July 11, 2022

Dr. Craig Scratchley
School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6

Re: ENSC405 Design Specification for *DIRTS*

Dear Dr. Scratchley,

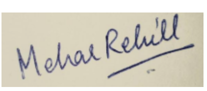
This document contains the design specifications for Direct Interface for Rapid Testing Soil, or DIRTS, prepared by Everyday Planting Solutions.

Our objective at EPS is to research and design a product beneficial to both amateur gardeners and industrial agriculture. DIRTS will provide an easy-to-read status on the current state of the soil where it is placed, along with a simple phone application to give recommendations of plants that would survive well in the tested environment.

The attached document includes the design requirements for the hardware, software, and electrical components of DIRTS, along with detailed explanations of the expected development phases throughout the project. Alternative design plans and comparisons will be included to ensure the device works properly in the most efficient manner. Details regarding the appearance and functionality of DIRTS at the end of term in August will be provided in this document.

Our team at Everyday Planting Solutions would like to thank you in advance for taking the time to review the design of our device DIRTS. If you have any more questions or would like further information regarding our project or team, please contact mrehill@sfu.ca.

Sincerely,



Mehar Rehill
CEO
Everyday Planting Solutions

School of Engineering Science
ENSC 405W

Design Specification:
DIRTS
(Direct Interface for Rapid Testing Soil)

Company 2:

Mehar Rehill, CEO

Gurparkash Singh, COO

Francis Chui, CAO

Shravan Gupta, CTO

Kyle Granville, CSO

Contact:
Mehar Rehill
mrehill@sfu.ca

Submitted to:
Craig Scratchley, ENSC 405W
School of Engineering Science
Simon Fraser University

Issued Date:
July 11, 2022

Abstract

In this document, the design aspects of DIRTS to assist both aspiring household gardeners and the agricultural industry are explained. The proof-of-concept and prototype phases will be included as stages of the overall project. The design specifications will cover the product's electronic, structural, and software aspects of design, along with our team's reasoning for the choices. To measure the soil, DIRTS will use its various sensors to gather information at set intervals of time, and deliver the data to the user's mobile phone through the application. The software application will show the user necessary components of the soil regarding plant growth, such as nutrients, pH, moisture, and temperature data from the sensors. A database within the application will be created to give better estimates for which types of plants thrive in the surveyed environment's soil. One of the main issues of the device is designing an effective optical NPK sensor, as the currently available market NPK sensors are all relatively expensive.

Table of Contents

List of Figures

List of Figures (Continued)

List of Tables

List of Equations

Glossary

Term	**Definition**
CSA	Canadian Standards Association
IDE	Integrated Development Environment (coding environment/application software)
MCU	Micro-Controller Unit (Arduino)
NFC	Near-Field Communication
NPK	Nitrogen Phosphorus Potassium
OS	Operating System
pH	Potential of Hydrogen, Acidity
RFID	Radio–Frequency-Identification
USB	Universal Serial Bus (typical data transfer standard used)
UI	User Interface

1 Introduction

Healthy soil is one of the essential factors in the growth of a plant, providing nutrients, anchorage for the roots, water, and regulating temperature. Over time and throughout the seasons, the soil in a specific location may alter, requiring outside help to achieve optimal conditions once again. For example, soil may become too acidic over time, such as from acid raid or water leaching away basic ions in the soil [1]. To monitor these conditions, Everyday Planting Solutions is creating a device to gather these soil properties and relay the information directly to the user's mobile device.

Figure 1.1: Showing Different Expressions Regarding the Plants

1.1 Scope

This document outlines and describes the design choices made by EPS while creating DIRTS, which includes expected dimensions, materials, software, and electronic component design choices. In addition, the document includes alternative design options, test plan appendix, and user interface design for our product.

The User Interface Design talks about how the user will engage with our product, enlisting thorough analysis on the expected use of the device. The test appendix summarizes the necessary steps to ensure that DIRTS meets our company's standards and fulfill the needs of our potential users by maintaining the government regulations.

1.2 Intended Audience

This document is meant to serve as DIRTS' design specifications for potential users, Everyday Planting Solutions (EPS) members, investors, partners, Dr. Craig Scratchley, and teaching assistants of the courses ENSC 405w and ENSC 440.

1.3 Challenges

Some of the challenges pertaining to the development of this systems are as follows:

- Natural decay and oxidation of components from extended periods in soil or outdoors
- Gathering and calibrating correct measurements from sensors
- Sending proper sensor data to mobile application
- Creating durable and simple sensor probes for device
- Designing an optical NPK sensor from basic materials

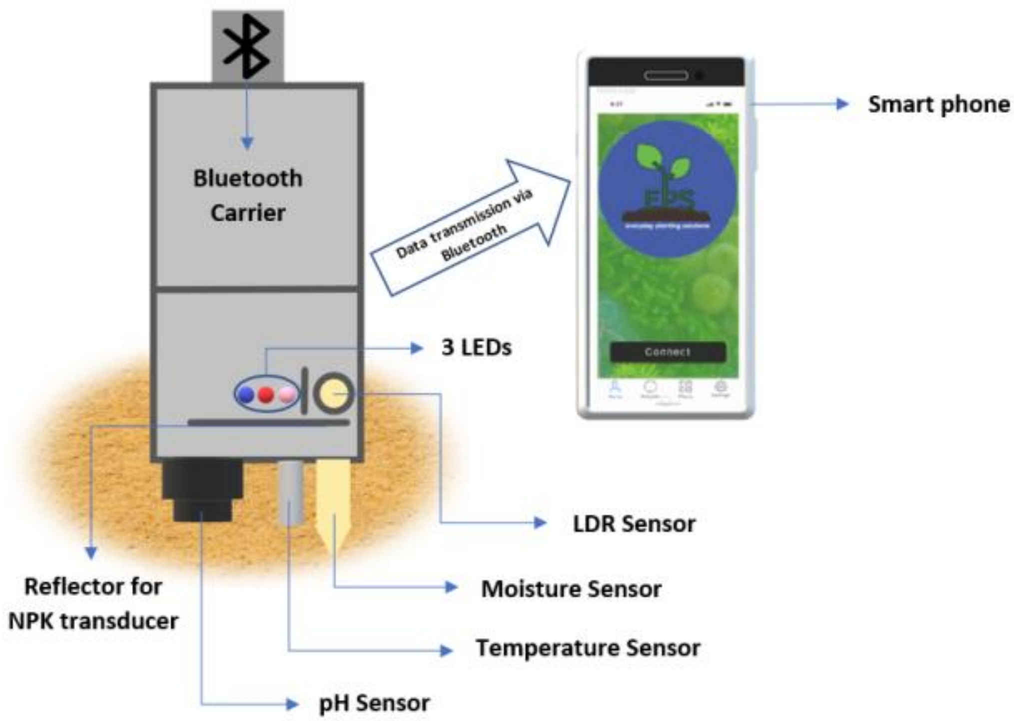


Figure 1.2: Hardware connection to software

1.4 Design Specification Classification

The design requirements in this document will be using the conventions and product phase notations as shown in Table 1.1.

DES <Document Section>.<Requirement Number>-<Product Phase>

Notation	Development Stage	Description
PC	Proof-of-concept	Focus will be mainly on the working of the device and the related software. Designs used here will be limited just to provide the first look.
B	Prototype	The desired designs will be implemented while encapsulating the whole circuitry into a case. The product will be ready for user testing
GG	Final Product	This will include final revisions of the product based on testing feedback. Additional specifications will be added for the ease of accessibility.

Table 1.1: Product Development Phases

2 System Overview

DIRTS is a sensor that can be inserted to the top level of soil, with several probes to measure pH, moisture, NPK concentrations, and temperature. The device is designed to be paired with a mobile phone application, which can recommend suitable plant types for the given soil conditions. For the prototyping phase, the system will be designed using a simple breadboard circuit with the microcontroller. During the product phase and beyond, a waterproof shell will be included to allow survivability in the changing conditions of nature.

With the increasing availability of technology, especially regarding phones, people are becoming more reliant on automation and simpler ways of dealing with everyday tasks. By creating a mobile application along with the sensor device, the need to design an integrated screen is removed. The application will need to portray enough information to be useful, but not too much information in order to keep it easy to understand the device usage.

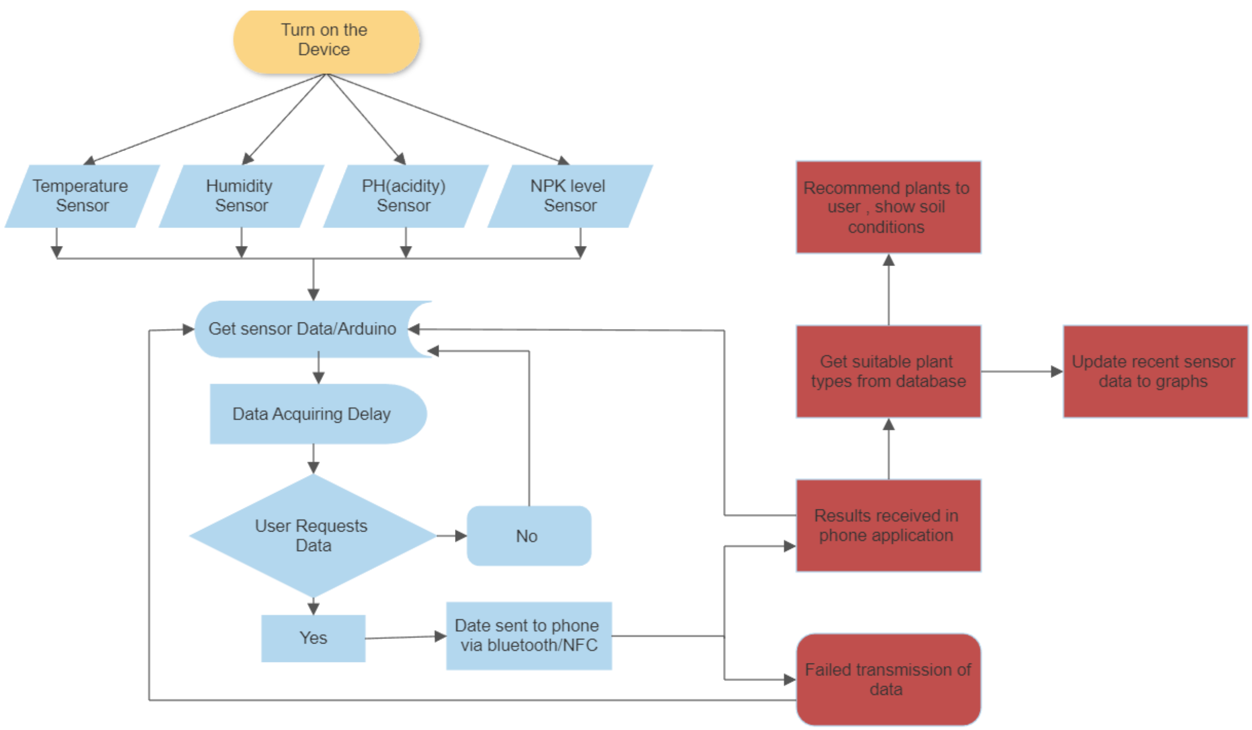


Figure 2.1: Device Overview Flow Chart

3 Hardware Design

There are several interconnected components to consider when designing this device. This includes the sensors, a power supply, a microcontroller, and a waterproof enclosure.

3.1 Electrical Design

For the proof-of-concept phase, the Arduino will be connected by USB to a computer to monitor signals and apply voltage to the system. The circuit will be created on a breadboard while we assess the signals and design a practical physical model for the device.

