 - Meeting Dates

8.0 Conclusion

Following the completion of this document, a team meeting will be held to discuss this document in its entirety. The team will discuss the overall feeling of how the document went, when to move forward with development, and note any changes that are felt should be made. A team review will be held on an individual level and told, as a group, where we think individual team members were lacking, and where they excelled. This review will serve as a check in for everyone to see how we are all feeling about the overall performance of the members. If there are any notable issues that occurred, they will be addressed in this meeting.

Moving forward post-review meeting, the team will hold a meeting to discuss how to initially start development. The earliest start of development will occur when the team does self-learning as to how to create an app from scratch, as it has never been done by any of the team members before. This self-learning will consist of talking with friends, coworkers, recent graduates, watching instructional Youtube videos, as well as browsing the Internet for tutorials. This will be one of the biggest hurdles moving forward, as it is a key component of our overall project design, but no one on the team has any experience with creation of it.

Another large hurdle moving forward will also be the 3D printing design creation. As stated previously, the team aims to create a 3D printed shell surrounding the entirety of the system, creating an enclosed, waterproof system to house all electronic components. This aim, however, will be futile without spending time learning and experimenting with various designs. One of the team members is certified in Solidworks 3D modeling software, so there is experience on the team. This experience is a few years old, and will most definitely need to be refreshed. After refreshing the knowledge of how to work and model in 3D, there will be extensive research and development that will occur with the model design. The model design has been briefly mentioned in prior meetings, allowing for an overall very basic consensus about how we desire the design to look. This desire may or may not be able to come into fruition. It will hinge on how the design of the internal structure turns out. 3D modeling will be performed, but nothing will be printed and tested until the full electrical structure is filled out.

Part procurement is the next issue to deal with. As previously mentioned, most of the parts have been purchased, and extensively tested already. During testing, a few issues were noted, and there are even parts that came in broken from the factory. These issues such as the inability to get particular sensors to communicate with the microcontroller will have to be dealt with, and resolved, before any feasible work is completed. These sensors are vital to the success of the project, and without them, the project will not be as well put together as planned. More time and research needs to go into these sensors, allowing the team to decide if there is an issue regarding the sensor itself, or if it is possibly as simple as a wiring, or a coding issue during testing. During testing, it was also noted that there was a

completely faulty piece. This part will have to be replaced, and a potential replacement will have to be considered. It has to be considered as to whether a different LCD will be purchased, or if a new version will be purchased from the same vendor. Purchasing from the same vendor has the potential to encounter the same issue, as it may be a faulty batch of screens that were manufactured. It very well could have just been a one-off issue, however. All of these things will have to be considered and discussed as a team before moving forward.

PCB design and procurement is another key piece to the success of this project. As stated in the requirements, a custom made PCB must be incorporated, and it must handle 2 significant features. As it currently stands, we have a tentative design, but are awaiting full critique on the design. This design will have to be vetted, approved, then once there is full approval, it will have to be ordered. A vendor will have to be decided, then ordered. For most vendors, on initial research, it seems as though there is about a 2 week period from when we order, to when the team receives the PCB. This time will be utilized to either finalize the software side, or make further improvements to the structure of the system. It should be noted that there is always a possibility of the PCB not working on the first try. Although we have the capability of having two separate professionals look over our design and schematic of the PCB, there is a potential for the design to either not fully incorporate all of the features properly, or the actual print of the board is faulty. This will not be determined until after waiting the necessary lead time for the board. If there is a significant error around the design of the PCB, a new board will have to be ordered asap, so as to avoid the issue of not having a PCB during the final conclusion of the project.

As the project moves forward, it is necessary to note the time constraint put upon us, as the conclusion of this project will be over the summer, which is about a month shorter than a normal fall or spring semester. This timing will be attempted to be overcome by starting early with both design, prototyping, and necessary part procurement, as early as possible. Tentatively, the team will hold meetings over the small break in between the two semesters to initially start the app and 3D modeling design portions. Getting this head start of the project, in the team's opinion, will be vital for success moving forward. The earlier we start, the more potential time we have to recover from any significant issues, or unforeseen circumstances moving forward. As it is the summer, everything will be rushed, and the team will make every effort to prevent any issues when it comes to timing or completion.

All in all, this project, thus far, has been a success. There have been very minor hiccups along the way, but nothing significant enough to warrant any worry or concern. Following this wave of success, the team hopes to move forward into the second half of the project with vigor. The team has plans to host meetings, and get started on the prototyping portion of the project sooner rather than later. With this in mind, the team will continue to host regular meetings, transitioning from fully virtual meetings to more in person, so that the team can collaborate and build

physically together, rather than one person doing the development and prototyping. All part procurement is hoped to be completed within the weeks following the conclusion of the first half of this project, allowing for adequate time for shipping, as well as prototyping when the parts arrive.