CSCE 4981
Senior Project II

Group 6
Software Design Description v2.0

Written By
Farah Seifeldin - 900160195
Lobna ElHawary - 900160270
Maram Abbas - 900153570
Mariam ElZiady - 900161869
Samer Basta - 900150910

Supervised By
Dr. Mohamed Shalan
Dr. Tamer ElBatt

May 22, 2021

1. Introduction

1.1 Purpose

This Software Design Document (Revision v2.0) is meant to be provided as reference for developers during the development process as it provides a detailed description of the **plantiful** system, along with its subsystems, hardware components, and software components, and also to be provided as a source of technical information for our clients to know how the system works.

1.2 Document conventions

The text is all Times New Roman.
Heading 1 is of font 18 and bold. It signifies the start of a section.
Heading 2 is of font 16. It signifies the start of a subsection.
<u>Heading 3</u> is of font 14 and underlined. It signifies a subsection within subsections (Heading 2).
Heading 4 is of font 13 and in bold. It signifies a subsection within subsections of subsections (Heading 3).

In **Section 6**, italics is used for the page names.

Normal text is of font 12. When bold is used here, it is to reference sections or subsections.

Figure numbering for the attached figures will be of font 12 and in bold. The first number will denote which section and the following numbers denote which subsections and the final number will denote the order of the figure in the section. So, **Figure 4.3.5** means this is the 4th section, 3rd subsection and this is the 5th photo in this subsection. So, **Figure 5.3.5.1** means this is the 5th section, 3rd sub-section, 3rd sub-sub-section and this is the 1st photo in this subsection. **Figure C** means this is in appendix C.

1.3 Intended audience

The intended audience for this Software Design Document is split into subgroups: firstly, future software engineers or developers looking to expand this project further by adding more features or testing existing features. Secondly, the smart-agricultural professionals that would want to understand how such a system works. Third and lastly, clients of our project, i.e. CARES' employees (where we are implementing our idea initially as a Proof of Concept) and any future users.

1.4 Additional information

Currently, we are collaborating with CARES, as they already have an equipped greenhouse where they run their experiments, which serves as a great central hub for our own testing. Their current experiment is on cucumbers, so that's the crop type we'll be focusing on. So, currently, the CARES' greenhouse is our Proof of Concept, and some of our decisions and design implementations currently are biased towards their physical structure. However, most of our components are generalized to work with greenhouses of multiple sizes, and it could even work outdoors (granted there's internet and sources of electricity to supply to our hardware components).

1.5 Contact information/SDD team members

Farah Seifeldin	-	farahseifeldin@aucegypt.edu
Lobna ElHawary	-	lobna1443@aucegypt.edu
Maram Abbas	-	maram_abbas@aucegypt.edu
Mariam El Ziady	-	mariamelziady@aucegypt.edu
Samer Basta	-	samerr@aucegypt.edu

2. Design Overview

2.1 Background Information

Our project is an intelligent agriculture monitoring system for a cucumber greenhouse in the Center for Applied Research on the Environment and Sustainability (CARES) that uses IoT and AI. As a community-driven project, its purpose is to generally aid those in the agricultural sector either conducting experiments on different plants to gain a better understanding of the effects of soil moisture, pH, temperature and humidity levels on their plants' yield and health, or to aid in regulating crops that are meant for harvesting to ease the process and allow for precision agriculture. The AI portion involves checking for the growth stage of the plants and plant health detection.

As our project is currently centralized around the CARES' cucumber greenhouse, seeing that this is where we can run our tests, we are taking more generic design approaches so that our design isn't completely biased by CARES. However, our design currently is based on the results we're achieving as we are operating in this greenhouse, so it won't be completely unbiased.
Moreover, we discussed with CARES' leading engineer what outcome they would like to see from a project like ours, and that's how we got the set of requirements for our project. Those requirements were in line with what we've researched would be ideal for other smart greenhouses. The number of components to purchase though will remain centralized around their greenhouse, as this is the only spatial reference we have.

For some background information regarding the agricultural section, their testing involves huge blocks of the planted crops, with each block having a different set of values for the numerous factors that affect plant growth (water amount, heat, nutrients, etc.) to compare and contrast between them and see which grows the fastest, which has better yield, the quality of yield, etc. For that reason, the three sensors measuring the four sensor measurements are the most crucial measurements. Checking for the growth stage to see how fast it's growing to see how well the experiment is doing in terms of the plant's rate of growth. Finally, the health status of the plants are important to track how the experiment is affecting the health of plants.

2.2 Constraints

2.2.1 Cost

This is the largest constraint, as our project involves a lot of physical parts, some of which are quite expensive. Furthermore, since we are creating a fully functional system for CARES, it involves a larger quantity of these expensive components. We have found solutions to minimizing the number of components required, but some components are required (i.e. the pH sensors or a high quality IP camera).

Component	Cost/$
ESP32	4.2
DHT11 Sensor	1.9
Soil Moisture Sensor	3.7
pH Sensor	18
PCB	4.3
Four 2300mAh Batteries	10
Total	**42.1**

Figure 2.2.1 - Sensor Node Cost

2.2.2 Data

As we were trying to find ready-made data sets, we couldn't find many open-source datasets, and those that weren't open source would not communicate back to our requests to access their datasets. Luckily, we found three datasets for our goals, but one is restricted to cucumbers, another limited to 4 crops, and the last limited to 14. These are decent numbers and sufficient for us currently, but not nearly enough for the thousands of different crops out there for future expansion of our software.

2.2.3 Resources

Due to the ongoing COVID-19 Pandemic, it has been proven difficult to ship electronic components from abroad. We were planning on ordering a multi-sensor with integrated Wi-Fi capabilities to improve the quality of our project, as well as an EC sensor which would have fulfilled a small requirement for CARES. Both the multi-sensors and the EC sensor are expensive due to shipping, and cost is already a big constraint of ours. However, this is possibly the least of our concerns due to the simple solution that we can gather these multi-sensors individually alongside an ESP32 for the Wi-Fi capability and that the EC sensor is only a very small requirement of a really big project.

2.3 Design Trade-offs

We are using Wi-Fi instead of Bluetooth (more specifically BLE) for the communication between the Raspberry Pi 4 and the ESP32s.
The tradeoff here is having longer range over energy efficiency, as BL has around half the energy consumption of WiFi, but WiFi has a larger range so it can be used across larger greenhouses with more obstacles that wouldn't let waves go through.

Each ESP has a singular sensor node connected to itself.
The tradeoff here is portability over cost, as it was made clear to us that CARES would want to randomly move the sensor block across 17 different plant pots within the same row, and most likely other research greenhouses would want to do the same, so portability for them would triumph over reducing the costs of the ESPs by half.

We are implementing a web application over a mobile application.
The tradeoff here is accessibility and availability over *complete* readability, as we want to make sure that no matter the device on hand of our user, that the information is always available (as a web application is accessible through phones and desktops). Having a mobile application would allow us to develop with a tiny mobile screen in mind to have *complete* readability with the graphs, the buttons, etc, but by following proper design rules and rigorous testing on the phone, the readability achieved on the phone is well maintained.

3. System Architecture

3.1 Hardware Architecture

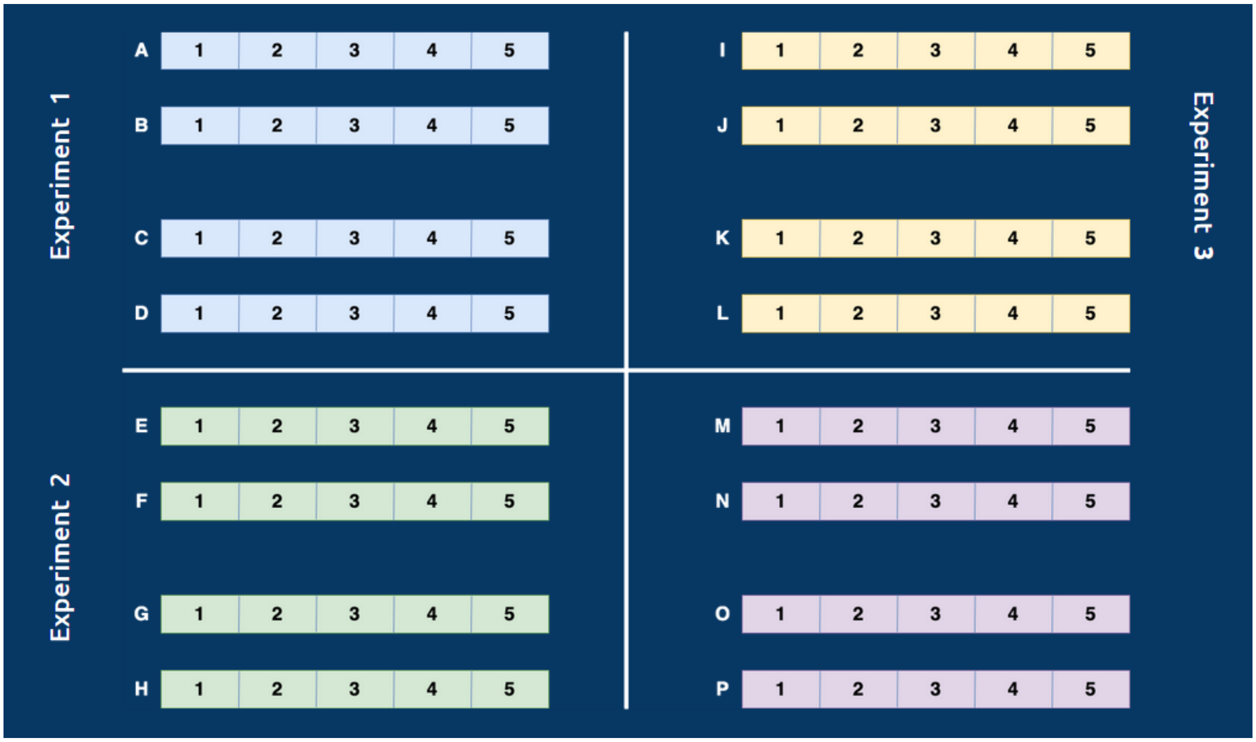


Figure 3.1.1 -CARES' Cucumber Greenhouse

This is how the greenhouse is structured, with each four rows named as a "group" for simplicity's sake by the CARES' engineers. Currently, all three of these experiments are on cucumbers, but each one will take in a different set of parameters to their growing mediums to differentiate them. They could also potentially, randomly change things very slightly between the different rows in the same group for further testing.

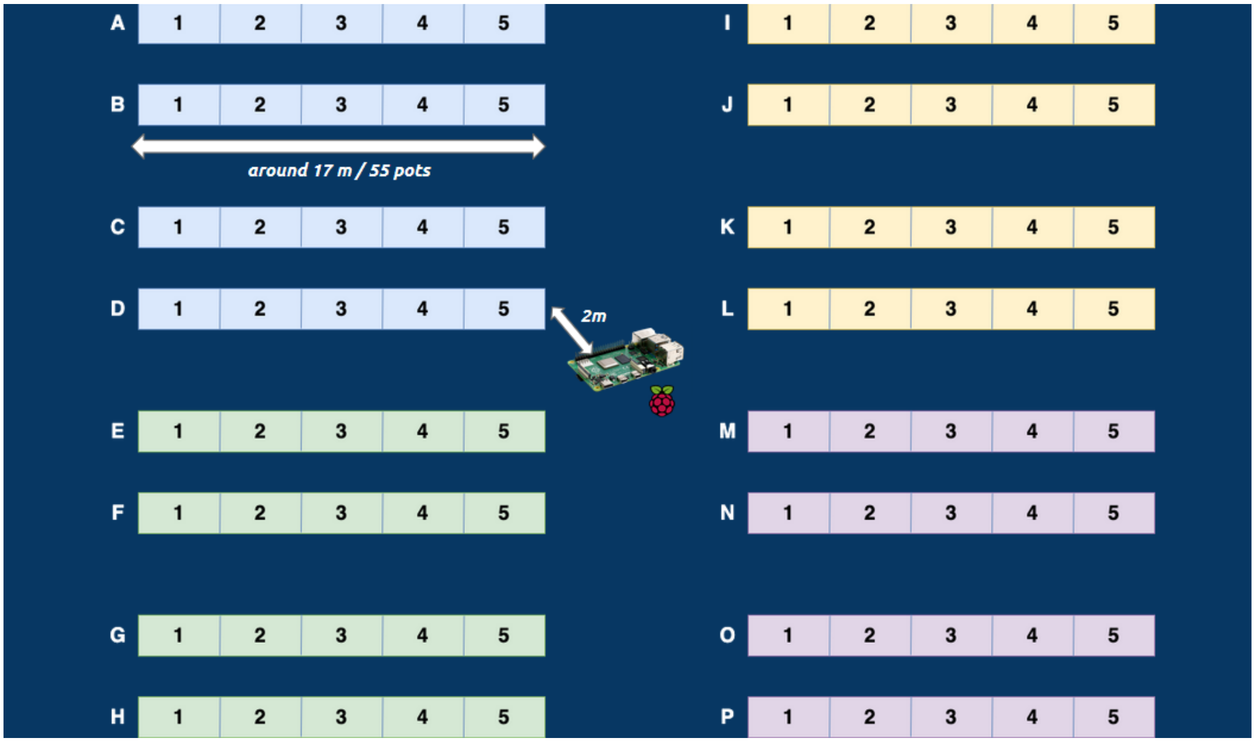


Figure 3.1.2 - CARES' Cucumber Greenhouse with Measurements

This is the overall look of the greenhouse with actual measurements from CARES' greenhouse. Each row is 17 meters and has 55 plant pots. With our Raspberry Pi in the center of the greenhouse as it is the center of operations. Each sensor node can be placed randomly in any pot, chosen by the CARES' engineers, and susceptible to being changed mid-experiment. The camera will be stationed on top of an automated car, to take photos between the 2 rows it's placed in.