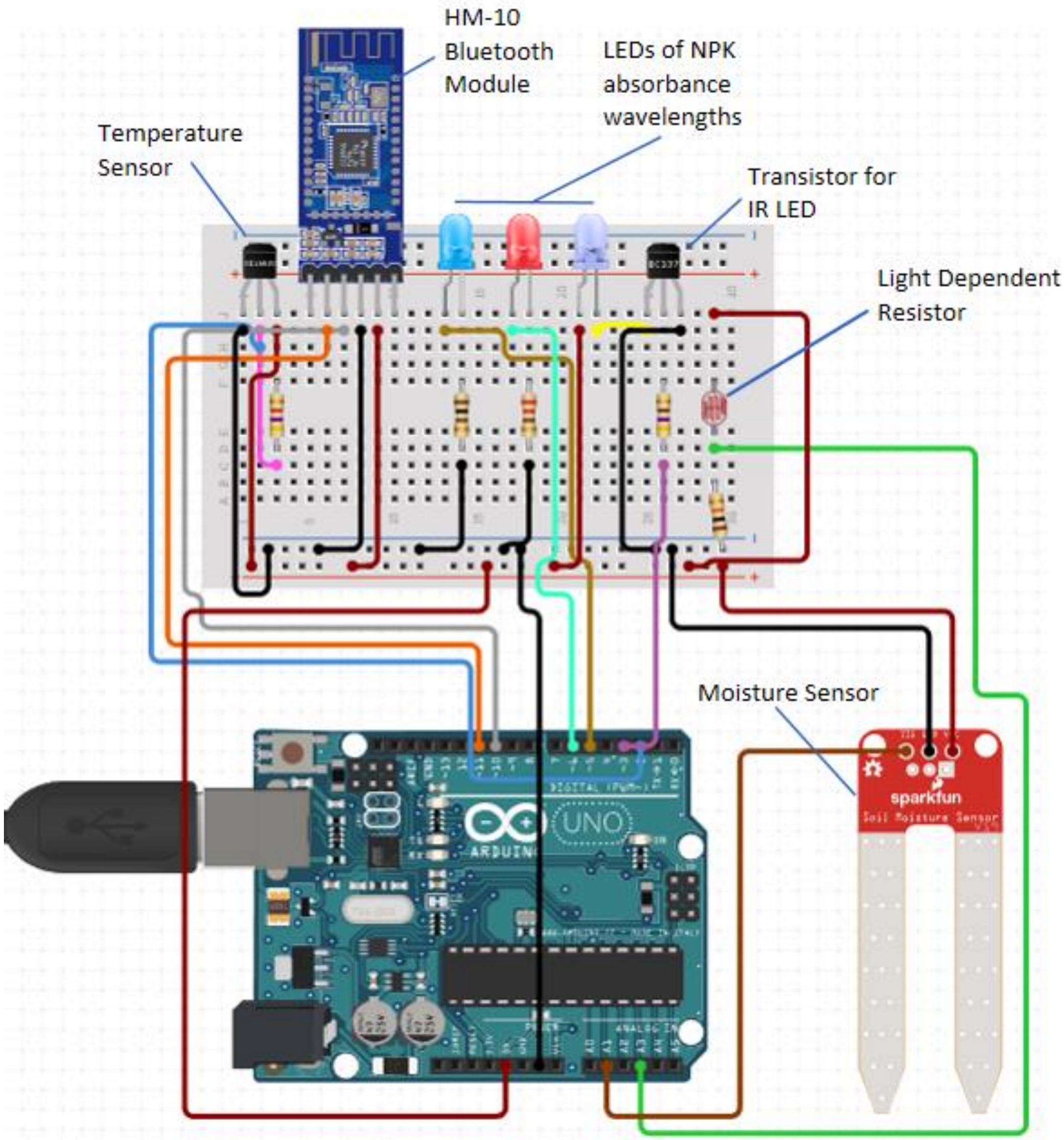


Figure 3.1: Hardware High Level Circuit

3.1.1 Sensors

3.1.1.1 Temperature

To collect the temperature of the soil, we will be using a digital thermometer with part number DS18B20. It can measure temperatures from-55°C to +125°C, and is accurate to within 0.5°C between the range of -10°C to +85°C, as shown in Figure 3.3. This specific model of the sensor was chosen due to the fact it already has a built-in probe to measure, making it easily compatible with our planned design. This sensor runs from a supply voltage between 3.0V and 5.5V, and has an active current of 1mA to 1.5mA. This will connect easily to the Arduino without requiring any advanced circuitry. [2]

Figure 3.2: DS18B20 Digital Thermometer

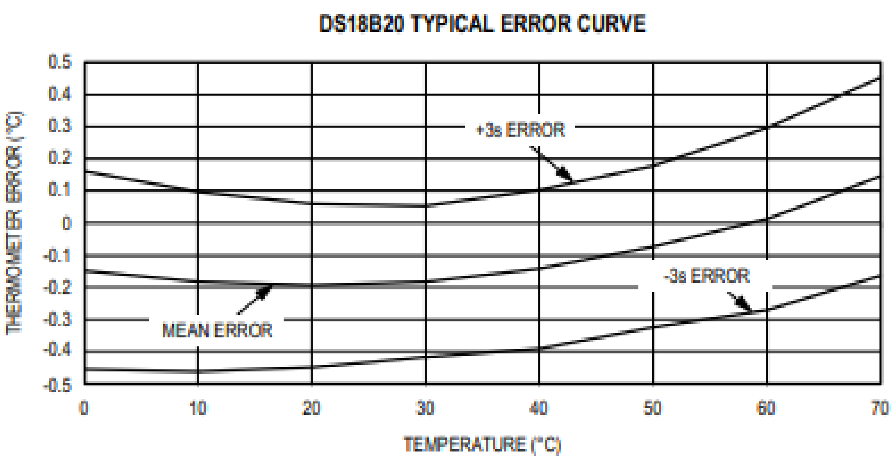


Figure 3.3: DS18B20 Typical Error Curve by Temperature

3.1.1.2 Moisture

To measure the moisture levels of the soil, we decided to use a dual-probed simple sensor, with part number SEN-13637. This sensor uses the two probes as variable resistors, where a higher moisture level in the soil results in higher conductivity, and therefore a higher signal output. It works best using an input voltage between 3.3V and 5V, which is what the Arduino will also be using. Depending on the exact voltage used, the analog output signal will change accordingly [3].

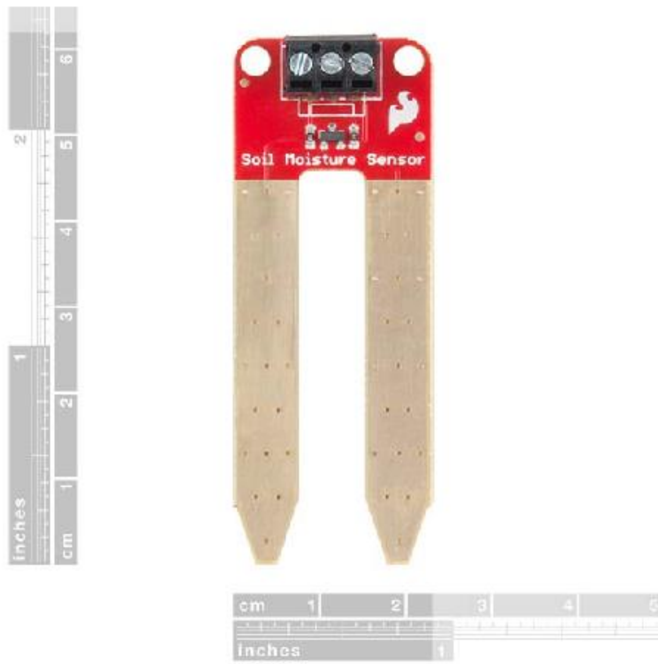


Figure 3.4: SEN-13637 Moisture Sensor

3.1.1.3 Nitrogen-Phosphorus-Potassium

Plants take up all their nutrients from soil through their roots and from air through their leaves. The nutrients are divided into two categories: macronutrients and micronutrients. These nutrients stay beneath the soil and plants absorb them in the form of ions [4]. The macronutrients are taken up in large quantities and among them nitrogen (N), phosphorus (P) and potassium (K) are the elements to be monitored. Nitrogen is used for good green color, Phosphorus is used by plants to make seeds and new roots, and Potassium helps to make stems strong and is used to fight disease [5].

These nutrients are taken up by plant roots as followed:

N is absorbed as ammonium or nitrate

$$NH_4^+ + 3O_2 \overset{Nitrosomonas}{\rightarrow} 2NO_2^- + 2H_2O + 4H^+$$

$$2NO_2^- + O_2 \overset{Nitrobacter}{\rightarrow} 2NO_3^-$$

Equation 3.1: Ammonium oxidation

P is absorbed either as H2PO4- or HPO4 2-, both anions

$$KH_2PO_4 + H_2O \rightleftharpoons K^+ + H_3O^+ + H_2PO_4^-$$
$$pKa_1 = 2.15$$

$$H_2PO_4^- + H_2O \rightleftharpoons H_3O^+ + HPO_4^{2-}$$
$$pKa_2 = 6.82$$

Equation 3.2: Potassium dihydrogen phosphate ionic dissociation

K is absorbed as K+, a cation [6]

$$K_2SO_4 + H_2O \rightleftharpoons 2K^+ + SO_4^{2-}$$

Equation 3.3: Potassium sulfate ionic dissociation

[Optical Sensing of Nitrogen, Phosphorus and Potassium: A Spectrophotometric Approach toward Smart Nutrient Deployment]

Detecting these elements in a lab environment relies on Ion-Electrodes, which are very useful in laboratories. However, their on-field application is limited because of a short life span and a need for constantly recalibrating. To overcome this issue, we decided to measure the NPK levels using spectroscopy.

The optical transducer for NPK is formed by the combination of a light transmission and detection system. The overview of NPK detection is illustrated in Figure 3.5.

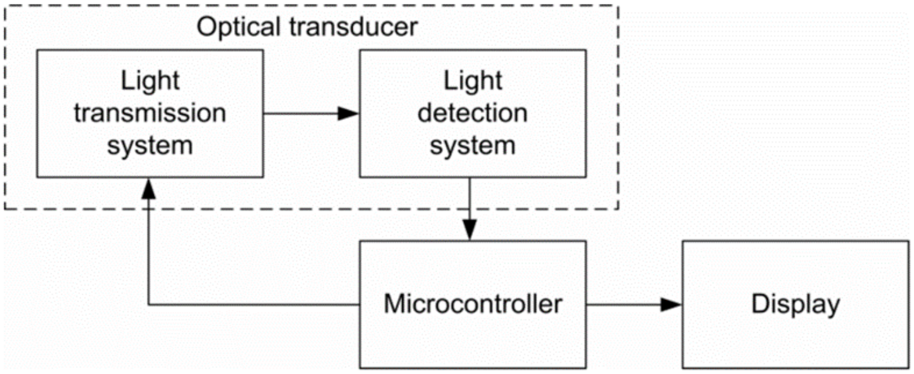


Figure 3.5: NPK Detecting System Overview

In a light transmission system, three LEDs with different wavelengths are used. Each LED was chosen according to the spectrum absorption wavelength by each nutrient. Table 3.1 lists the optical characteristics of nitrogen, phosphorus, and potassium and corresponding LED emittance [7].

Nutrient	Absorption wavelength (nm)	LED type	Wavelength (nm)
Nitrogen (N)	438-490	LED 1	460-485
Phosphorus (P)	528-579	LED 2	500-574
Potassium (K)	605-650	LED 3	635-660

Table 3.1: NPK Optical Characteristics

In the setup, the LED and the photodiode are positioned in a parallel manner, both facing the same direction shown in Figure 3.6.