In Section **5.1 Hardware Detailed Design**, we will get more into the detail of the overall hardware design.

3.2 Software Architecture

Having the RPi 4 to receive data from the ESP32 and uploading it to the PostgreSQL database is what mainly happens in this section. Briefly, the ESP32 acts as a host and the RPi 4 is a client to it. The RPi 4 sends requests periodically to the ESP32 and hence, it receives the requested data. It is the job of the RPi 4 to then upload it to the database as the most recent reading taken from the sensors.

In Section **5.2 Software Detailed Design**, we will get more into the detail of the overall software design.

3.3 Communications Architecture

Generally speaking, our main communication is based on WiFi between the ESP32 and the RPi 4, and WiFi between the C6CN camera and the RPi 4.

The sensors are connected to the ESP32 through physical connections.

In Section **5.3 Communications Detailed Design**, we will get more into the detail of the overall communications design.

4. Data Design

4.1 Database Management System Tables

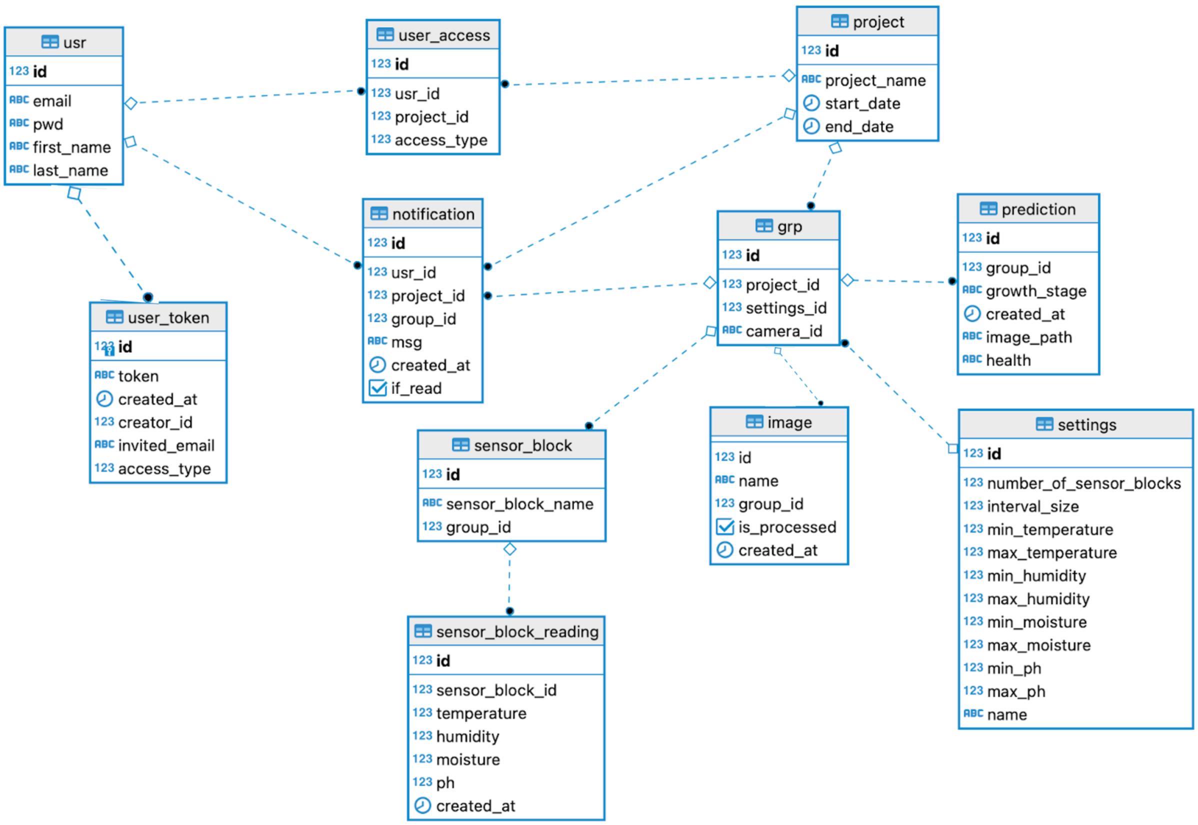


Figure 4.1.1 - Database Entity Relationship Diagram

Figure 4.1.1 is used to better explain how our Database Management works.

The **user_token** entity is for the temporary security tokens for creating your account.

The **image** entity is for images and to know which group it belongs to.

The **sensor_block_reading** entity is for the real-life sensor readings taken by our sensor nodes.

The **sensor_block** entity is for knowing which sensor nodes belong to which group.

The **prediction** entity holds the analysis from our plant analyzers.

The **user_access** entity is for knowing the type of access a user has (viewer, editor, creator).

The **notification** entity is to hold any notifications to be sent to any user from any project/group.

Finally, **usr** is for the users, **project** holds the project details, **grp** holds the group details.

5. Detailed Design

5.1 Hardware Detailed Design

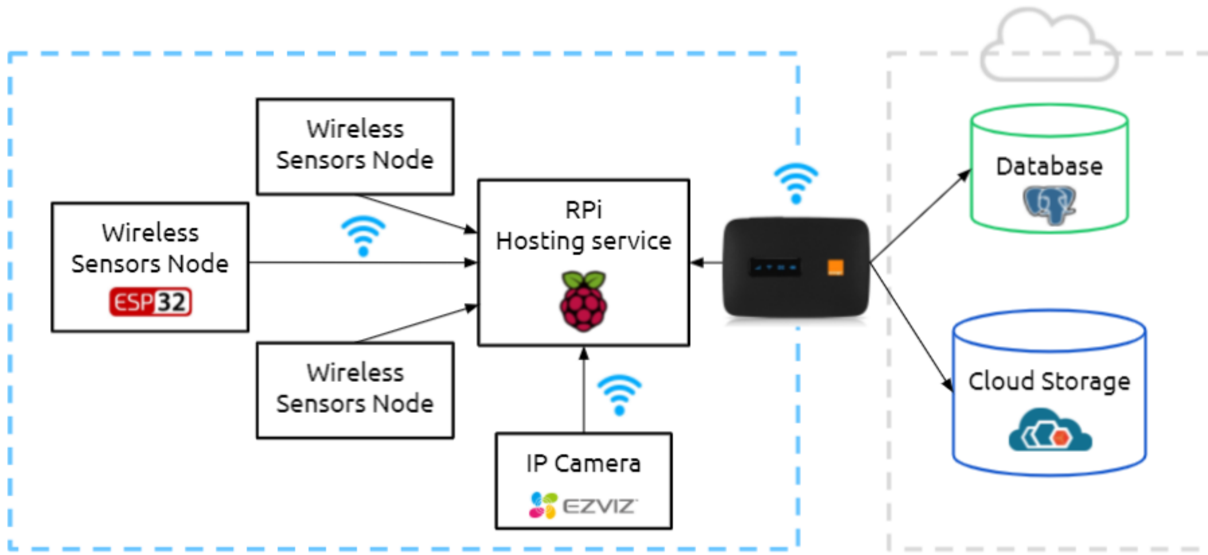


Figure 5.1.1 - Remote-sensing System Architecture

For a more detailed look on the hardware design, we are showcasing the different hardwares connected between one another as they communicate with the entire system as a whole. The sensor nodes, RPi, and IP camera are all connected in the same network provided by the Orange Mi-Fi as shown in the figure above. The power supply for the ESP32s is provided by 2300 mAh rechargeable AA batteries.

There is a singular camera connected to an automated car. The camera, the EZVIZ C6CN model, streams wirelessly, and the RPi whilst knowing the IP of the camera, can access this stream to take periodic images as the car moves. The car's timing is linked to the image-taking timing.

There is a singular Raspberry Pi 4 in the center, connected to a power source, that receives all sensor readings from the ESP32s (sensor nodes), and the images from the camera to be uploaded to the cloud for our web application and ML model.

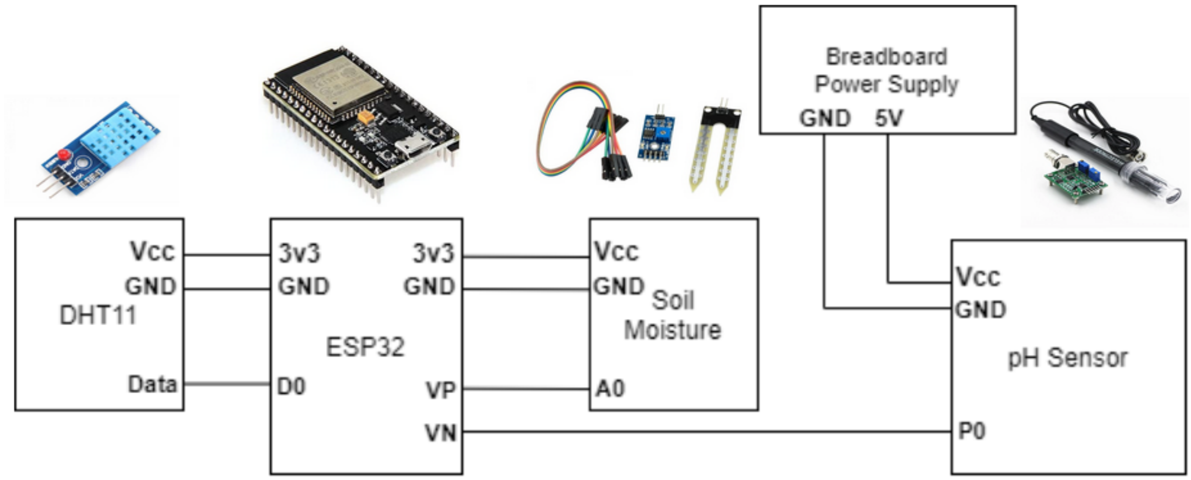


Figure 5.1.2 - Sensor Node Connections

For a more detailed look on the hardware design of the ESP32, Figure 5.1.2 was prepared.

The DHT11 (temperature and humidity sensor) is connected to the 3.3V pin on the ESP32, and its Data pin is connected to the D0 GPIO pin of the ESP32.

The Soil Moisture sensor is connected to the 3.3V pin on the ESP32, and its A0 pin is connected to the Analog VP GPIO pin of the ESP32.

The pH sensor is connected to the same 5V power supply powering the ESP32, and its P0 pin is connected to the Analog VN GPIO pin of the ESP32.

All of this is placed onto a PCB. Figure 5.1.3 was prepared to showcase this.

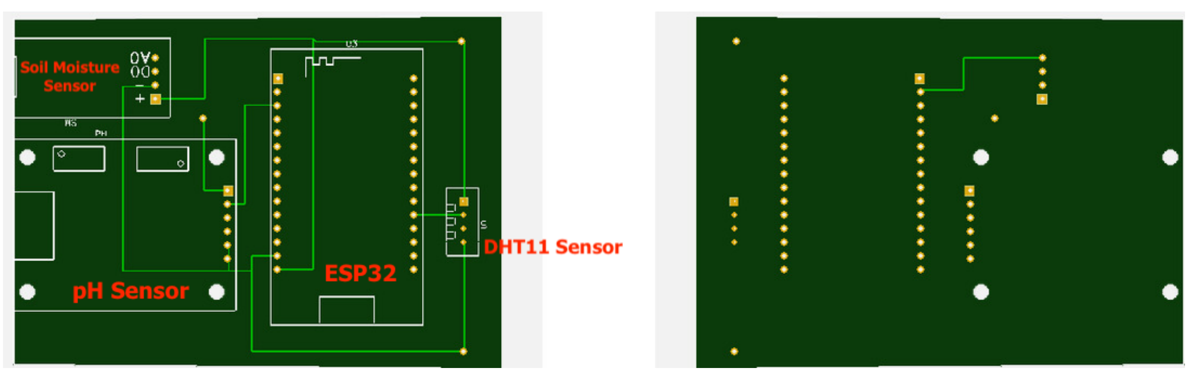


Figure 5.1.3 - PCB Layout for Sensor Node

Low Power Mode	Active mode
• ESP32: 0.8mA • DHT11 Sensor: 0.05mA • Soil Moisture Sensor: 30mA • pH Sensor: 1.5mA	• ESP32: 160mA • DHT11 Sensor: 0.45mA • Soil Moisture Sensor: 30mA • pH Sensor: 20mA

Average power consumption = (0.8 + 0.05 + 3 + 1.5)mA(0.99167) + (160 + 0.45 + 30 + 20)mA(0.00833) = 7.06mAh

Lifetime = 2,300mAh / 7.06mAh = 325.85 hours ~ **14 days** usage before requiring a recharge/replacement

Figure 5.1.4 - Power Analysis (assuming 10 minutes reading interval time)

5.2 Software Detailed Design (With a design of each component)

As mentioned before, the ESP32 will act as a server to the RPi 4. The RPi 4 will send a request to the ESP32 asking for the data it needs. When it gets a reply, it will instantly upload the received information to the PostgreSQL.

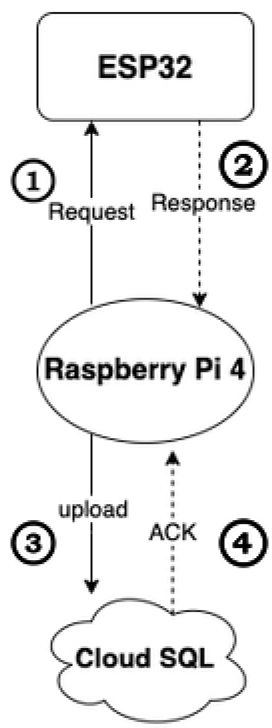


Figure 5.2.1 - Detailed Software Design

Figure 5.2.1 shows the loop that the RPi 4 goes through. As shown, when it is time to update the readings of the sensors, the RPi 4 sends a request to the ESP32. The ESP32 will respond back with the data. Then, the RPi 4 will upload it to the PostgreSQL. When it is uploaded, an acknowledgment is sent to the RPi 4.

5.3 Communications Detailed Design

In this subsection, we will be going through three sequence diagrams that will paint the three main scenarios of our project that involve communication between our different components and subsystems. One of these scenarios is how the user interacts with the frontend of the application. The second scenario is describing how sensor readings are updated in the PostgreSQL along with retrieving it. Lastly, the third scenario describes how the images taken from the camera get stored on the Heroku Storage. In the description of each one, the specific communication protocols will be mentioned.

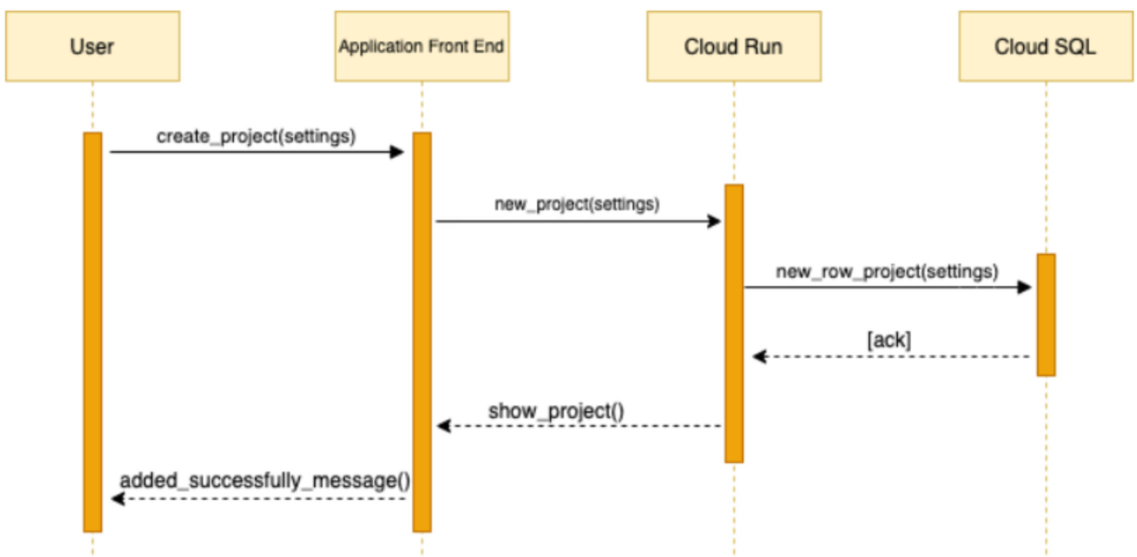


Figure 5.3.1 - New Project Sequence Diagram

Figure 5.3.1 is a sequence diagram showing the steps taken whilst creating a new project through our web application.

When the user presses on the create project button with the settings chosen, this is handled by the application communicating with the Heroku over WiFi, sending it a request to place this project into PostgreSQL. Once PostgreSQL receives the request, it'll send back an acknowledgement, which will return back to the application that the project has been successfully added. This scenario is just one example of how the user will communicate with the application. There are several other scenarios concerning the user interacting with the application using its other features. Some of these features include creating an account, editing a project, viewing sensor readings, downloading CSV files showing the trend of the past sensor readings for a specified period of time, and much more.

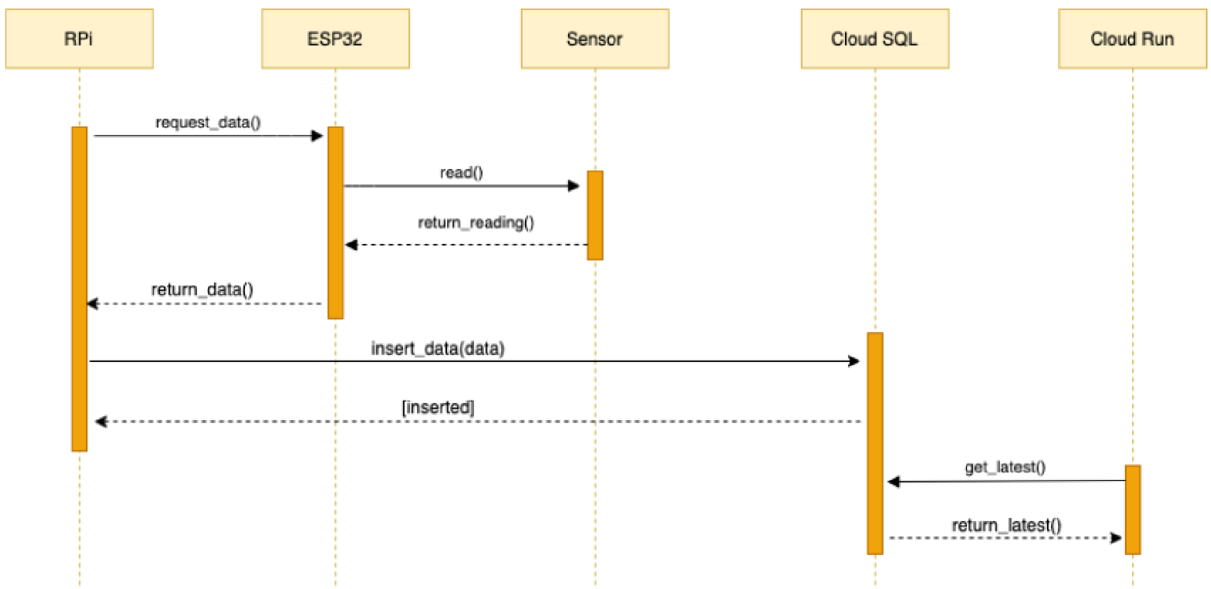


Figure 5.3.2 - Sensor Readings Sequence Diagram

Figure 5.3.2 is a sequence diagram showing the steps taken whilst retrieving the sensor readings all the way from the sensors, unto Heroku to be available for whenever the user opens the page. The RPi, through WiFi, will communicate with the ESP32 requesting the sensor data. The ESP32 will read the data through the physical wiring, and through WiFi return back the sensor data to the RPi. The RPi, through WiFi, will attempt to insert the data into our PostgreSQL database, and respond back with an acknowledgement that it was placed. When the user attempts to open up the sensor readings page of our web application, Heroku, through WiFi will get the latest sensor readings from our PostgreSQL database.

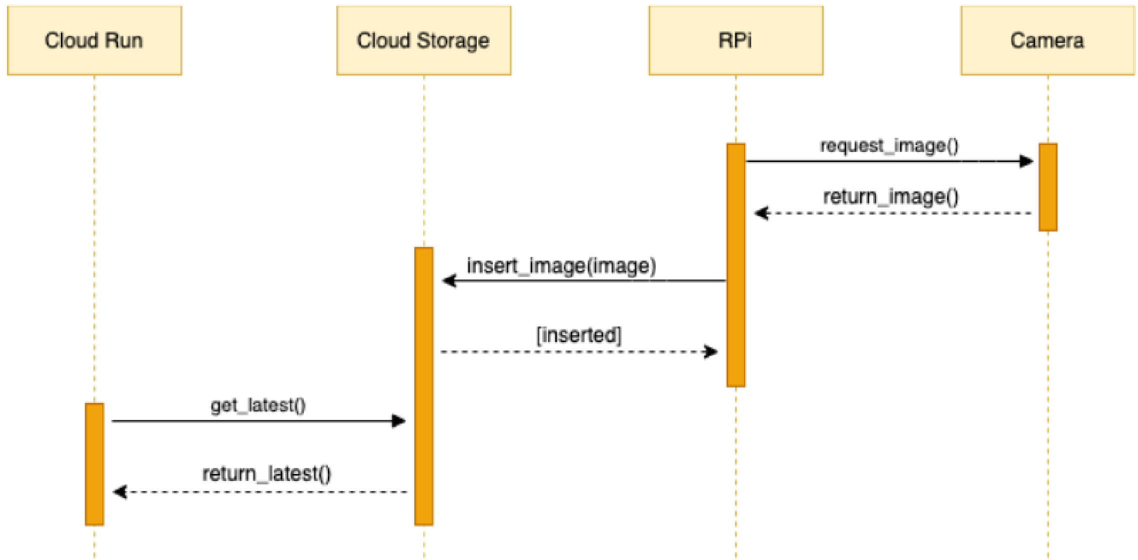


Figure 5.3.3 - Camera Processing Sequence Diagram

Figure 5.3.3 is a sequence diagram showing the steps taken whilst retrieving the camera photos all the way from the cameras, unto Heroku to be available for whenever the user opens the page and to be used in our Machine Learning models later.

The RPi, through WiFi, will communicate with the camera requesting to take a snapshot of the current frame the camera is on. The camera will take a snapshot of the current frame to be saved to the RPi. The RPi, through WiFi, will attempt to store the image into Heroku's Storage file system, and respond back with an acknowledgement that it was stored. When the user attempts to open up the images page of our web application, Heroku, through WiFi will get the latest photo from Heroku Storage file system.