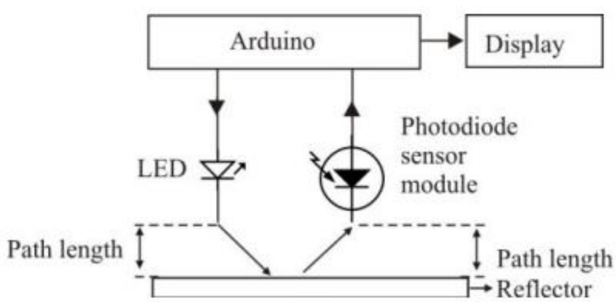


Figure 3.6: NPK Optical Sensor Lab Setup

The light is reflected by the reflector and then detected by the photodiode. The intensity of reflected light from the reflector to the detector is analyzed to determine the optical path length. After various comparisons of the responses from the photodiode for different OPL, the optimum path of 1cm is chosen due to the highest light intensity, as shown in Figure 3.7.

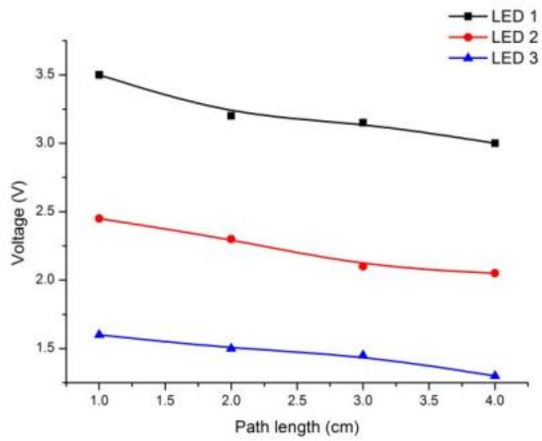


Figure 3.7: Voltage vs Path Distance Graph

From the above analysis, a threshold voltage was decided to differentiate the level of nutrients into three voltage levels shown in Table 3.2.

Nutrient	Low (V)	Medium (V)	High (V)
Nitrogen	3.5<x<3.8	3.8<x<4.1	x>4.2
Phosphorus	2.45<x<2.8	2.9<x<3.3	x>3.4
Potassium	1.6<x<2.2	2.3<x<2.8	x>2.9

Table 3.2: Voltage vs Path Distance Ranges

3.1.1.4 Acidity

When taking care of plants, we must know what kind of soil is required. The pH sensor will determine the acidity of the soil [8] and covers the full range of 0-14 pH with a response time of less than 1 minute.

The sensor to be used in DIRTS is a differential pH sensor, which is made of an industrial electrode. The pH electrode is made of a sensitive glass membrane with low impedance and provides output as linear voltage [9]. When not in use, it is advisable to cover the electrode as shown in Figure 3.8.

Figure 3.8: Digital pH Sensor

The sensor consists of a reference electrode containing a potassium chloride solution with a block of mercury chloride at the end. This electrode is used to provide a stable zero-voltage connection, which is then used to find the potential difference between this electrode and the measuring electrode. The pH level is then calculated using the Nernst equation.

3.1.1.5 Sensor Design Specifications

To create the device, design specifications for the sensors are included in Table 3.3.

Design Label	*Design Description*	*Corresponding Requirement*
DES 3.1.1.1-PC	The sensors shall be placed on a breadboard	REQ 3.1-A, REQ 5.1.1-A
DES 3.1.1.2- PC	The sensors must function independently.	REQ5.2.2-A
DES 3.1.1.3-PC	Leakage of Infrared Red should be provided by enclosing the LEDs	REQ5.1.4-B
DES 3.1.1.4-PC	The pH sensor should be covered when not in use.	REQ5.1.4-B
DES 3.1.1.5-PC	Must work with the regulated DC voltage supply	REQ4.1-A, REQ4.2-A, REQ5.2.3-A
DES 3.1.1.6-B	Device accuracy must remain high for a long time.	REQ5.2.4-A
DES 3.1.1.7-B	Individual covering must be provided for storing purposes	REQ3.4-B, REQ5.1.4-B
DES 3.1.1.8-GG	Wiring must be shielded properly to avoid the addition of noise.	REQ 5.1.3-A, REQ5.1.2-A, REQ5.2.6-B

Table 3.3: Sensor Design Requirements

3.1.2 Power Supply

For the proof-of-concept phase, we will be mainly testing the device while connected to a computer by USB, which supplies the power and can view real-time data from the sensors. The Arduino runs using the 5V power from the USB, which should be enough to supply current through each sensor as well as the microprocessor.

For the prototype phase, the device will ideally need to be more independent, so there will be no wires extending from the device. This requires a power supply to be within our device, which can be achieved by a typical 9V battery. The Arduino Uno model comes with a barrel connection port, which can easily be utilized with a proper adaptor.

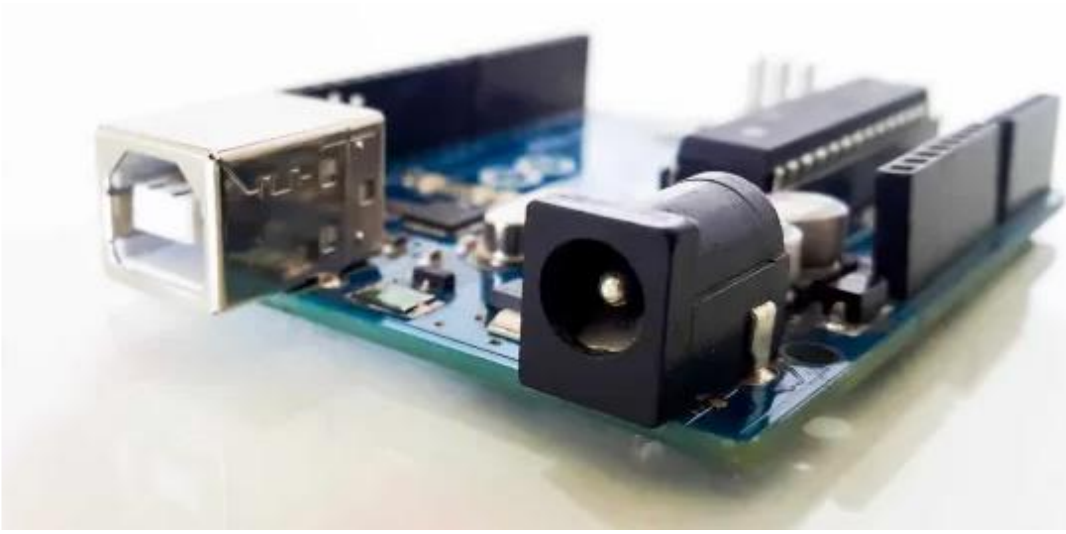


Figure 3.9: Arduino Uno Isometric Power Port View

3.1.3 Microcontroller

For the proof-of-concept phase, we will be using an Arduino Uno. This microcontroller is relatively compact, is easy to debug, and inexpensive. This microcontroller runs using a voltage between 3.3V and 5V, which is regulated within the unit when powered by USB.

Figure 3.10: Arduino Uno Top View

For prototype phases and beyond, the Arduino Uno may be replaced with a smaller version, such as an Arduino Nano or Arduino Micro.

Figure 3.11: Arduino Nano

3.1.4 Bluetooth/NFC

To allow the system of the physical device and the application, we require a way to send information from the sensors to the user's mobile phone. Our goal is to complete this task by using a bluetooth HM-10 BLE model. This model uses less energy than many other bluetooth components for Arduino [10].

Figure 3.12: HM-10 BLE Module

3.1.5 Alternative Designs

As an alternative sensor to the optical NPK sensor that we are designing, there also exist sensors that measure nutrient concentrations in other ways. One such example is the RS-NPK-TR [11], which measures the capacitance of different frequency signals between the three probes. Normally, soil nutrient levels are determined in a lab situation or by using a chemical testing kit. The different nutrient levels of nitrogen, phosphorus, and potassium are monitored by accessing different locations within the sensor's memory.

Figure 3.13: RS-NPK-TR Sensor

3.2 Structural Design

The structural design of DIRTS is planned with a great consideration by our team members. The device will be kept outside in the soil for extended periods of time experiencing weather changes. Therefore, our goal is to make the device robust so that not only can it withstand changing environmental conditions, but also protect the internal circuitry from damage. Figure shows the first revision of casing for the device encapsulating all the sensors and circuitry. For the final design, the case will need to be made waterproof to protect the circuit when the device is in use.

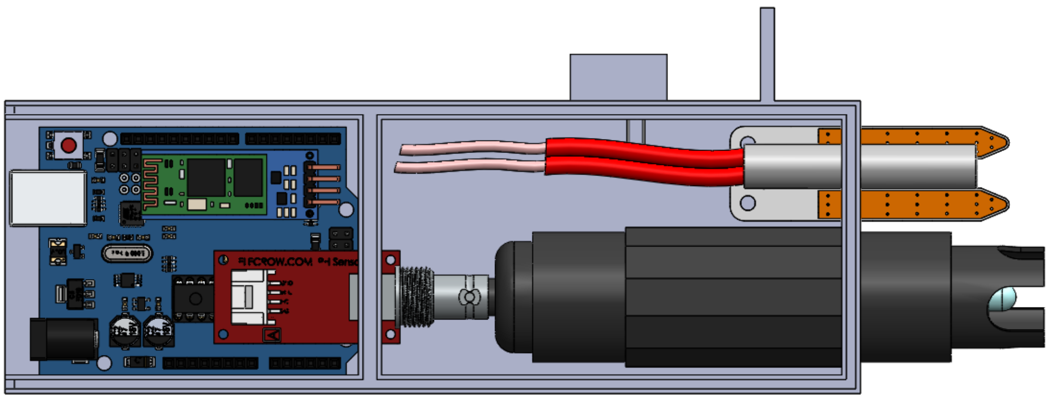


Figure 3.14: Side Schematic View of DIRTS

DIRTS is designed to be accessible among the public, and to make that happen, the device must be cost-effective. Keeping cost and environmental considerations in mind, the case will be 3D printed using PLA plastic for the proof-of-concept phase. PLA is a thermoplastic monomer derived from renewable and organic sources which makes it biodegradable. Various properties which make PLA a suitable option [12] according to our surroundings are listed in Table 3.4.

Property	Value
Heat Deflection Temperature (HDT)	126 °F (52 °C)
Density	1.24 g/cm^3
Tensile Strength	50 MPa
Flexural Strength	80 MPa
Impact Strength (Un-notched) IZOD (J/m)	96.1
Shrink Rate	0.37-0.41% (0.0037-0.0041 in/in)

Table 3.4: PLA properties

In terms of Canadian temperature, there have been few recordings of temperatures exceeding 50°C, which is slightly below the maximum temperature threshold for the plastic. This means that the product will stay rigid over time. However, because the plastic is biodegradable, routine inspection must be done after every 4 months.

To make it waterproof, o-rings made from rubber can be used in-between the enclosure. Another alternative is to use the rubber seal with metal clips for longer durability in adverse climate situations. These seals will retain a permanent grip and won't expand or reduce in extreme temperatures

All the above-mentioned structural analysis gives us the design requirements of Table 3.5.

Design Label	*Design Description*	*Corresponding Requirement*
DES 3.2.1- PC	Rain and moisture should not affect the internal circuitry	REQ 5.1.4-B
DES 3.2.2-GG	Dust should not get collected inside the enclosure	REQ 5.1.4-B
DES 3.2.3-GG	Extreme heat should not cause the enclosure lose rigidity	REQ 5.1.4-B

Table 3.5: Design Specifications for Structural Design

4 Software Design

Software design is a critical part of the product as the mobile application is the main form of communication between the user and the system.

4.1 Application

Initially, the app will be available solely on Android devices, but eventually we plan to extend it to Apple devices. Through the application, users can view real-time pH, moisture, NPK concentrations, and temperature values once the sensors are submerged in the soil. We are developing the app in C# and will be using Xamarin tools to develop cross-platform implementations for Android (Xamarin.Android) and iOS (Xamarin.iOS) with the help of the C# shared codebase. Our choice of IDE is Visual Studio 2022, along with Google Android emulator to visualize and design the UI of the application.