6. Usability Design Approach

6.1 User Interface Overview

The user will interact with the system through a web application. The user will be able to perform the following actions:

1. Register for a new account
2. Login
3. Change password
4. Reset password through an emailed link if old password was forgotten
5. Initialize settings for new project
6. Initialize settings for new groups within a project manually
7. Initialize settings for new groups within a project by selecting the settings of a previously created group
8. Name sensor blocks that exist within each group
9. Create new project with groups and sensor blocks
10. Share a project with another user
11. Change project settings
12. Change group settings
13. View latest sensor readings
14. View line graphs of sensor readings
15. Download sensor data
16. Receive and view notifications
17. Generate token (in case of admin user)

Action	**Interface Feedback**
Register for new account	The interface displays the registration form and validates any input data including a token generated by an admin user to authenticate the registering user.
Login	The interface redirects the user to the projects page if the login is successful.
Change password	The interface allows the user to change their password.
Reset password through an emailed link if old password was forgotten	If the user has forgotten their password, the interface redirects them to a page where the user enters their email. If the email belongs to an already registered user, then a link is sent to this email that redirects the user to a page where they can reset their password.
Initialize settings for new project	The interface allows the user to enter project settings for a new project.
Initialize settings for new groups within a project manually	The interface allows the user to enter the settings for a group within a project by filling out all the required input manually.
Initialize settings for new groups within a project by selecting the settings of a previously created group	The interface allows the user to enter the settings for a group within a project by selecting the name of the settings of a previously created group.
Name sensor nodes that exist within each group	The interface allows the user to name each sensor node within a group
Create new project	The interface allows the user to create a new project after all the project, group, and sensor settings have been filled.
Share a project with another user	Share a created project with another user by entering that user's email.

Change project settings	The interface allows the user to change the project settings of any existing project.
Change group settings	The interface allows the user to change the group settings of any existing group.
View latest sensor readings	The interface allows the user to view the latest sensor readings of soil moisture and pH, as well as the room's temperature and humidity in numeric form, of a selected project's corresponding group.
View line graphs for sensor readings	The interface allows the user to view a line graph of the sensor readings of soil moisture and pH, as well as the room's temperature and humidity, of a selected project's corresponding group.
Download sensor data	The interface allows the user to download sensor data as a csv file
Receive and view notifications	The interface pushes the user notifications, and enables them to see all previously sent notifications.
Send token (if admin user)	The interface allows the user to generate, and send a token to a chosen email, so that the receiver of the token can register an account.

6.2 Sample Screens

6.2.1 Login Page

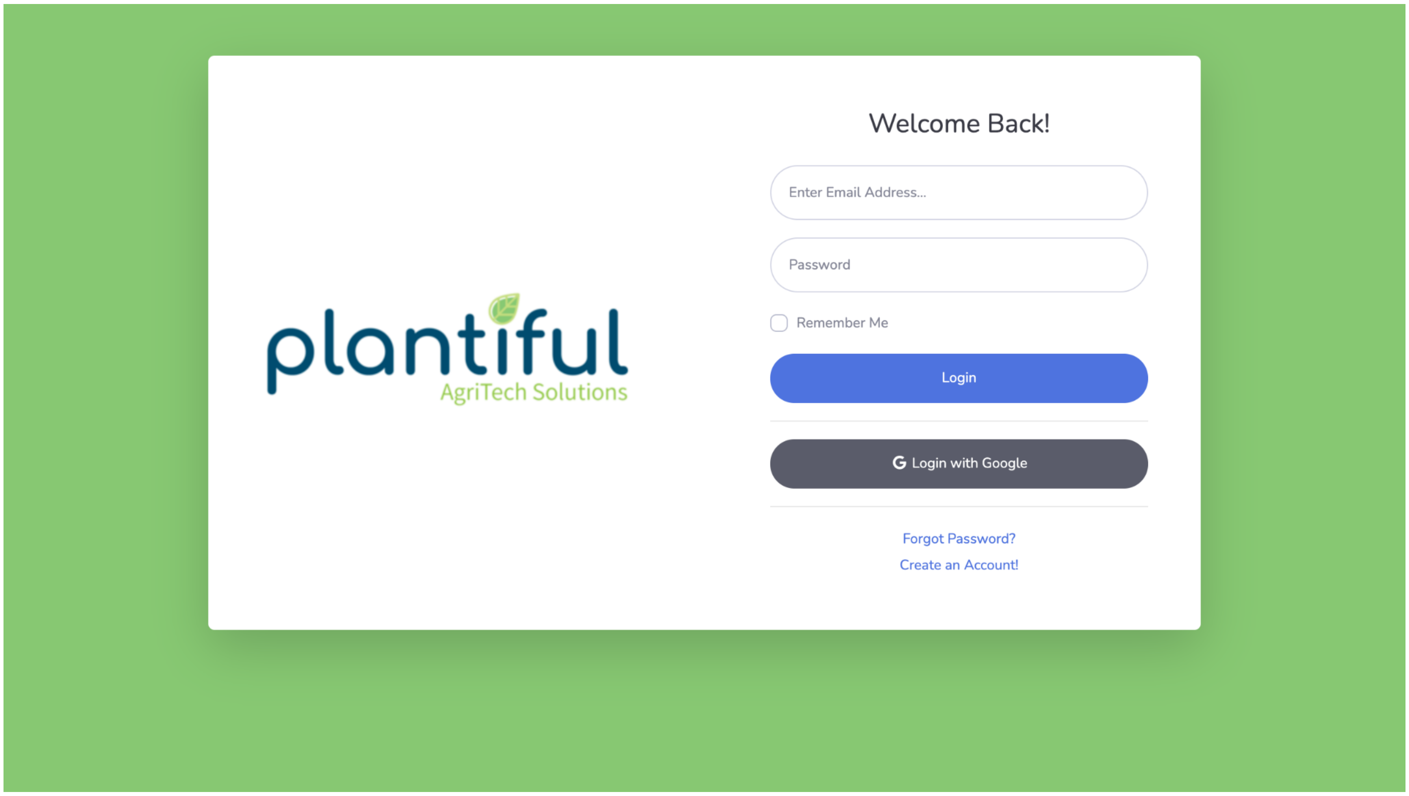


Figure 6.2.1.1 - Login Page

In this page, the user should enter their registered email and password, and press on the "Login" which will direct them to the *home page*.
If the user is not yet registered, then they will press on the link that displays the text "Create an Account!", which will lead them to the *registration page*.

6.2.2 Registration Page

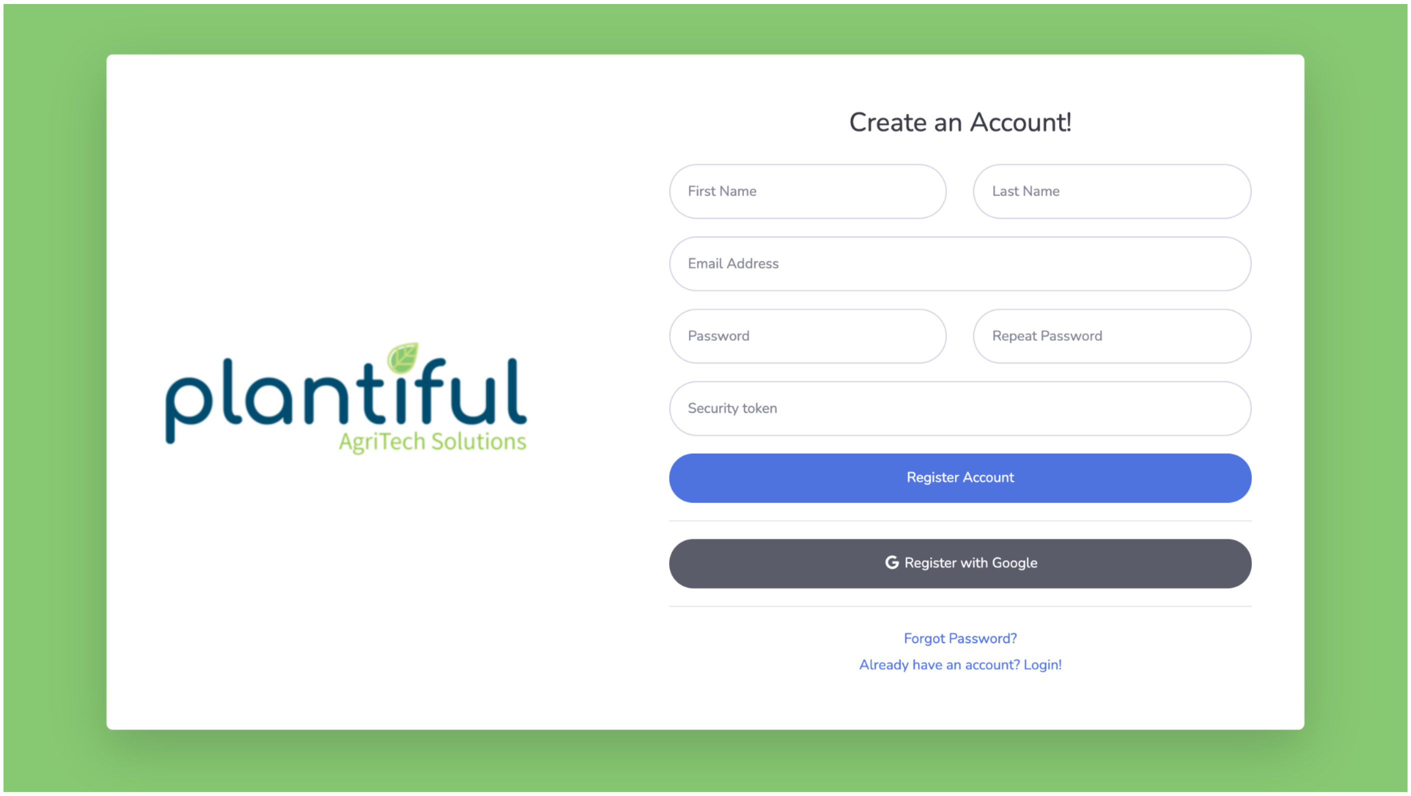


Figure 6.2.2.1 - Registration Page

The user should input the information required for them to sign up.

If all the fields are correctly filled, and the user presses on the "Register Account", then the user is registered in the system, and is redirected to the *login page*.

If the user is already registered, then they should click the "Already have an account? Login!" link which will redirect them to the *login page* straight away.

6.2.3 Home Page

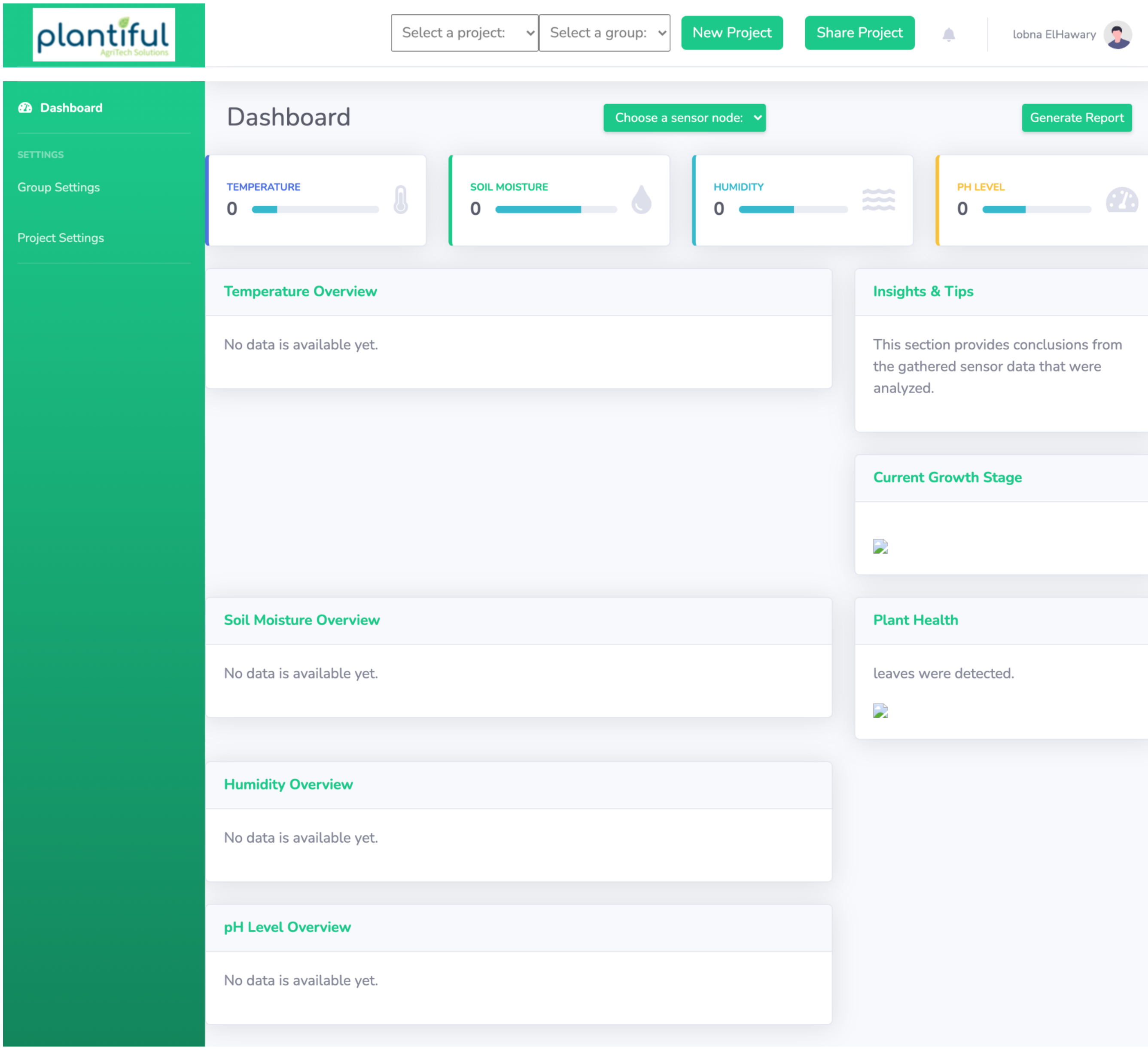


Figure 6.2.3.1 - Home Page

As mentioned before, this is the initial page users are directed to once they are logged in. The page allows users to:

- Add new projects by pressing the "New Project" button which directs them to the [new project settings page.]
- Select an existing project from the drop down menu with "Select a project:" text, and then after this selection is made, the groups within the selected project will be available in

the drop down menu “Select a group:” for the user to select from. Once the project and the corresponding project’s group are selected, the user can then select from the sensor nodes that belong to that corresponding project and group, or they can select “average”. In this example, the selected project and group contain 2 sensor nodes named 24:62:AB:FF:66:F8 and 30:AE:A4:18:BE:8C so the “Choose a sensor node” drop down will produce the following options:

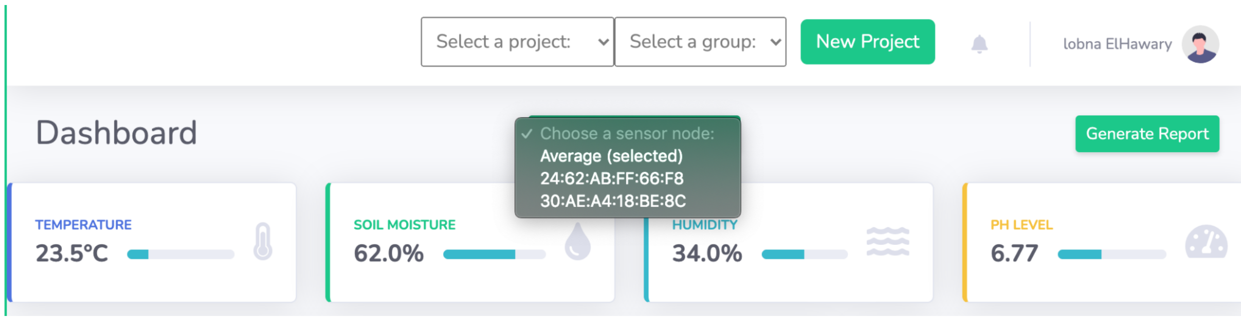


Figure 6.2.3.2 - Dashboard

If the user selects “average”, then the average of the latest sensor readings of each sensor node in the selected project and group will be displayed (in this case then the average of the sensor readings of sensor nodes 24:62:AB:FF:66:F8 and 30:AE:A4:18:BE:8C) in numeric form in the boxes titled “Temperature”, “Soil Moisture”, “Humidity”, and “pH level”. Additionally, the average results of the sensor readings over time will be displayed in the form of a line graph which is shown in the figure below.
If the user selects a sensor node’s name instead, then the same applies, but instead of average results, the results displayed correspond to only the selected sensor node.

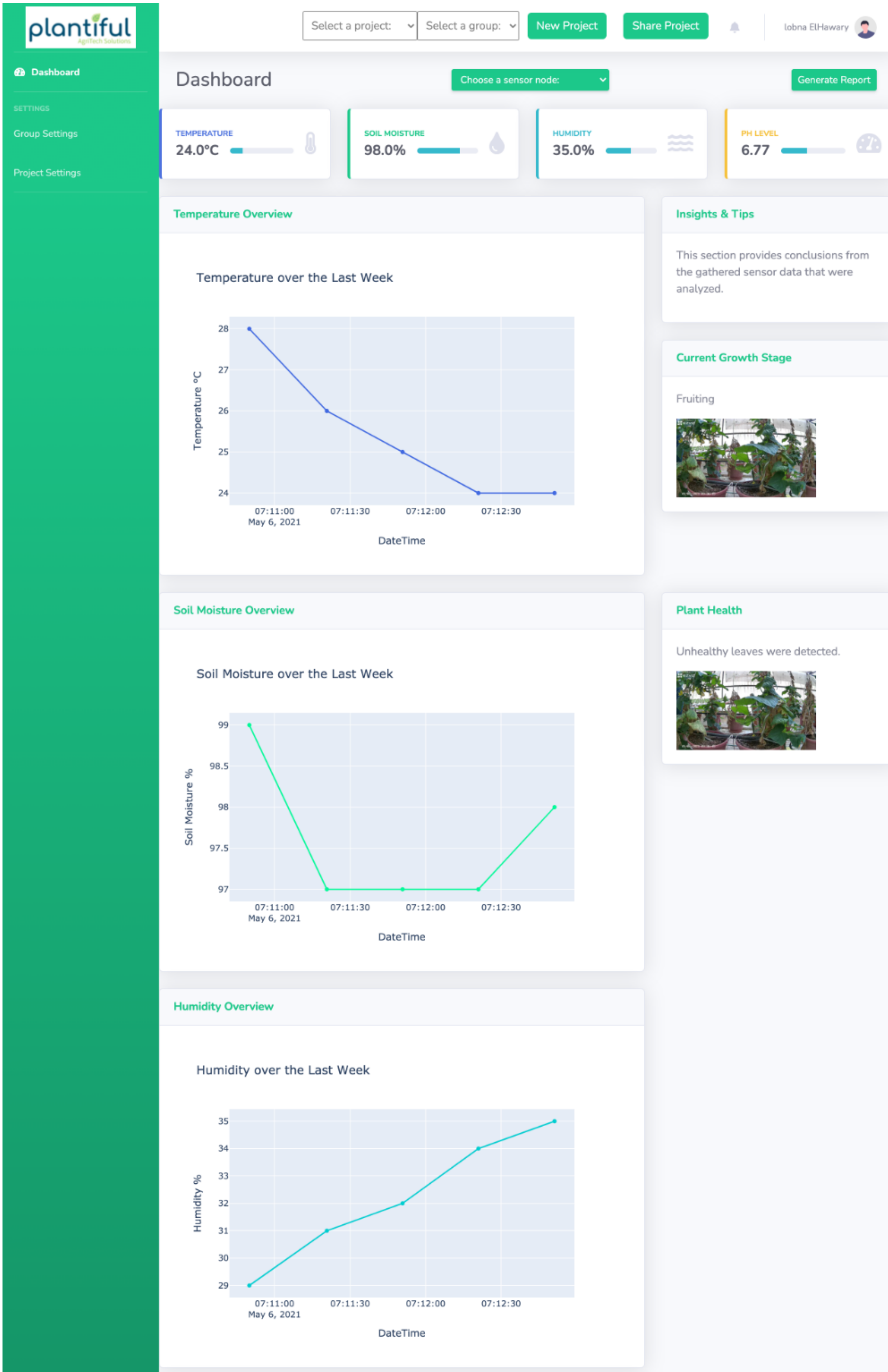
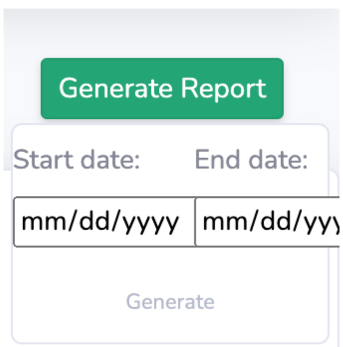
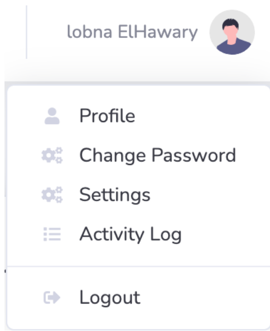

plantiful
Select a project:
Select a group:
New Project
Share Project
lobna ElHawary
Dashboard
SETTINGS
Group Settings
Project Settings
Dashboard
Choose a sensor node:
Generate Report
TEMPERATURE
24.0°C
SOIL MOISTURE
98.0%
HUMIDITY
35.0%
PH LEVEL
6.77
Temperature Overview
Temperature over the Last Week
Temperature °C
DateTime
Insights & Tips
This section provides conclusions from the gathered sensor data that were analyzed.
Current Growth Stage
Fruiting
Soil Moisture Overview
Soil Moisture over the Last Week
Soil Moisture %
Plant Health
Unhealthy leaves were detected.
Humidity Overview
Humidity over the Last Week
Humidity %
07:11:00
May 6, 2021
07:11:30
07:12:00
07:12:30

Figure 6.2.3.3 - Average Sensor Graphs

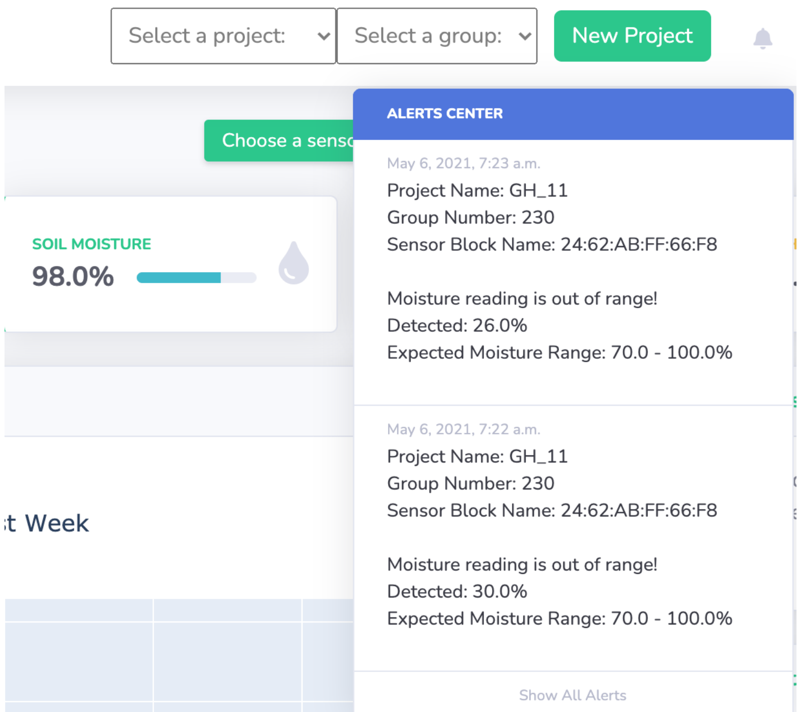
- Generate a csv report containing the collected sensor data taken from a time that the user input till a time that the user selects by pressing the "Generate Report" button.