We choose to develop an android application so we can use bluetooth to transfer data from the sensor to the user in a more user-friendly format. This will eliminate the need for wires and allow the user to simply focus on putting it into the soil. Not only that, our target audience will likely have a device with Bluetooth capability, thus this is not a concern. We intend to use mobile devices as plants can get planted and relocated in various places, so having a mobile device will make it easier to be more flexible location-wise. iOS is also a target audience as well, and will be developed after the android product is done.

A primary reason on why we have chosen C# as the language is due to the experience of our team being more experienced in C++ and C. Due to this experience, We believe that it will be easier to learn than the alternatives. Another reason is that we intend to develop for both iOS and Android and using C# is a viable option for both platforms. It is also easy to use same C# code for iOS and Android with the help of Xamarin. Xamarin is an Open-source Android app platform that we also intend to use, which can be used for iOS as well.

Design Label	*Design Description*	*Corresponding Requirement*
DES 4.1.1 -PC	The software application would be available for Android devices	REQ 6.1-A, REQ 6.1-B
DES 4.1.2 - PC	The software application would be able to retrieve plant information from a database	REQ 6.1-A
DES 4.1.3 - B	The software application would display sensor results in real-time	REQ 6.1-B, REQ 6.1-GG
DES 4.1.4 - B	The database will be stored on a cloud and the software can update its information based on the cloud information	REQ 6.1-B
DES 4.1.5 - B	The software application would be available for iOS devices	REQ 6.1-B

Table 4.1: Design Specifications for Application

4.2 Front-End Design

The app will consist of four main pages: Home, Analysis, Plants, and Settings. When a user opens the app, they will be on the **Home** page where the Bluetooth connection would be established to the sensors once the users press the connect button. If the connection is not established, they will get an error message with a prompt to attempt connection again (Req 6.1).
Once connected, they will need to go to the **Analysis** page where the results from different sensors would be displayed, along with the computed plant suggestions from the database. They will also have the option to enter a plant they know and the computation lets them know if the plant would survive given soil condition. At the backend, we compare data from different sensors with their specific tolerances for different plants from our database and that will satisfy Req 6.2.

The **Plants** page is a general information page of all the plants from the database where users can find relevant information for the plants they want. We will make this page functional by having filters, drop down menus, hovers etc. to make the UI more accessible.

The last page is the **Settings** page where there are basic accessibility (font, font size, light/dark mode) settings, documentation for products, and a help button to contact the support team.

Design Label	*Design Description*	*Corresponding Requirement*
DES 4.2.1 - A	The application allows users to input a plant to search database	REQ 6.1-A
DES 4.2.2 - B	The application gives an error if the connection is not established	REQ 6.1-B
DES 4.2.3 - B	The application automatically takes users to the Analysis tab once the results are available	REQ 6.1-B
DES 4.2.4 - B	The application shows values of NPK, Humidity, pH, and Temperature	REQ 6.1-B
DES 4.2.5 - B	The application establishes a bluetooth connection once users press 'Connect' button on the home page	REQ 6.1-B
DES 4.2.6 - B	The application informs users regarding any abnormalities in	REQ 6.1-B

	the results	
DES 4.2.7 - B	The application has a friendly UI	REQ 6.1-B
DES 4.2.8 - B	The application has user accessibility settings accessing font, font size, and light and dark mode settings.	REQ 6.1-B
DES 4.2.9 - GG	The application compares data from different sensors with their specific tolerances	REQ 6.1-GG

Table 4.2: Design specification for Front-end

4.3 Back-End Design

4.3.1 System Communication

The main back-end component of our application is our connection to the Bluetooth sensor. Our application is expected to be able to connect to the Bluetooth sensors outside the application. The android OS will have a Bluetooth settings page that can be accessed by the application. We will have the user connect via Bluetooth through the OS. Once the Bluetooth is connected outside the application to the sensor, it is expected that the application itself can see the sensor connected and grab data from it.

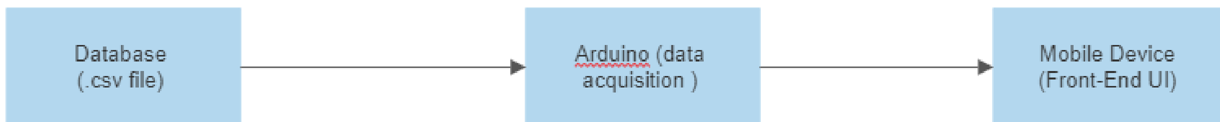


Figure 4.1: Software System Communication

4.3.2 Database

The data will be stored as a csv file in the app library. For the proof of concept there would be ten demo plants which would serve as test cases. The full description of the plant is divided into six subcategories: Plant name and photo, ideal soil specifications, type, required sunlight level, required temperature, and its full plant cycle.

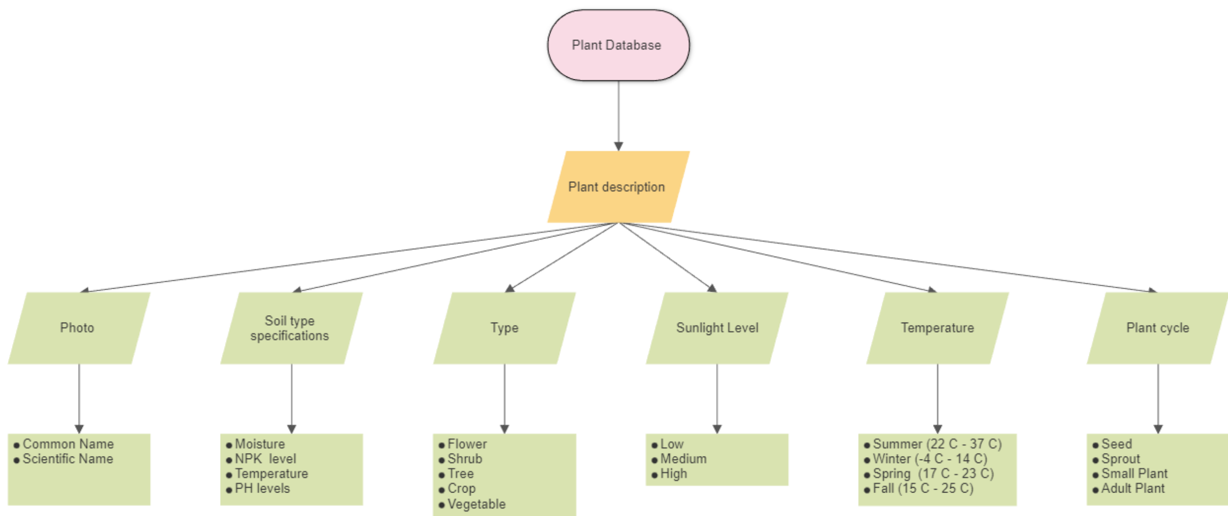


Figure 4.2: Database Schema

Design Label	*Design Description*	*Corresponding Requirement*
DES 4.3.2.1- PC	The database will be stored as a .csv file in the App library.	REQ 6.2-B
DES 4.3.2.2- PC	The database will provide a full description of the plant.	REQ 6.2-B

Table 4.3: Design Specification for Database

5 Conclusion

The design document provides a detailed description of the design implementations for DIRTS, attending specifically to the proof of concept. An elaborative explanation of the design choices, which are further justified through considering design alternatives. A summary of each major section is listed below:

1. Hardware Design
   a. Sensors
      - Temperature is detected using DS18B20 digital thermometer
      - Humidity level is detected using SEN-13637 dual probe moisture sensor
      - NPK levels are detected by using optical transducer using an LDR sensor
      - Ph level is determined by the HAOSHI Ph sensor.

   b. Power Supply
      - the arduino uses 5V power supply to run the bluetooth module thereby a typical 9V battery is used.

   c. Microcontroller Unit
      - Arduino Uno is used as a microcontroller, since it's cheaper and accessible.

   d. Bluetooth/NFC
      - Bluetooth HM-10 BLE model is used to transfer data from the sensor to the mobile application.
2. Structural Design
   a. Body Casing
      - PLA a thermoplastic will be used for the outward casing of the sensor and hardware unit, as it provides durability.
      - O-rings would be used in-between the enclosure in order to make it waterproof.
3. Software Design
   a. User- Interface -Mobile application

      Front end
      - C# and Xamarian tools will be used to develop cross-platform.

      Back end
      - A .csv file will provide the database of plants categorized accordingly.
      - soil data will be provided via bluetooth module.

The current technical document serves as the design implementation guidelines for DIRTS.

6 References

[1] E. S. F. O. of Communications, “Around your world,” *Soil pH: What it Means*. [Online]. Available: https://www.esf.edu/pubprog/brochure/soilph/soilph.htm#:~:text=Soils%20tend%20to%20become%20acidic,inorganic%20acids%2C%20such%20as%20nitric .

[2] Maxim Integrated “DS18S20 high-precision 1-wire digital thermometer.” [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS18S20.pdf.

[3] Alldatasheet.com, “Alldatasheet.com - electronic parts Datasheet Search,” *ALLDATASHEET.COM - Electronic Parts Datasheet Search*. [Online]. Available: https://www.alldatasheet.com/ .

[4]“Plant Nutrition,” Wikipedia, 08-Jul-2022. [Online]. Available: https://en.wikipedia.org/wiki/Plant_nutrition .

[5] “Show me the basics,” *What Are Nitrogen, Phosphorous and Potassium? (Feed Me)*. [Online]. Available: https://web.extension.illinois.edu/firstgarden/basics/feedme_03.cfm#:~:text=Nitrogen%20is%20used%20by%20plants,stems%20and%20keep%20growing%20fast.

[6] “Crop Plants Take Up (Absorb) Nutrients in Inorganic Form,” *International Plant Nutrition Institute (IPNI) - Publications*. [Online]. Available: http://www.ipni.net/publications .

[7] M. Masrie, M. S. A. Rosman, R. Sam , and Z. Janin , “Detection of nitrogen, phosphorus, and potassium (NPK) nutrients of ...” [Online]. Available: https://www.researchgate.net/profile/Marianah-Masrie/publication/323710446_Detection_of_nitrogen_phosphorus_and_potassium_NPK_nutrients_of_soil_using_optical_transducer/links/5b03613aaca2720ba09905fa/Detection-of-nitrogen-phosphorus-and-potassium-NPK-nutrients-of-soil-using-optical-transducer.pdf

[8] “Measuring ph with an Arduino and ph sensor - arduino tutorial,” *Latest Open Tech From Seeed*, 14-May-2020. [Online]. Available: https://www.seeedstudio.com/blog/2020/05/14/measuring-ph-with-an-arduino-and-ph-sensor-arduino-tutorial.

[9] “Gravity: Analog pH sensor / meter pro kit for Arduino,” *Arduino Online Shop*. [Online]. Available: https://store-usa.arduino.cc/products/gravity-analog-ph-sensor-meter-pro-kit-for-arduino.

[10] “HM-10 Datasheet,” *ECE 4760*. [Online]. Available: https://people.ece.cornell.edu/land/courses/ece4760/PIC32/uart/HM10/DSD%20TECH%20HM-10%20datasheet.pdf

[11] S. Mathew, “NPK soil sensor making - nitrogen phosphorus potassium levels in soil,” *Arduino Forum*, 03-Apr-2021. [Online]. Available: https://forum.arduino.cc/t/npk-soil-sensor-making-nitrogen-phosphorus-potassium-levels-in-soil/703167.

[12] “What is PLA? (everything you need to know),” *TWI*. [Online]. Available: https://www.twi-global.com/technical-knowledge/faqs/what-is-pla#IsitEnvironmentallyFriendly .

7 Appendix A: Test Plan

This section outlines the testing of the functionality to ensure correct system performance and satisfy the functional requirements. It is divided into three sections:

- Hardware Testing
- Software Testing
  - General App Testing
  - Database Testing
  - Performance Testing
- User testing

7.1 Hardware Testing

Test:	Basic Sensor Input and Output	**Time:**	____________	**Date:**	_____________
Testing Procedure:	- Connect the moisture sensor to the Arduino - Connect the digital thermometer to the Arduino - Connect the pH sensor to the Arduino - Connect Arduino to a computer and open serial monitor				
Expected Outcome:	- Moisture, pH, and temperature sensors should be uploading data to the serial monitor via the Arduino analog inputs				
Observed Outcome:					
Comments:					
Score:	▢ **Pass**	▢ **Fail**	**Tester Signature:**	______________________________________	

Test:	Basic Sensor Input and Output	**Time:**	____________	**Date:**	____________

Testing Procedure:	- Connect the LEDs and Light Dependent Resistor to the Arduino - Connect Arduino to a computer and open serial monitor
Expected Outcome:	
Observed Outcome:	
Comments:	

Score:	▢ **Pass**	▢ **Fail**	**Tester Signature:**	____________________

7.2 Software Testing

7.2.1 General Requirements Tests

This section involves regression testing the application for functionality.

Test:	Opening up Application	**Time:**	___________	**Date:**	____________
Testing Procedure:	1. Open up mobile application				
Expected Outcome:	- Mobile app should open up to the home page				
Observed Outcome:					
Comments:					
Score:	▢ **Pass**	▢ **Fail**	**Tester Signature:**	________________________________	

Test:	Plant Retrieval	**Time:**	____________	**Date:**	____________

Testing Procedure:	1. Open Mobile Application 2. On bottom navigation Tab, click Plants 3. Search for “Peas”
Expected Outcome:	- Database should be filtered only to plants with the “Peas” in its name - User should only see any result with Peas
Observed Outcome:	
Comments:	

Score:	▢ **Pass**	▢ **Fail**	**Tester Signature:**	______________________________

Test:	Connection to Bluetooth Sensor	**Time:**	___________	**Date:**	___________

Testing Procedure:	2. Open up mobile application 3. Click on Connect 4. Open to Bluetooth Settings 5. Click on "DIRTS" Device and return to App
Expected Outcome:	- In the mobile application, it should say that DIRTS is connected
Observed Outcome:	
Comments:	

Score:	▢ Pass	▢ Fail	**Tester Signature:**	________________________________

<table>
<tr><td>Test:</td><td>Android Minimum Version Test</td><td>Time:</td><td>____________</td><td>Date:</td><td>____________</td></tr>
<tr><td>Testing Procedure:</td><td colspan="5">1. Run application on emulator, Android OS Version 7.0</td></tr>
<tr><td>Expected Outcome:</td><td colspan="5">- Application should be able to run</td></tr>
<tr><td>Observed Outcome:</td><td colspan="5"></td></tr>
<tr><td>Comments:</td><td colspan="5"></td></tr>
<tr><td>Score:</td><td>▢
Pass</td><td>▢
Fail</td><td>Tester Signature:</td><td colspan="2">____________________________________</td></tr>
</table>

Test:	Soil Suggestion	**Time:**	____________	**Date:**	____________
Testing Procedure:	1. Open up App 2. Attach Sensor into soil 3. Connect to Sensor 4. Click on recommended Plants				
Expected Outcome:	- Application should display a list of plants that are suitable for the soil based on measurements				
Observed Outcome:					
Comments:					
Score:	▢ **Pass**	▢ **Fail**	**Tester Signature:**	____________________________	

Test:	Retrieval of Soil Data multiple times	**Time:**	____________	**Date:**	____________
Testing Procedure:	1. Open up App 2. Attack Sensor into soil 3. Connect to Sensor 4. Click Next to go to Analysis Page 5. Click Refresh Scan 5 Times				
Expected Outcome:	- Application should be updating the measurement to the most recent scan				
Observed Outcome:					
Comments:					

7.2.2 Database Requirements Tests

This section is for tests on the database.

Test:	Minimum 100 Plant Database	**Time:**	____________	**Date:**	_____________

Testing Procedure:	1. In application, go to plants page 2. Inspect via debugger or visually and count the number of plants in the database
Expected Outcome:	- There should be at least 100 plants
Observed Outcome:	
Comments:	

Test:	Tolerance for every plant	**Time:**	____________	**Date:**	_____________
Testing Procedure:	1. In application, go to plants page 2. For every plant existing in the database inspect via automated test, debugger, or visually whether there is a tolerance values of pH, NPK and humidity				
Expected Outcome:	- All properties for each plant are filled in database				
Observed Outcome:					
Comments:					

Test:	Database in Application	**Time:**	____________	**Date:**	_____________

Testing Procedure:	1. Inside the application folder, check for the existence of a "plants_database.xlsx" file.
Expected Outcome:	- This file should exist
Observed Outcome:	
Comments:	

7.2.3 Performance Tests

This section is in regard to testing for performance.

Test:	Performance Boot	**Time:**	____________	**Date:**	_____________
Testing Procedure:	1. Open up Application				
Expected Outcome:	- App should open in less than 10 seconds				
Observed Outcome:					
Comments:					

Test:	Bluetooth Connection speed	**Time:**	____________	**Date:**	_____________
Testing Procedure:	1. Once mobile device is connected via bluetooth, go back to the application screen				
Expected Outcome:	- App should take less than 5 seconds to know that the sensor is connected				
Observed Outcome:					
Comments:					

Test:	Crash Behavior	Time:	___________	Date:	____________

Testing Procedure:	1. Induce a crash via debug option
Expected Outcome:	- Software should exit automatically - Software should not freeze in application
Observed Outcome:	
Comments:	

Test:	Search for plant in database	**Time:**	____________	**Date:**	_____________
Testing Procedure:	1. In application go to plants page 2. Search for “Peas” plant				
Expected Outcome:	- Software should filter the result in less than 3 seconds				
Observed Outcome:					
Comments:					

8 Appendix B: Design Justification

8.1 Sensors and Casing Structure

Requirement ID	*Requirement Description*
Req 3.1-A	The system should work in both indoor and outdoor environments
REQ 5.1.1-A	The electrical circuit must be protected to reduce risk of damage or shock
REQ 5.1.3-B	Wiring must be completely contained within the device enclosures.
Req 3.4-B	The system must have the capability to be planted in soil for extended periods
REQ 5.1-B	The system must be an enclosed device to make it waterproof and dustproof.
Req 3.7-GG	The final price of DIRTS shall be less than $300CAD

Table 8.1: Requirements for Sensors and Casing structures

Design Option	*Specification*
PLA plastic for case and rubber gasket for waterproofing	The case will be light, easy to design using software, and cheap to 3d-print. The material can withstand high temperatures and provide water protection. A rubber gasket will provide the rain proofing and will prevent water from entering the case from the adjoining sections. The material will not expand or contract in varying temperatures.
Wood and metal	The case may be lightweight, but may not fulfill all the structural design requirements. The water will ruin the wooden casing and flow through the material, thus short-circuiting the electronics.
Steel or any other metal allow	A case constructed from this material will fulfill all the structural needs, but will greatly increase the weight.The cost of metal printing or creating the case from metal sheets would be expensive, and our company does not have much experience working with sheet metal.

Table 8.2: Design options for Casing and Structures

PLA plastic was selected to design the case because it satisfies all the requirements for the product in both the proof-of-concept and prototyping phase. For the final design, some additions to this option can be done to make it more sturdy and robust. Following this plan will not increase the cost and will maintain the optimal weight.

8.2 Bluetooth LE

Requirement ID	*Requirement Description*
REQ 3-A	The controller must establish a connection with the wireless device via Bluetooth
REQ 6.1 - B	Able to establish a connection to sensor remotely

Table 8.3: Requirements for Bluetooth LE

Design Option	*Specification*
Bluetooth LE	Bluetooth LE has similar capabilities to standard bluetooth. This version of bluetooth uses less energy than standard bluetooth, which is important for our sensor. The main downside is it does not have the ability to constantly stream data. However, due to requirement 6.1-GG this is not a requirement.
NFC	The main disadvantage of NFC is that the range is short and may not satisfy the consumers requirements. The main concern is that the plant will grow, and it may be difficult to connect to the sensor via NFC.

Table 8.4: Design options for Bluetooth LE

Bluetooth LE was selected as it is the most suitable for all the requirements for all phases. As long as streaming data within milliseconds does not become a requirement, Bluetooth LE will remain the optimal choices for its decent range and low energy usage.

8.3 Microcontroller Unit

Requirement ID	*Requirement Description*
REQ 5.2.3-A	Microcontroller and all the sensors must operate between 3.3V to 5V
REQ 5.2.5-A	The sensor must be able to provide analog signals to the MCU
REQ3.2-A	The controller must establish a connection with a wireless device via Bluetooth or Wi-Fi if possible

Table 8.5: Requirements for Microcontroller Unit

Design Option	*Specification*
Arduino Uno	This microcontroller is relatively easy to use and satisfies all the requirement specifications. A minor drawback is that the modules need to be purchased separately and programming them as a whole can take some time. There is no built-in bluetooth, so buying an external module and working with it can cause difficulties. The size of the board makes the device larger than necessary, which may cause some issues regarding accessibility.
Raspberry Pi	This microcomputer satisfies all the requirements and comes with the built-in modules. The size of the board can be an issue, but can be compensated by the efficiency of the board. It provides its own OS for simplicity. However, despite all the advantages, they are difficult to get because of the current on-going chip shortage, resulting in an increase of cost.
Particle Photon	This device can be viewed as a small but more efficient alternative to the Arduino. This device already comes with in-built wifi chip and a free cloud storage along with the ability of coding via Wi-Fi. This device uses the same coding environment as Arduino so a switch is easy. The price is also comparable to the Arduino.

Table 8.6: Design options for Microcontroller Unit

From the above design options, the Arduino Uno is chosen for the proof-of-concept phase as it is easily available, and our team members have more experience with it. The modules can be configured easily via Arduino. Although, a switch can be made in the prototyping and final stage of the product to the Particle Photon. This MCU meets all the requirements listed without compromising the product cost of the device. In addition, the transition would be easy as both the Arduino and Photon share the same coding environment and provide the cloud storage ability as well. Not only would the efficiency be increased, but also the size of the device would decrease significantly, providing the user a better device handling experience.

8.4 Mobile Software

Requirement ID	*Requirement Description*
REQ 6.1-A	Software must be able to run on a mobile device/tablet
REQ 6.1-B	Run on Android OS version 7.0 (API 24, Nougat) and newer.

Table 8.7: Requirements for Mobile Software

Design Option	*Specification*
Mobile	The mobile option allows the user with a smartphone to have access to a bluetooth option to connect to the sensor. The user will also be able to use the sensor outside easily. The main issue is that a mobile option tends to lose CPU power, which makes it weak for data processing. The other issue is that the screen size may make it difficult for the user to read information.
Computer	The computer option allows the user to have access to a more powerful CPU which will allow for better data processing speed. The mobile issue can also be alleviated by utilizing laptops instead of desktop computers. The screen size is also bigger as well. The downside is that the user’s computer may not have bluetooth capability, so a USB adaptor may be required.
Embedded	By having a display on the sensor itself, it may be possible to ignore the usage of bluetooth in order to receive data. This may also reduce the complexity of the project. The downside is that the user will likely have a more difficult time understanding the sensor’s results.

Table 8.8: Design options for Mobile

Our choice is to use mobile design due to the higher availability and mobility of the device. We think that there is not enough data processing, so a weaker CPU is okay. The user is mainly concerned with whether the soil is recommended or not, so there is not a lot of information that is required for the user to read. So the screen size being smaller than a laptop is okay. The mobile device fits our requirements for all phases of our product, and is the ideal choice.