- Redirect to the selected project's *change project settings page* by pressing "Project Settings"
- Redirect to the selected groups's *change group settings page* by pressing "Group Settings"
- Navigate to the *change password page* by clicking the user's icon and then "Change Password"



- Navigate to the share page
- View the notifications sent to the user by pressing the "bell" icon in the navigation bar.



6.2.4 Creating a Project

Let us say that the user has two groups in a greenhouse that they want to monitor. A group in this case is a set of plants that have the same environment, for example if they get their water and pH from the same sources. Let us say that in group 1 there are two sensors with the names 30:AE:A4:18:BE:8C and 24:62:AB:FF:66:F8 and in group 2 there is one sensor with the sensor name 26:62:AB:FF:46:90 as shown in the figure below. In this section, we will be going through the steps that the user has to take in order to initialize this project on our web application.

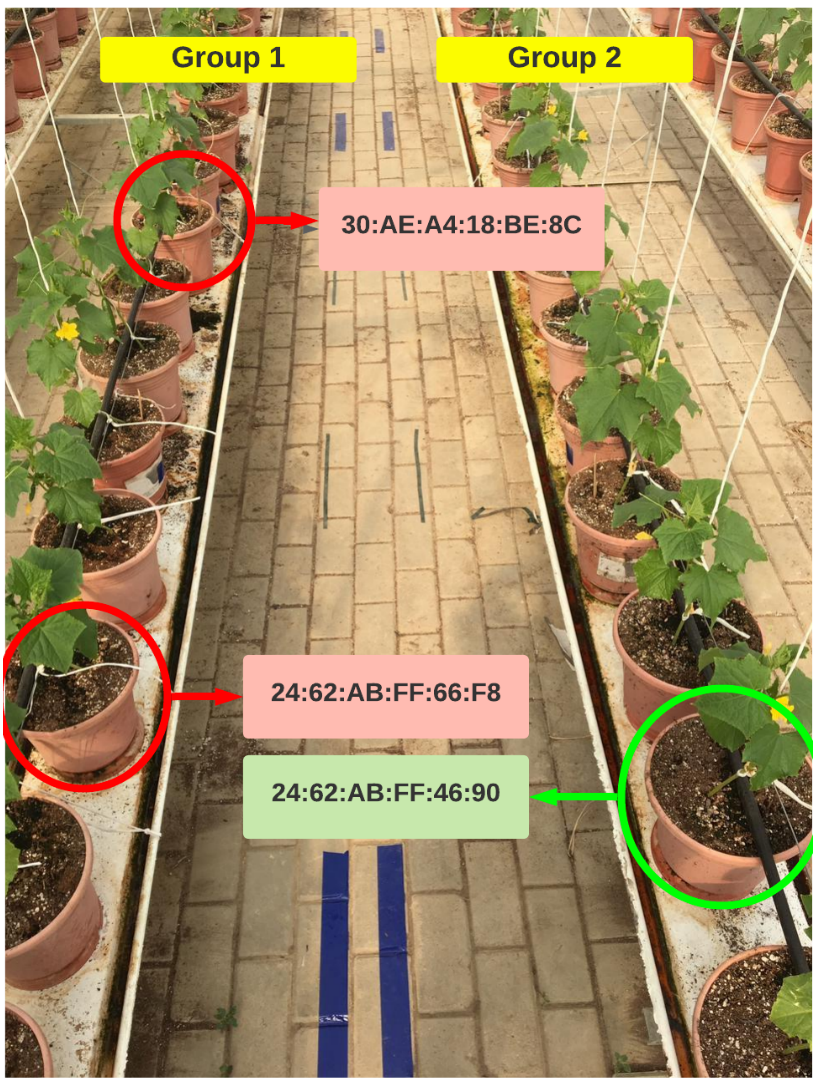


Figure 6.2.4.1 - Real Life Illustration of a Project

6.2.4.1 New Project Settings Page

This is the first page the user has to fill out to create a new project. Here, the user can fill out the project's name and start and end dates along with the number of groups in the project. In this case since the user is mirroring the scenario above, they will fill out that there are 2 groups in the project. After clicking on the "Next" button, the user is then directed to the [new group settings page.]

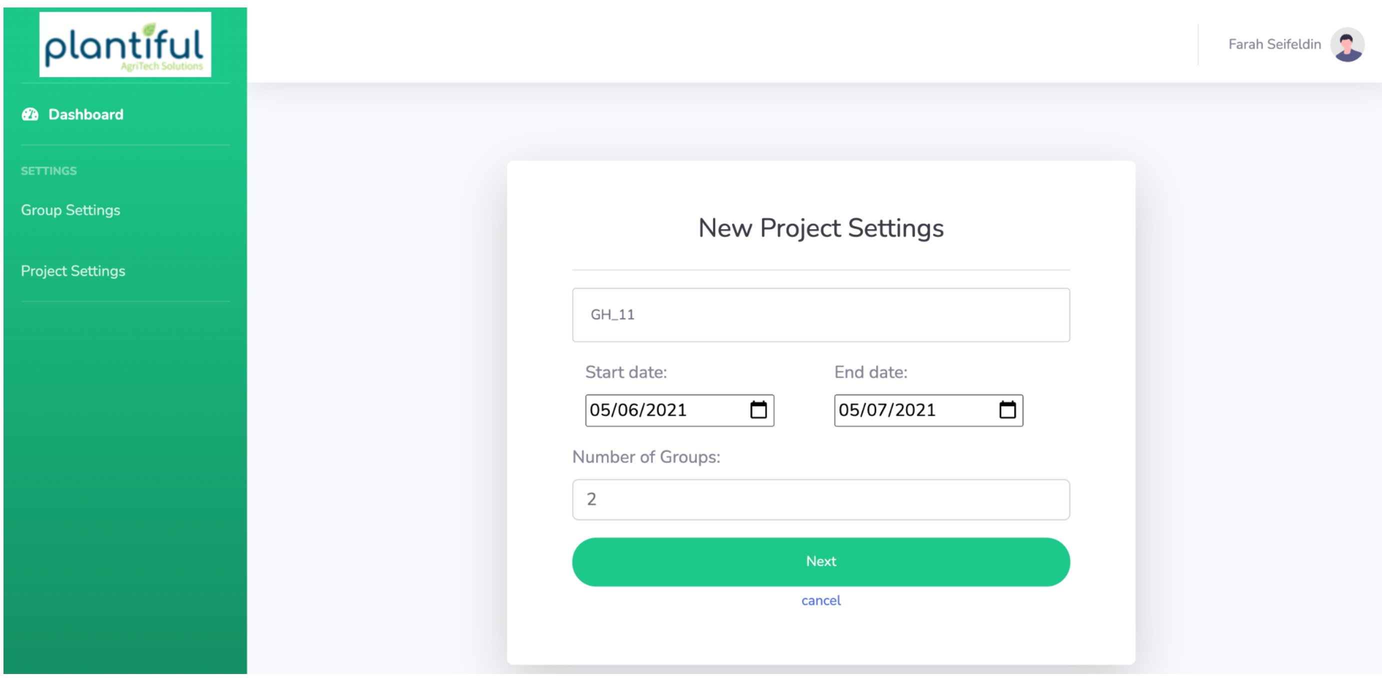


Figure 6.2.4.1.1 - New Project Settings Page

6.2.4.2 New Group Settings Page

Since we selected 2 groups in the project settings, then the user will be prompted to fill the settings of two groups as seen in the figure below.

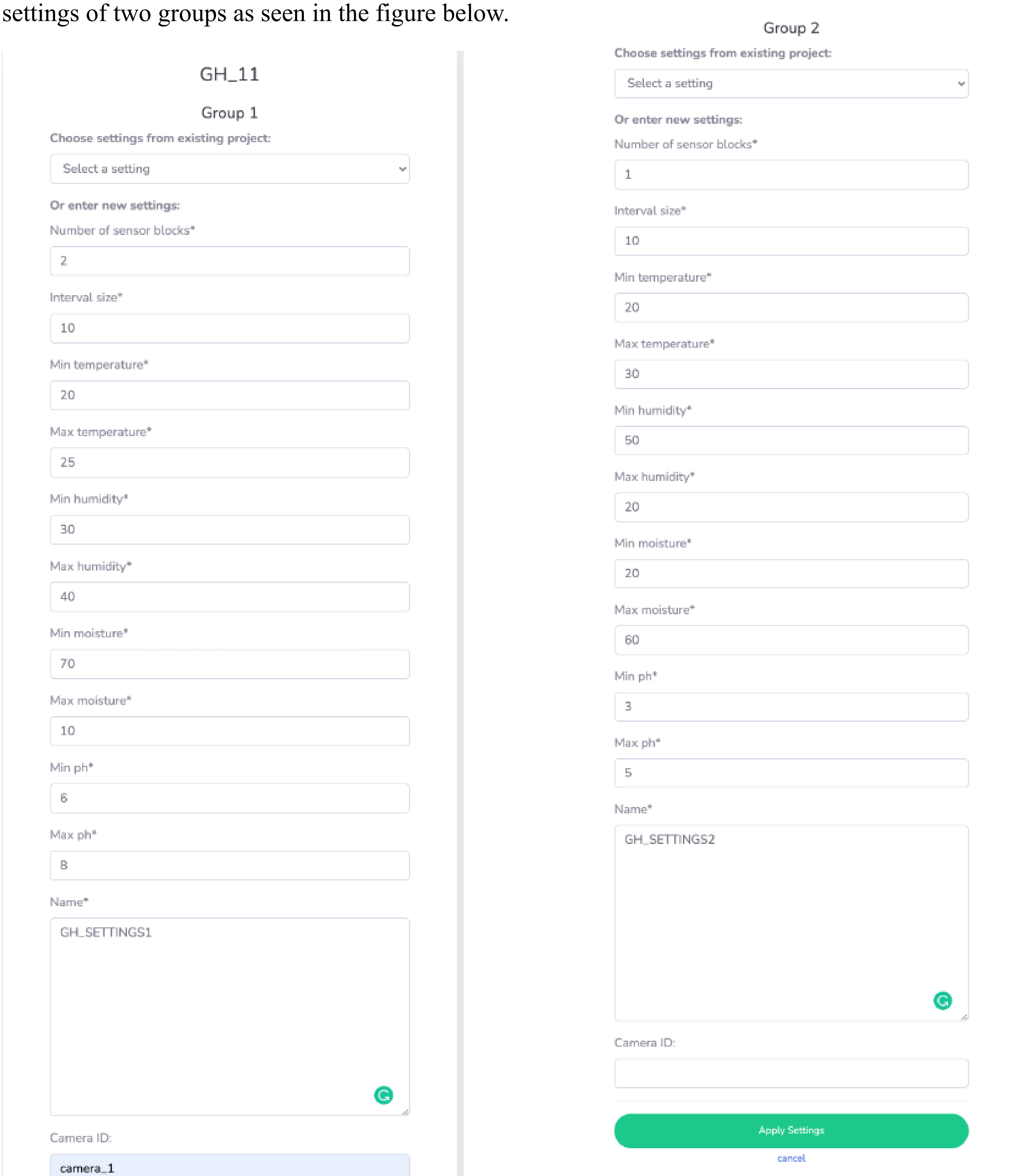


Figure 6.2.4.2.1 - New Group Settings Page (2 Groups)

We selected that group 1 has two sensor blocks and group 2 has one as we saw in the previous figure of the greenhouse (**Figure 6.2.4.1**).

- The number of sensor blocks dictates how many sensor block settings pages will be filled, as will be shown in the section below.
- The interval size is the time (in seconds) between each sensor block reading.
- The minimum and maximum inputs specify the lower and upper limits for each attribute. If a sensor reading is out of bound, then the user is notified.
- The name is the name given to the settings so that next time, the user can just select already made settings from the dropdown list instead of having to input the same settings several times.
- The camera ID is the name of the camera associated with the group, and a group is not required to have a camera functionality which captures images for health and stage detection.

The user can instead of manually entering settings, can choose to pick settings from other groups from the "Choose settings from existing projects" if they match the same settings criteria, for the sake of saving time.

After the user clicks on "Next", they will be taken to fill out the last page in the creation of a new project, and this page is the *sensor blocks settings page*.