Design Option	*Specification*
Android	Android apps generally have a more flexible environment than the iOS. The main advantage comes with the interaction with hardware, which the android development environment will have an easier time than with iOS. Another advantage is that this particular team also has more access to android devices than iOS. The disadvantage is that quality assurance is often more difficult as there are multiple android platforms for multiple phones.
iOS	iOS apps are also a great start for development. They are often faster apps and it is easier to develop a strong UI. The main downside is that the environment is fairly restrictive, only being able to run on Macs. In order to develop for this, Apple technology is required.

Table 8.9: Design option for Mobile Software

For initial development in PoC, Android is the clear choice. The main reason is that more people in the team have access to android devices so it will make starting development easier. We also believe that the more flexible environment will make coding the bluetooth data transfer easier. In the beta phase and in the final product, we can easily transfer our code with Xamarin to convert the Android application to an iOS one. We can utilize resources provided by the SFU school to help us develop for the iOS application, primarily the Mac computers.

8.5 Back End :System Communication

Requirement ID	*Requirement Description*
REQ 6.3 - B	Connection to sensor via Bluetooth must take less than 5 seconds when less than 10 cm from device

Table 8.10: Requirements for BackEnd -System Communication

Design Option	*Specification*
Standard Bluetooth	Standard bluetooth would work for our purposes however we leave the advantage of lower battery drain with Bluetooth LE. The main advantage standard bluetooth would bring is the ability to stream data constantly. However, by requirement 6.1-GG we need a set of data only when the user presses a button. Thus, we do not need standard bluetooth.

Table 8.11: Design option for BackEnd -System communication

Among these options, the standard bluetooth was selected as the system communication because it satisfies all the requirements while ensuring the ability to stream data constantly with lower battery drain.

8.6 Back End : Database

Requirement ID	*Requirement Description*
REQ 6.2-PC	Storage data of at least 10 plants
REQ 6.2-B	Each plant will have tolerances specifying pH , NPK , temperature and humidity
REQ 6.2-B	Database will be a .csv file within the app library.

Table 8.12: Requirements for Back End - Database

Design Option	*Specification*
Storage data set: 10	The following data set is limited to 10 as in the initial stage it's providing as test cases based upon general knowledge.
Category of plants	The plant's description is categorized into 6 different categories since the plant growth also depends upon the weather and sunlight. It is an attempt to include different factors with respect to soil specification.
.csv database	The database for the plants would be included in the app as a .csv file since we do not have a large chunk of data in the POC stage , but in beta phase we can expand it as an SQL database.

Table 8.13: Design option for Back End - Database

Among these options , .csv was selected as the database because it satisfies all the requirements as the data used is not large and .csv can simply solve the purpose of providing plant data for the App.

9 Appendix C: User Interface

9.1 Introduction

This section of the document provides a detailed evaluation of all the prerequisites to ensure that our intended audience can effectively utilize the primary function of our device, DIRTS.

9.1.1 Purpose

The purpose of this appendix is to give our readers an in-depth understanding of interaction with DIRTS to make it as simple and efficient as possible.

9.1.2 Scope

The presented information in this document will go over the detailed technical and user analysis as well as detailed testing plans. The sections will outline our target users and required experience they must have. Following this, the UI will be analyzed by breaking it into the seven elements of UI interaction along with the required engineering standards for safe operations. In addition to this, the appendix will discuss analytic and empirical user testing to ensure productive outcome. For analytical testing, the device will be tested to make sure that needs of a user are fulfilled, and for empirical testing, hands on experience will be provided to some users before actual launch.

9.1.3 Intended Audience

The document is for the whole team of Everyday Planting Solutions to use as a guideline for developing and improving DIRTS and as well as during the testing phase. The document can also be used by instructors and teaching assistants of ENSC-405W (Capstone-A) as a reference guide to make it available for future classes.

9.2 User Analysis

The target market of DIRTS includes the general public, who want to grow some plants at home and don't know where to start. This market could be expanded further towards small scale agriculture and botany labs to monitor the plants. Since the large share of our market consist of people with no prior experience in gardening, a user analysis was conducted and using that the following list of necessary requirements was created:

· Users should have a basic knowledge in operating a smartphone, specifically connecting to Bluetooth

· Users should know how to plant in pots as well as in their garden.

· Users should know where to find the required gardening supplies.

However, if the user is unable to fulfill these capabilities, a reference manual will be provided to help. Any other person can perform these tasks for the user on their behalf with the help of instructions given by the app associated with DIRTS.

9.3 Technical Analysis

The analysis underlined in this section is based on how we follow the principles listed in Norman's Seven elements of UI interaction. These elements will help us to form an intuitive design in any system, and provide the best possible user experience.

9.3.1 Discoverability

Discoverability is one of the key elements for a good design as it relates to the user dependence on the working of the product. It indicates how easily users can understand how to operate the device or familiarize themselves with the product just by looking at the interface. While designing, the primary functions of the device and corresponding software will be determined, and they will be focused in a strongly discoverable way. The table below shows the primary functions and tasks of our device and their considered discoverability.

Function/Task	**Design Consideration**
Powering up the Device	A switch is placed near the battery compartment with ON/OFF positions marked.
Orientation of the sensor	The orientation in which the sensor goes into soil is marked and the probes are sticking out from the same side.
Setting up the App and default sensor settings	The app will provide a login page for the user to set up the device and store their own settings.
Bluetooth connectivity and data gathering	The app will pop-up the screen while connecting to the sensor and gathering data via Bluetooth.

Table 9.1 Discoverability Tasks

9.3.2 Feedback

The importance of feedback from the physical device is to indicate to the user that it is functional and to give alerts about different states once an action is performed. Taking this into consideration with our device DIRTS, the source of feedback is our mobile application.

In the proof-of-concept phase, the feedback will be provided through the LEDs over the microcontroller in addition to some small messages over the app. However, in prototyping stage, EPS decided to include the following feedback commands for a better responsive experience:

- Feedback will be provided when successfully logged into the application.
- App provides feedback in the form of vibration or sound when it is connected to the sensor successfully.
- A warning signal for the low battery status will be given as feedback.
- When data transfer is complete, a notification will be provided along with viewing data options.
- Once a matching set of plant and soil are found from the database, a notification will pop-up displaying the results.

9.3.3 Conceptual Models

A conceptual model allows users to understand the functionality of a product easily and in a better way. Keeping this in mind, DIRTS' innate design will allow users to use the product right away with just a few flicks of switches. Since the majority of our projected market shares consist of non-technical and inexperienced people, extensive user testing will be performed in order to identify user problems and work-arounds will be created as solutions. The menu of the app will be kept simple and compatible with various devices to be convenient for users of any age.

9.3.4 Affordances

Affordance relates to a quality of design that describes the relationship between the appearance and intended use of the product. It helps an individual to understand how each part of the device should be used. The probes sticking out of the case in our device will tell users the orientation for placing the device into the soil. The app will provide various options to select, which will include connectivity, visualization of data, and sensor status. All the buttons will be formatted to fit the screen to be easily accessible.

9.3.5 Signifiers

Signifiers ensure affordance, discoverability and well-defined feedback. It is important that the user is able to determine correct actions for each aspect of the device. All the information from the sensor will be provided directly to the smartphone, which will act as a signifier for our product. The pop-up notifications will notify the user about:

- Connectivity status of the sensor
- Data collection and corresponding results
- Low battery with a warning of changing the battery

In addition, a small LED on the sensor will also provide information about the battery status and when the device is in use.

9.3.6 Mappings

The motive behind good mapping is to create an innate design layout where a user can easily understand the controls of the device. Since the sensor goes into the soil and stays there, the only mapping is for power on/off. The rest of the controls are done via software buttons such as navigation buttons, action item buttons, and a log in/out button. The navigation buttons have a natural mapping for going back and forth in the menu, while action items are mapped for viewing specific types of data gathered from the sensors.

9.3.7 Constraints

Constraints limit the actions that can be performed by the user, resulting in increased usability of design and reducing the chance of operating error. There is a physical constraint on the sensor, preventing the device from being submerged too deep in the soil. Software constraints are also imposed within the app. The main software constraint is regarding connectivity via Bluetooth, which is handled by using a generic low-powered Bluetooth device compatible with all current versions of Bluetooth. Another main constraint is the storage of data, since the memories on both the phone and sensor device are limited. However, this can be managed by defining a length of time after which the previous data is overwritten with newly gathered data. The last constraint is separating the collected data with the existing database when comparing.

9.4 Engineering Standards

Engineering standards specify the technical details and characteristics that should be fulfilled by the product. Our team heavily focuses on ensuring that the validation and structural planning of the product takes place according to the given standards by various organizations. A widely used standard for human-computer interaction is the ISO 9241 from International Organization for Standardization. In addition to this, the International Electrotechnical Commission (IEC) has laid out guidelines for excellent implementation of physical user design. DIRTS will be built and tested against the following engineering standards.

Standard	**Description**
IEEE 802.15.1	Standards for WPAN/Bluetooth
IEEE 829	Standard for Software and System Test Documentation
CSA C22.2 No. 0-10	General requirements - Canadian electrical code, part II
CAN/CSA-C22.2 NO. 61508-1:17	Functional safety of electrical/electronic/programmable electronic safety-related systems
CSA C22.2 No.0.23:15	General requirements for battery-powered appliances
C22.2 No. 205-17	Signal Equipment
IEC 60335-1:2020	Safety of electrical appliances for household and similar purposes
IEC 60065	Audio, video, and similar electronic apparatus – Safety requirements
IEC 61508	Standard for functional safety of electronic safety-related systems

Table 9.2: Engineering Standards

9.5 Analytical Usability Testing

Analytical usability testing describes the approach to address flaws and critical usability issues before getting any user feedback. Keeping this in mind along with the user analysis, the UI, and the general functions of DIRTS are kept simple. Ideally, the system will not require any complex updates, and will be implemented in a user-friendly, easy to understand manner.

Each designer will perform the following test cases without any bias or prior discussion about the outcomes. These test cases are proposed with a 5-star rating scale where 1 is Bad, and 5 is Excellent.

The tasks are categorized based on proof-of-concept phase and prototype phase of the design and are listed below in two different tables.

Task/Test for PoC Phase	**1**	**2**	**3**	**4**	**5**
The device is easily powered on/off					
Circuit connected via breadboard to Arduino properly					
All sensor probes are differentiated properly					
Sensor connects easily to the phone via Bluetooth					
The menu items are consistent with the design and easy to follow up					
Information retrieved from database is clear					

Table 9.3: PoC analytical testing

Task/Test for Prototyping Phase	1	2	3	4	5
The layout of the device is intuitive					
The power and reset button on the sensor are easily accessible and understandable					
Battery replacement is easy					
The instructions on the software are easy to understand and follow					
Notifications and error messages are expressed properly					
Everything described in plain text (i.e. no jargon)					
Received data is categorized and stored properly					

Table 9.4: Prototype analytical Testing

9.6 Empirical Usability Testing

To achieve our goal of developing an easy-to-use product meeting the desired design standards, members of EPS perform empirical usability testing as a requirement for DIRTS' design process.

Since the interface will be minimal during the proof-of-concept stage, the user testing capabilities will be limited. There are several test cases to be completed by SFU faculty or selected engineers. The purpose is to ensure that the usage of the device is possible and safe. Even though our target market does not necessarily have a technical background, this will provide a baseline of feedback for our prototype design. Taking these results into account, necessary changes will be implemented and new features will be tested. The process will repeat until both the device testers and our company are confident about an accessible and valid design.