6.2.4.3 Sensor Block Settings Page

Since the user previously made two groups where we selected that group 1 has two sensor blocks and group 2 has one, the user will have to fill out the names of the two sensor blocks belonging to group 1 and the one sensor block belonging to group 2 as shown below.

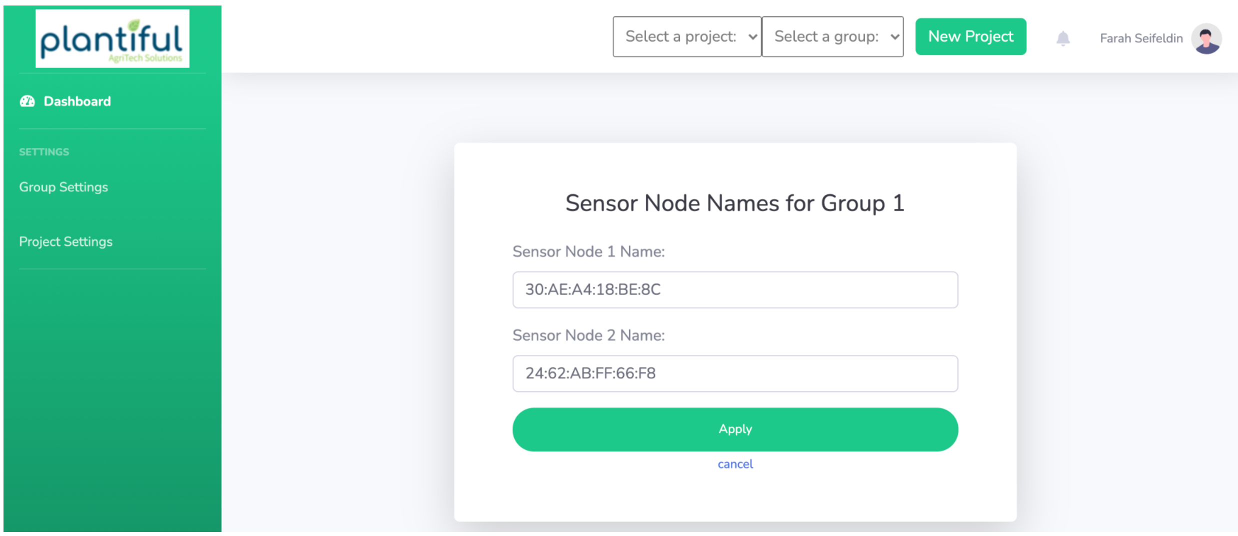


Figure 6.2.4.3.1 - Sensor Node Name Page 1

After the user clicks on "Apply", they will be taken to the second sensor block settings page, given that we have two groups.

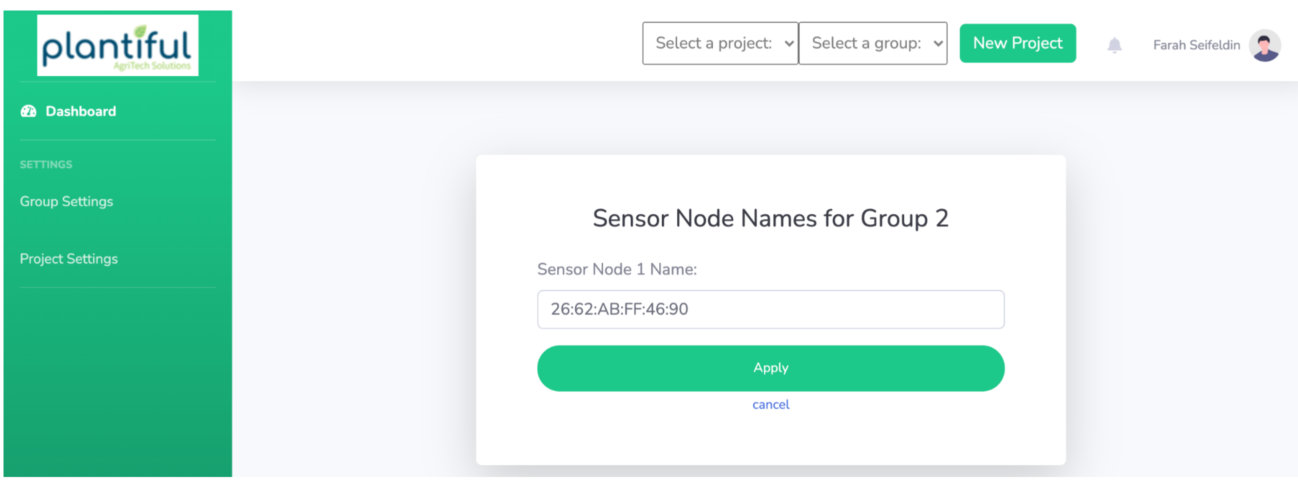


Figure 6.2.4.3.2 - Sensor Node Name Page 2

Now that the user clicks on "Apply" for the last sensor settings page, the user has now created a new project, and the user can view the sensor readings in the homepage as seen in [section 6.2.3](#).

6.2.5 Change Project Settings Page

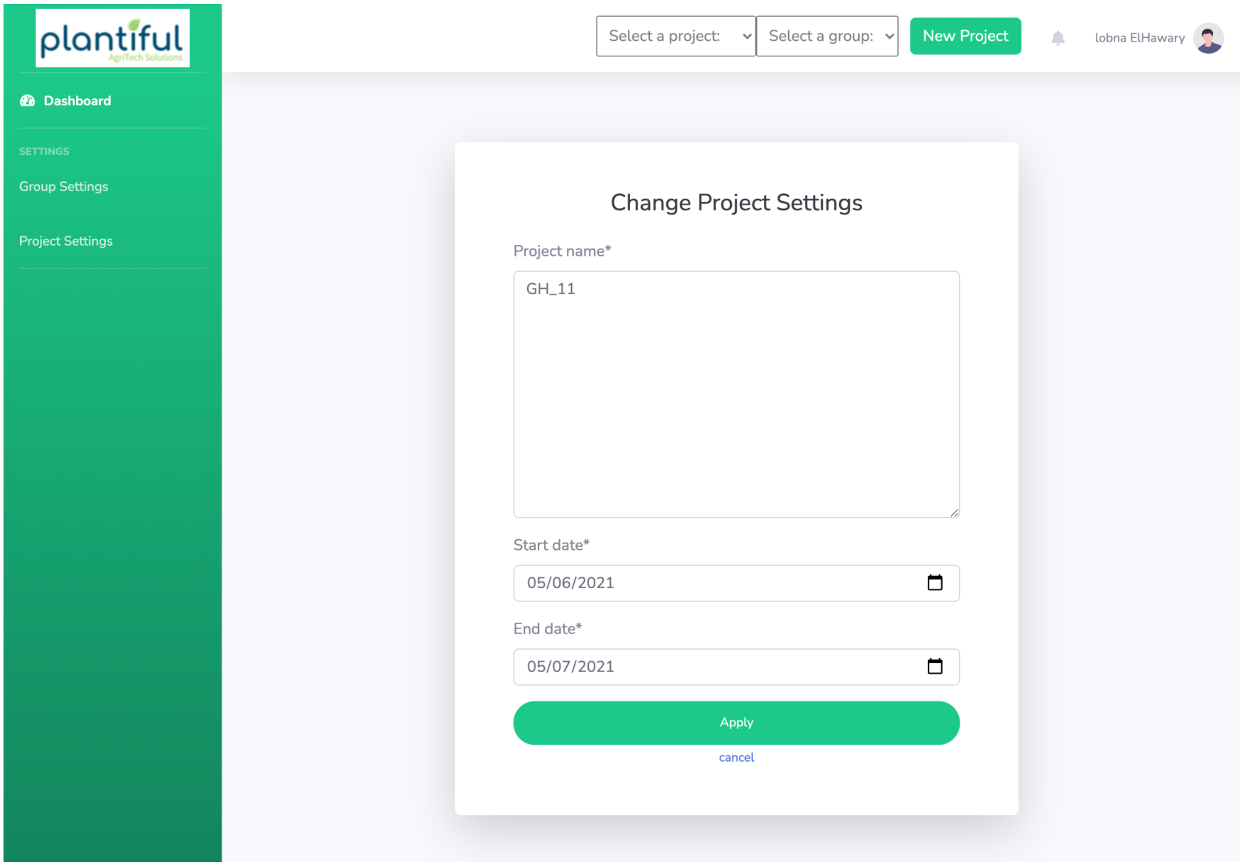


Figure 6.2.5.1 - Change Project Settings Page

In this page, the user can view the previous project settings entered for the project that has been selected from the "Select a project" dropdown. The user can then select any of the inputs and change their value, and then press on the "Apply" for the changes to take place.

6.2.6 Change Group Settings Page

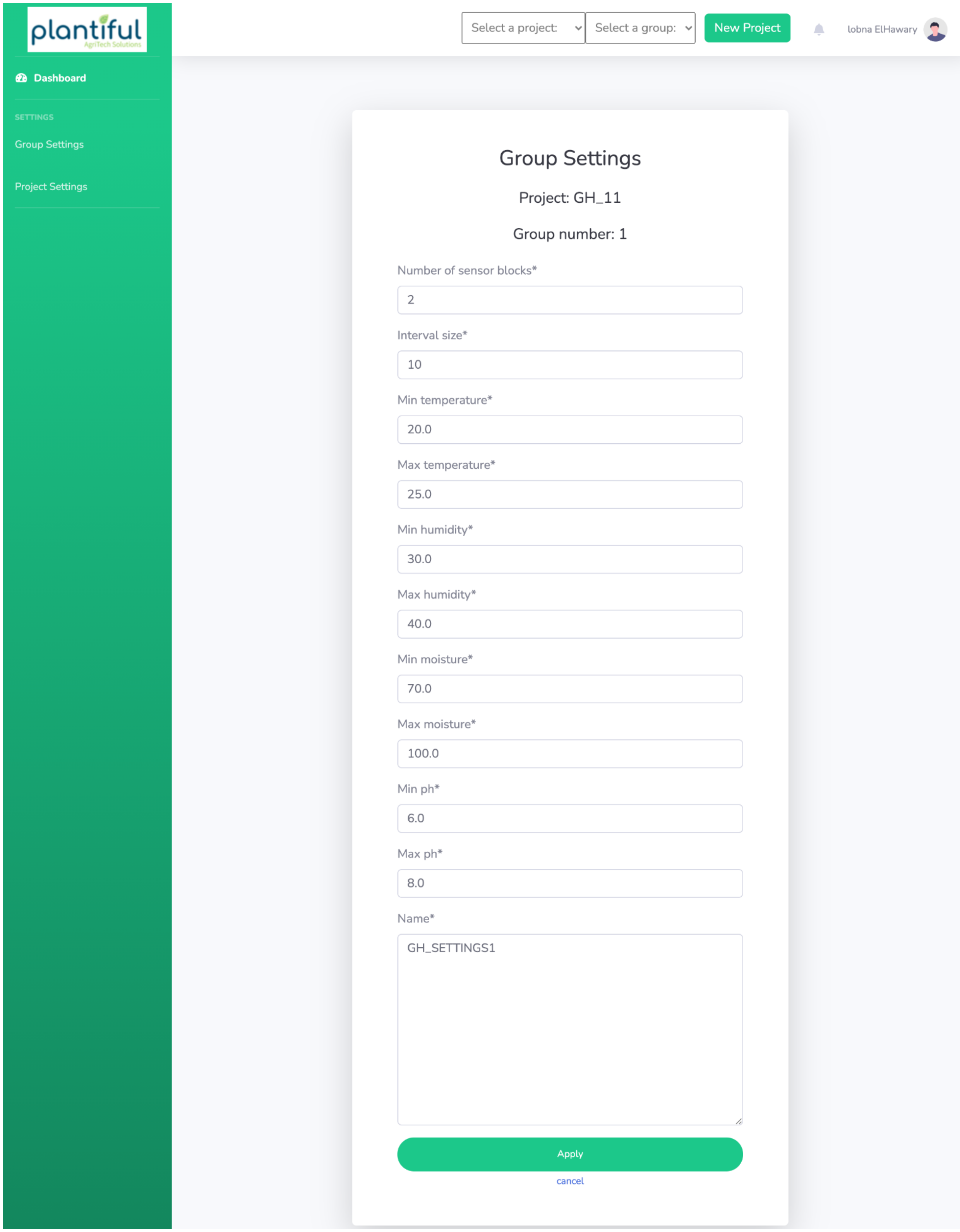


Figure 6.2.6.1 - Change Group Settings Page

In this page, the user can view the previous group settings entered for the group that has been selected from the "Select a group" dropdown. The user can then select any of the inputs and change their value, and then press on the "Apply" for the changes to take place.