When the product is in development stages, testing with a desired clientele will begin. This may include allowing elderly or middle-aged people with various levels of gardening experience to use our product temporarily in order to gauge the complexity of the system to a new user. The testing audience can be expanded by including people working in nurseries and botany labs. Our team members will provide a user manual and a demonstration to familiarize the user with all parts and functionality. Once the initial user training is complete, the device trial phase begins and all team members are to refrain from answering questions. This is to observe the user experience and recognize any complications that may arise during the physical device setup, any compatibility issues, and the general application usage. The Users/Testers will be asked to fill a survey to evaluate the accessibility of all features, give suggestions for improvement, and comment about how useful the product is.

After analyzing the collected feedback data, necessary design revisions will be made to create an updated version of our device, if the deadline allows. At the end, a final round of testing will be executed for quality assurance and the design will be finalized for DIRTS.

Test Environment: Indicate all the parameters and occasions where the device was used

Rating Scale:1-5 (1: Very Bad and 5: Excellent)

Questions	**1**	**2**	**3**	**4**	**5**
First Impression of the DIRTS sensor					
How well reference manuals are documented?					
How easy is it to set up the device along with the app?					
How well does the sensor go with your plants?					
How well can you recognize the notifications?					
How intuitive was the application design?					
How sturdy/robust does the sensor feel?					
Does the system provide you with all information that you should know?					

How likely are you to use this device daily?					
How likely are you to recommend this device to others?					
Additional Comments:					

Table9.5: User Feedback form

9.7 Projected Appearance

9.7.1 Hardware Appearance

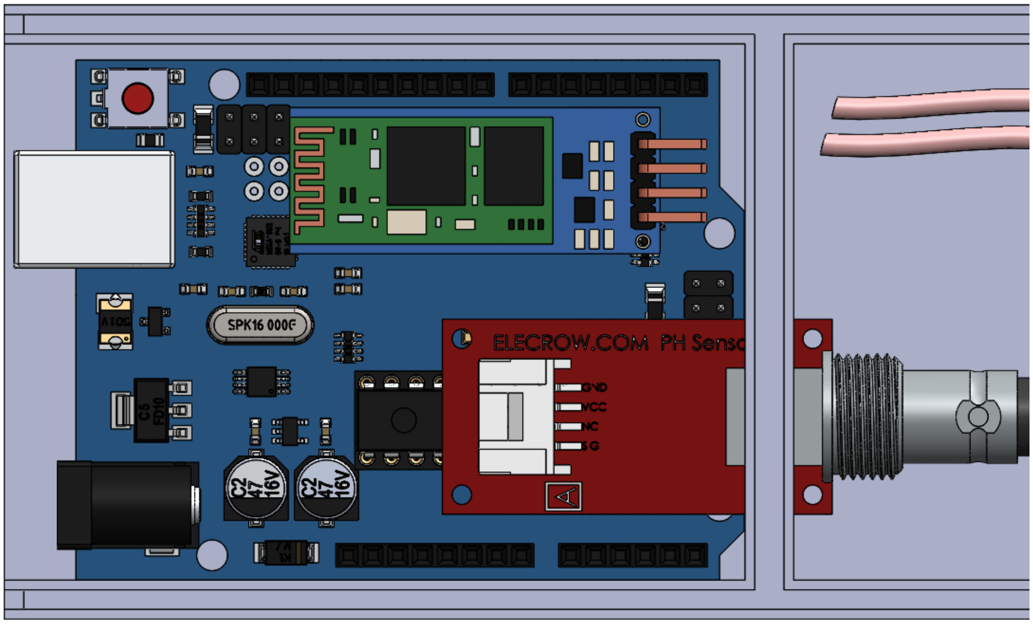


Figure 9.1: Top Schematic View of DIRTS

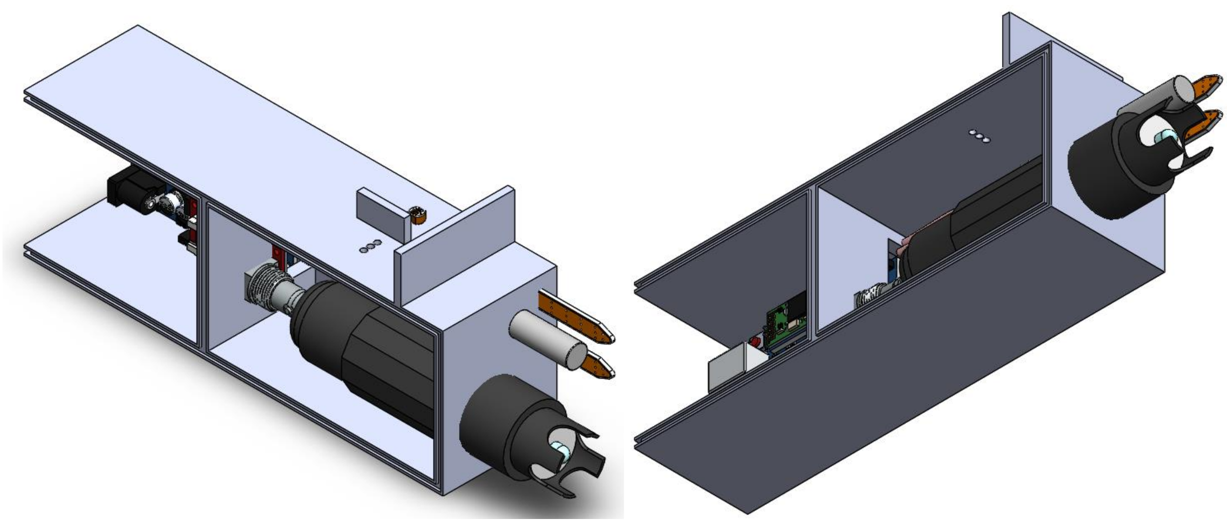


Figure 9.2: Outside Perspective Views of DIRTS

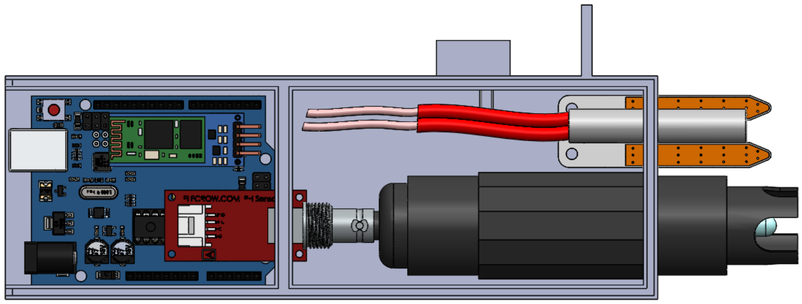


Figure 9.3: Side Schematic View of DIRTS

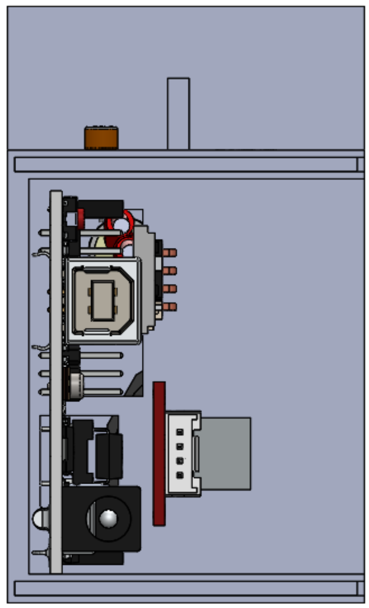


Figure 9.4: Back Schematic View of DIRTS

9.7.2 UI Appearance

Below shows the UI we will have for our mobile application. It will consist mainly of 4 pages: Home Page, Analysis, Plants, Settings.

9.7.2.1 Home Page:

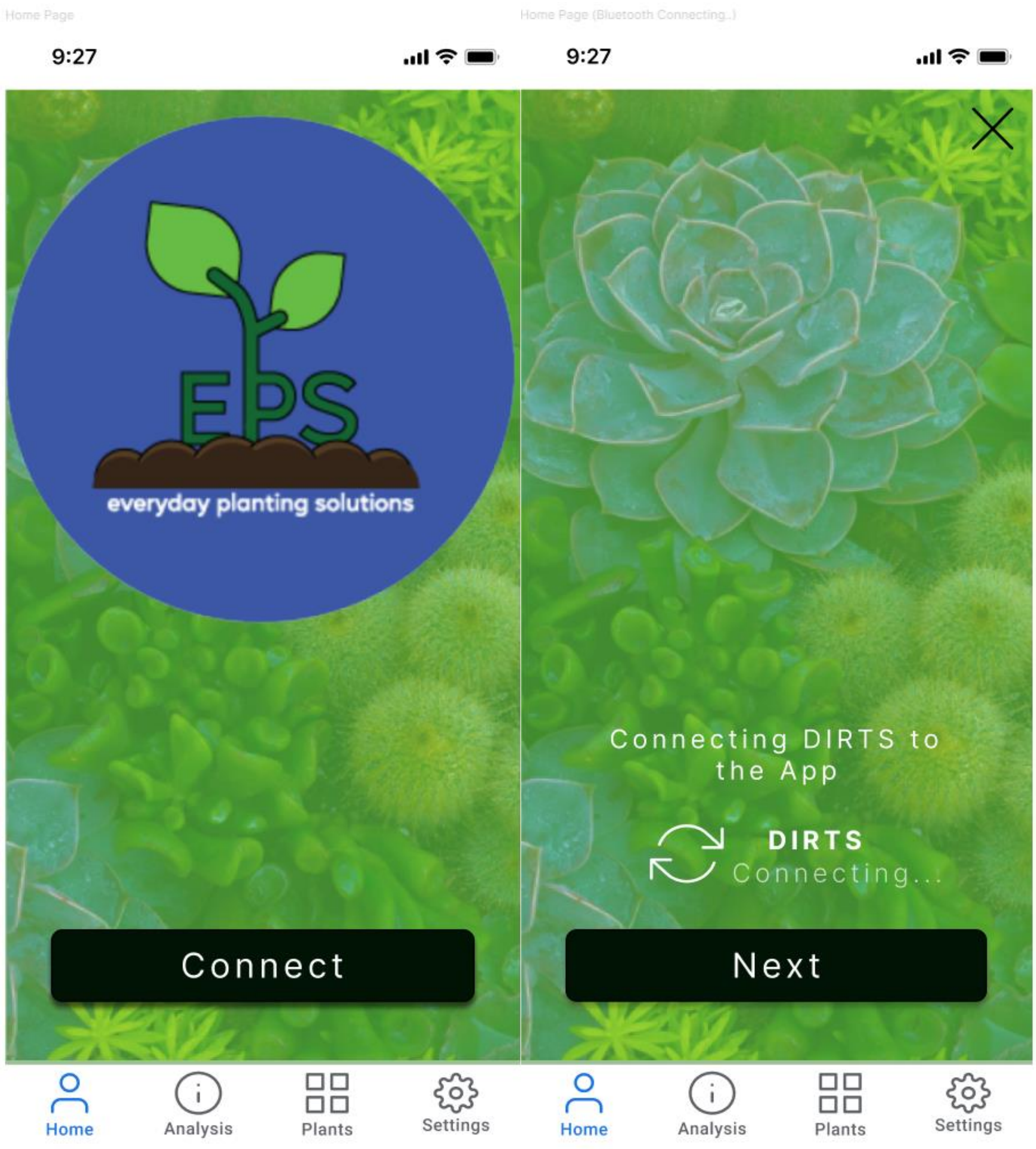


Figure 9.5: Software UI Prototype Home

9.7.2.2 Analysis Page:

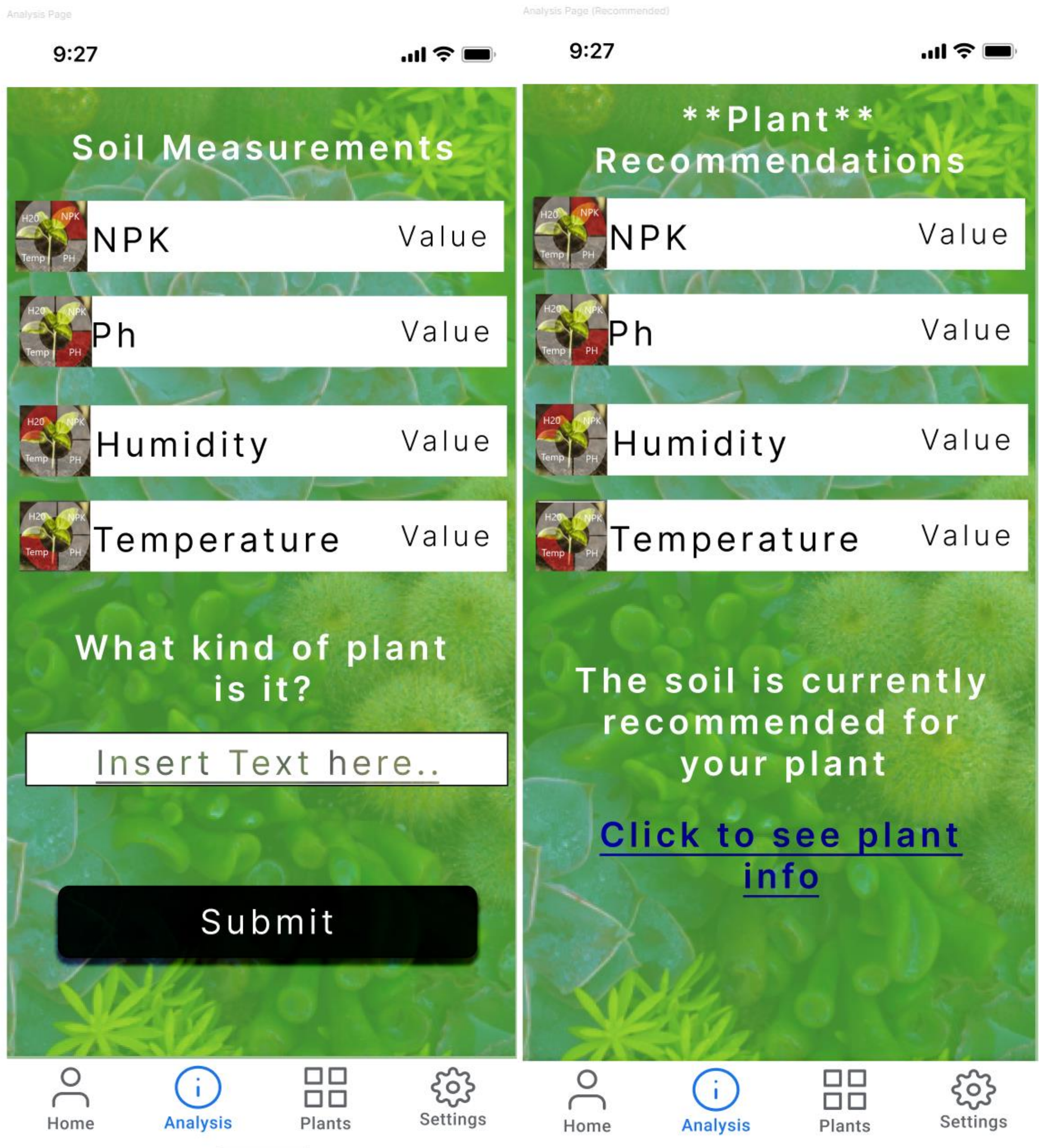


Figure 9.6: Software UI Prototype Analysis

9.7.2.3 Database Page:

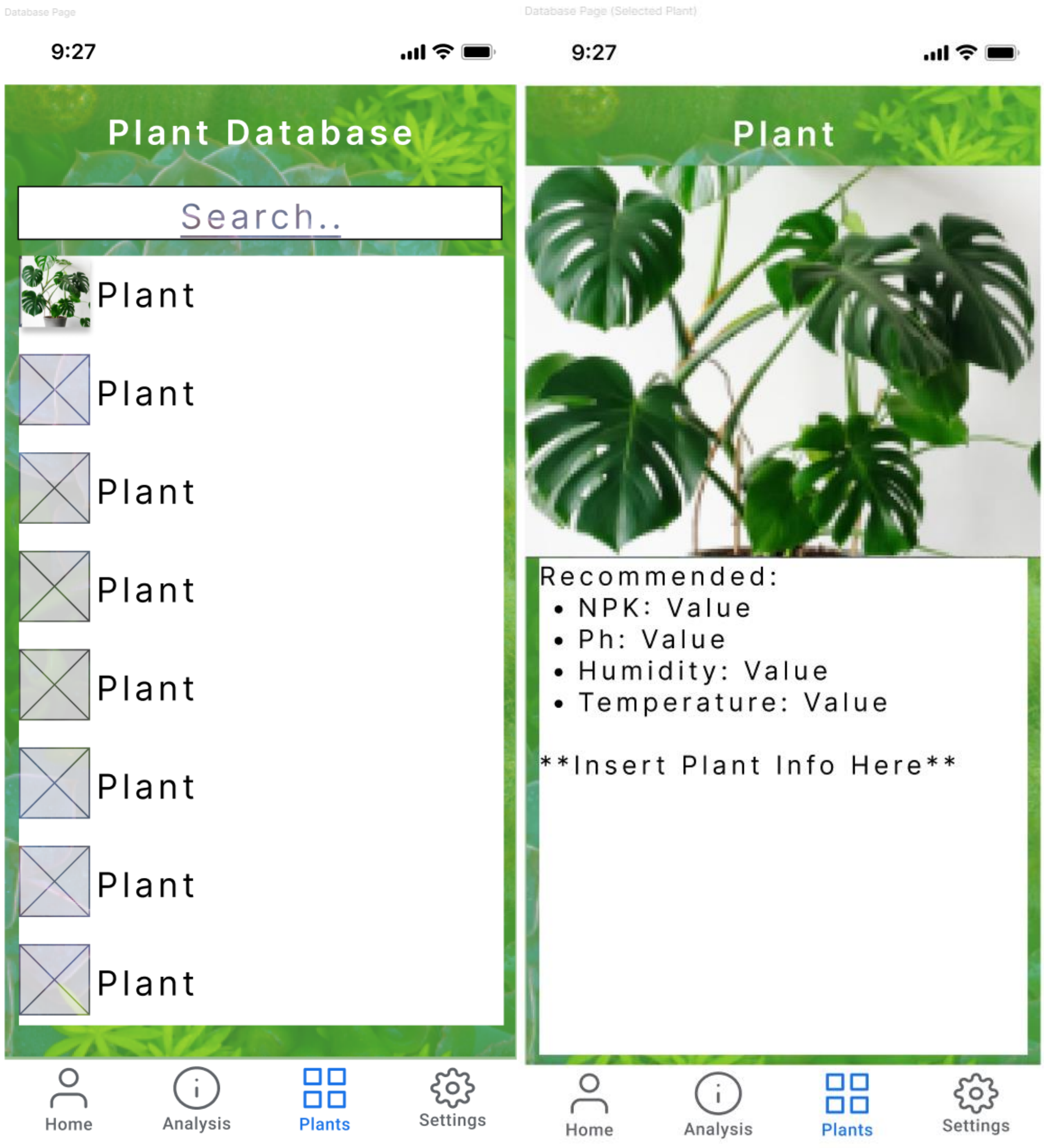


Figure 9.7: Software UI Prototype Database

9.7.2.4 Settings Page:

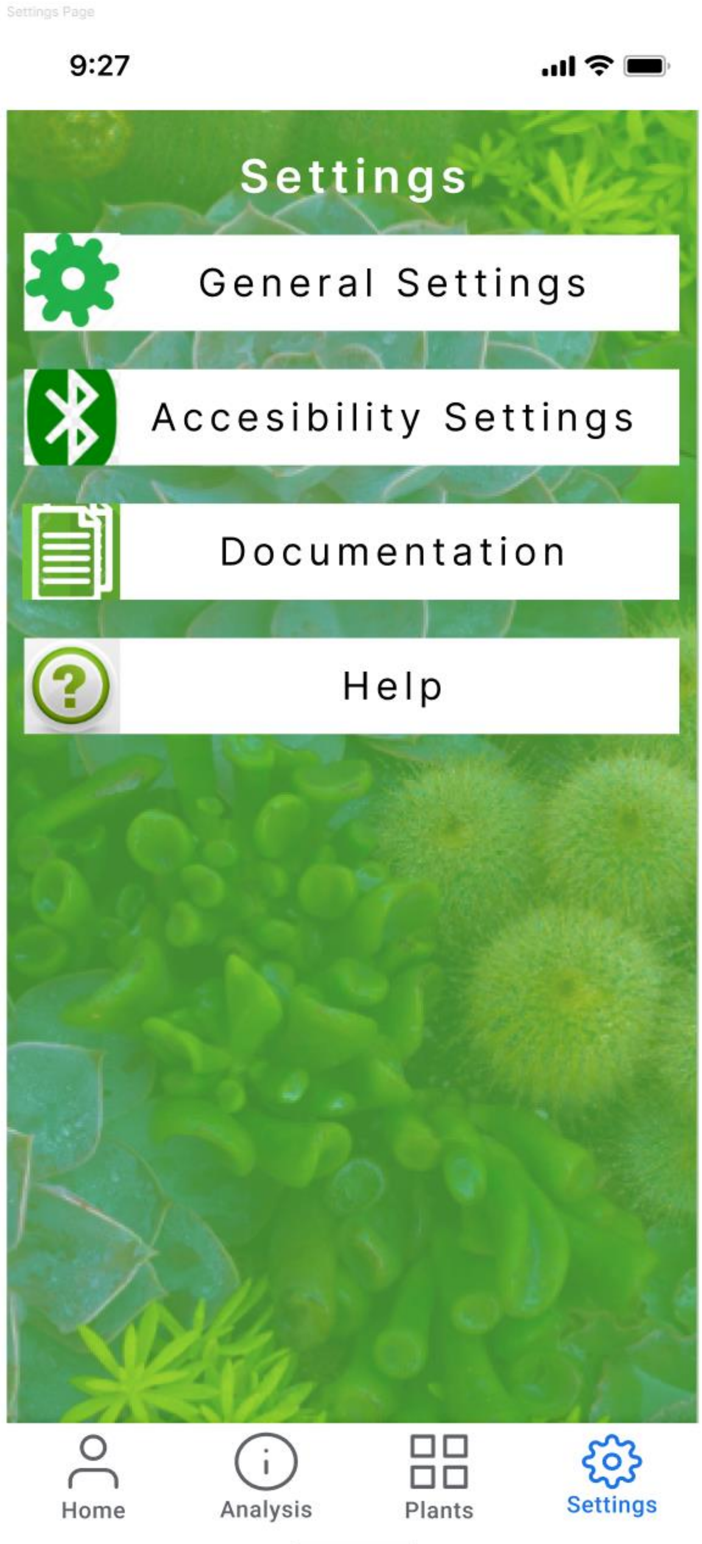


Figure 9.8: Software UI Prototype Settings

9.8 Conclusion

The User Interface Design appendix outlines the design choices made for the development of DIRTS sensor with a corresponding mobile application. The document also covers various test methods used by our team in creating the best possible user interface. Our company, Everyday Planting Solutions (EPS), has taken into consideration the demands and knowledge level of our target market while giving a complete analysis of the system. Currently, the device is in the proof-of-concept phase which involves the underlying circuitry of the sensor on breadboard combined with the app for providing basic testing functionality. Therefore, the focus is mainly on fulfilling the requirements of the PoC phase. Future work includes 3D printing the shell for the circuitry and sensors along with the expansion of software to various other platforms. The next four months will be utilized for the development of an aesthetic, easy-to-use prototype that will invite more users to engage with the product.