6.2.7 Share Project Page

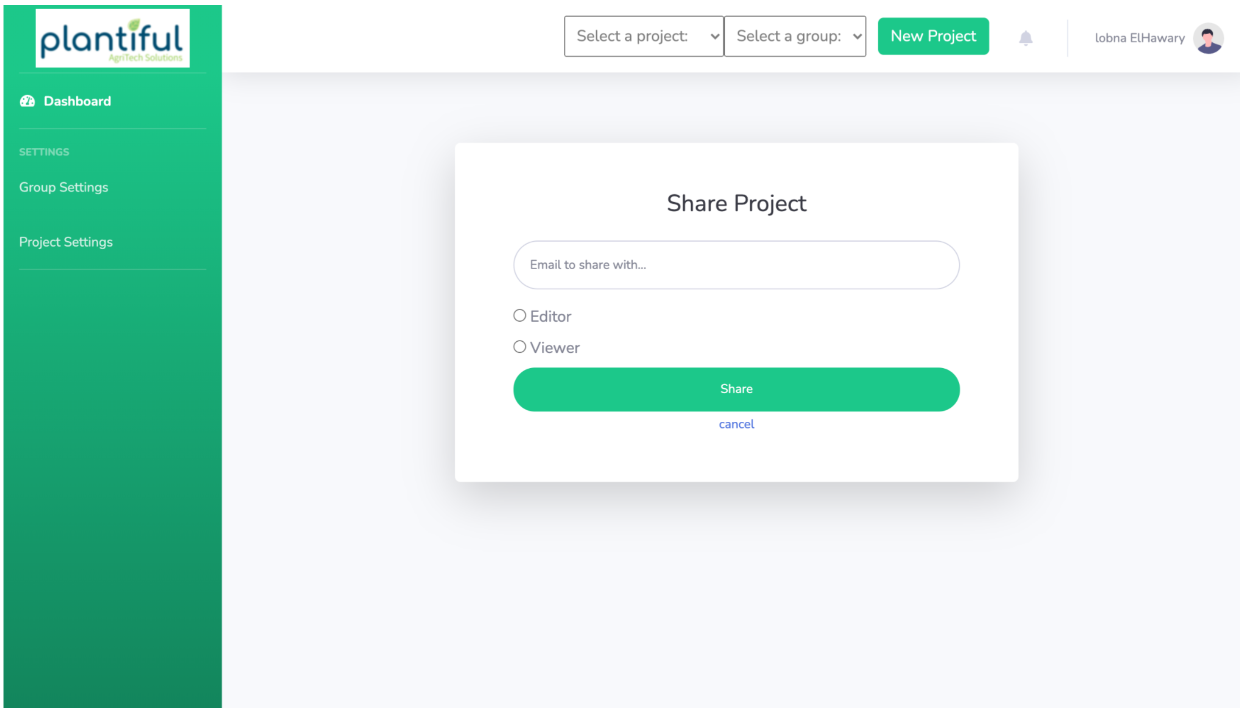


Figure 6.2.7.1 - Share Project Page

In this page, a user can share a project with another user by entering their email address in the input field. So when the user with whom the project was shared with now logs in, the shared project and its corresponding groups will be available for selection which means the user will be able to see the data belonging to the project.

6.2.8 Sent Token

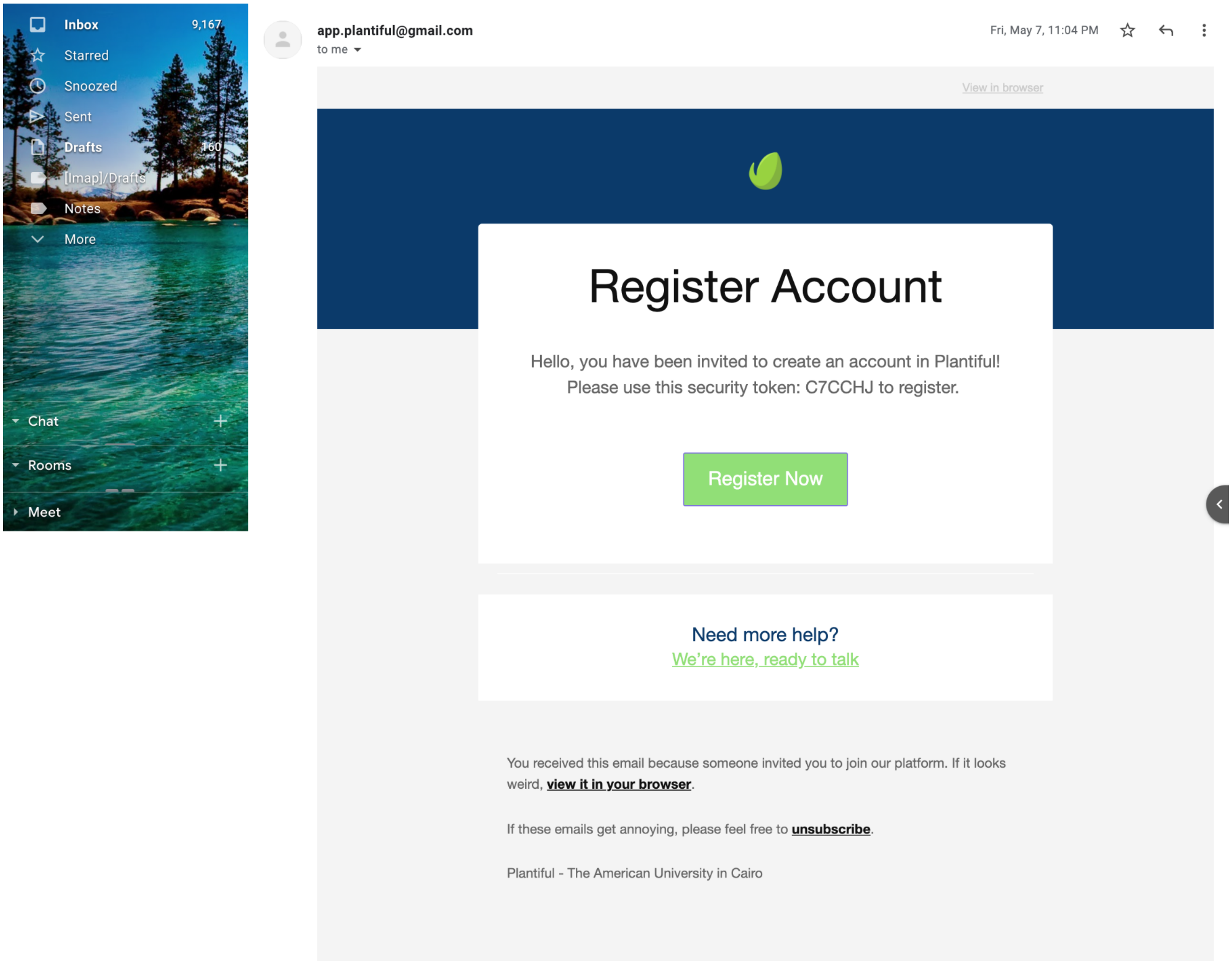


Figure 6.2.8.1 - Sent Token

The user can now press on the "Register Now" button to be redirected to the *registration page*, where they can use this new token to register.

6.2.9 Change Password Page

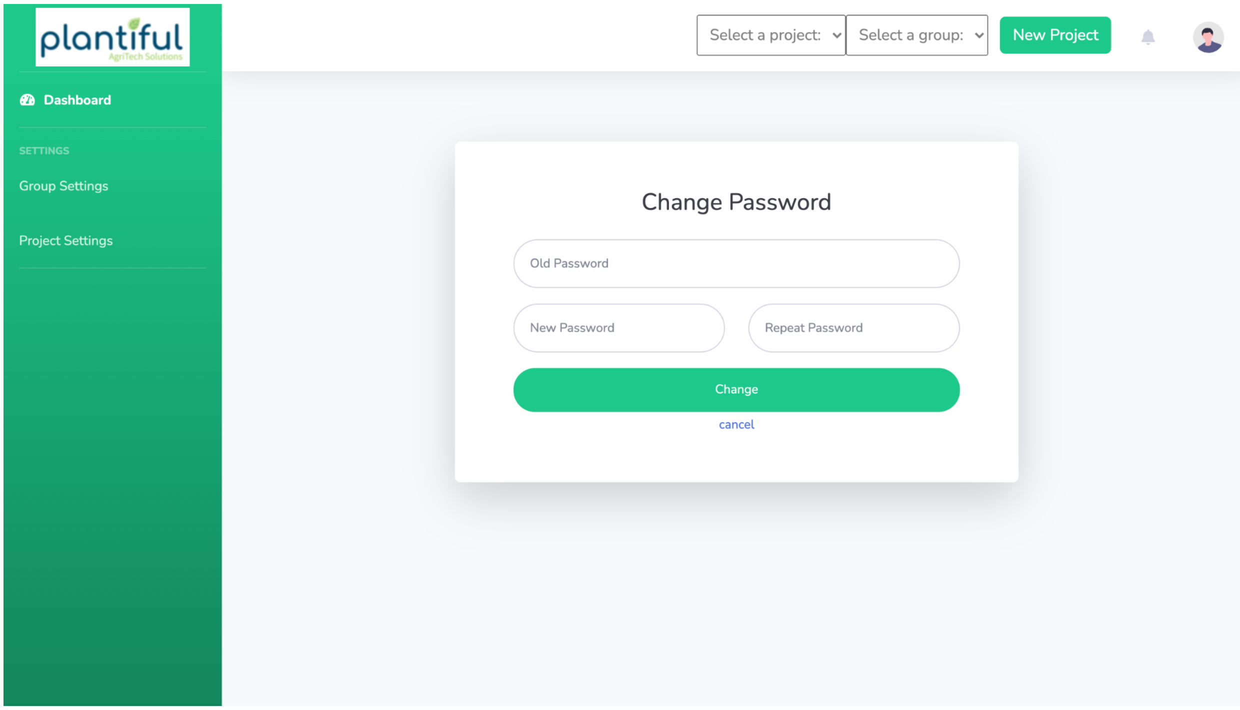


Figure 6.2.9.1 - Change Password Page

This page allows the user to change their password by filling out their old password, and repeating it then entering their new password. If the inputs are correct, and the user clicks on "Change", the password is changed.

7. Requirements Traceability Matrix

Requirement ID	Requirement	Test Cases
1	Login	Login with correct email and incorrect password
		Login with wrong email and incorrect password
		Login with correct email and incorrect password
		Login while leaving email or password blank
2	Register	Register with correct email, password and token
		Register with expired token
		Register with incorrect token
		Register with short password
		Register with empty password
		Register with empty email
		Register with already existing user
		Register with password not same as re-entered password
		Register while leaving any inputs blank
3	Create New Project	Create new project while a required input is blank
4	Create New Group	Create new group while a required input is blank
5	View data	View data when sensor is sending data
		View data when sensor is not sending data
		Verify that data sent from ESP to RPi is correct
		Verify that data sent from RPi to Database is correct
6	Download data	Download data without entering one of requirements
7	Send notification	If sensor reading is outside of range
		If sensor reading is inside of range

8. Glossary of Terms

AI

Artificial Intelligence. The field of computing that simulates human intelligence

CV

Computer Vision. The field of computing that deals with allowing computers to gain understanding from digital images

ML

Machine Learning. The field of computing that deals with automated data analysis.

IoT

Internet of Things. The field of computer engineering that deals with the embedding of real-life objects with sensors

Sensor Block / Sensor Nodes

A block of the sensors of our project: DHT11 (temp and humidity), moisture, and pH

RPi 4

Raspberry Pi 4.0. A microcontroller with an operating system and Wi-Fi module

ESP32

A system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth

Wi-Fi

A network protocol used to communicate over the internet