App Documentation

Team Information

This team name is Connected Tractors, which is represented by the letter I. Connected Tractors consists of Braeden Giasson (*2043670*), Payal Rathod (*2058810*), and Sarim Syed (*1783364*).

Project Description

The project is an IoT product that consists of a container farm which is a stand-alone farming system for growing food inside a shipping container and a mobile app which will be used to view information on the container and control some devices.
The container farm, which uses the reTerminal as a computing device, has different sensors and actuators to be able to function.

All hardware used for this project include the reTerminal and each subsystems specific hardware:

Plant Subsystem Hardware	**Geo-Location Subsystem Hardware**	**Security Subsystem Hardware**
Water Level Sensor	GPS (Air530)	PIR Motion Sensor
Soil Moisture Sensor	USB Power Bank	Magnetic door sensor reed switch
RGB Led Stick	reTerminal’s built-in accelerometer	MG90S 180° Micro Servo
Cooling Fan	reTerminal’s built-in buzzer	Sound Sensor/ Noise Detector

AHT20 Temperature & Humidity Sensor		reTerminal's built-in buzzer

The mobile app contains all the information the user needs as a farm technician or fleet owner. The farm technician can manage all the plants, adjust its environment and monitor the growing conditions. They can also view some readings through graphs to view the sensor readings over time. The fleet owner can view all their deployed containers, monitor its location, assure its proper installation and security, and monitor the containers. They can also view their container's information through a map.
The app and the container farm can communicate and connect using Azure infrastructure, especially an IoT Hub.

D2C Messages

Plant subsystem

```
# Sending temperature

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"temperature\":{\"value\":20,\"unit\":\"C\"\}\}\]\}
```

```
# Sending humidity

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"humidity\":{\"value\":50,\"unit\":\"%HR\"\}\}\]\}
```

```
# Sending water level

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"water-level-sensor\":{\"value\":20,\"unit\":\"%
submerged\"\}\}\]\}
```

```
# Sending soil moisture level

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
```

```
{\"sensors\":[\{\"soil-moisture\":{\"value\":20,\"unit\":""\}\}\]\}
```

Geo-Location subsystem

```
# Sending location

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"location"{\"value\":"",\"unit\":"loc"\}\}\]\}
```

```
# Sending pitch

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"pitch\":{\"value\":10,\"unit\":"°"\}\}\]\}
```

```
# Sending roll angle

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"roll-angle\":{\"value\":10,\"unit\":"°"\}\}\]\}
```

```
# Sending vibration

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"vibration\":{\"value\":20,\"unit\":"m/s2"\}\}\]\}
```

Security subsystem

```
# Sending door state

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"door\":{\"value\":"open",\"unit\":""\}\}\]\}
```

```
# Sending noise level
az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors\":[\{\"noise\":{\"value\":100,\"unit\":""\}\}\]\}
```

```
# Sending motion detection

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
```

```
{\"sensors":[\{\"motion":{\"value":"detected",\"unit":""\}\}\]\}
```

```
# Sending luminosity level

az iot device send-d2c-message -n {iothub_name} -d {device_id} --data
{\"sensors":[\{\"luminosity":{\"value":30,\"unit":""\}\}\]\}
```

Mobile App

App Overview

This mobile app allows a user to login as a farm technician or fleet owner. The farm technician can control the environment by controlling the fan, light and door lock and they can measure data in their container farm by getting the temperature, humidity, soil moisture levels and water levels readings from their container farm. The farm technician can also view all the temperature and humidity readings over the last 24 hours on a graph and in a table.

The fleet owner can view two tabs: geo-location and security. The geo-location tab allows the fleet owner to view all their container farms and its location information (address, pitch, roll angle and vibration levels) and control the buzzer in their container farm. The security tab also allows them to view all their container farms, and its security information (noise level, luminosity level, motion detection and door state) and control the actuators (buzzer and door lock) in their container farm. The fleet owner can also view the location of all their container farms on a map.

There is also a settings page where the farm technician can adjust the thresholds for high and low values for temperature, humidity, water level and telemetry interval. The user can also logout any time from the app.

App Setup

Iot Hub Connection Strings

The connection strings below need to be acquired from your IoT Hub and should be stored in a file called “appsettings.json” in your app project folder. The file should contain the following:

```json
{
    "Settings":
    {
        # Event Hub Connection Strings
        "EventHubConnectionString ": "",
        "EventHubName": "",
        "ConsumerGroup": "",
        "StorageConnectionString": "",
        "BlobContainerName" : "",
        # IoT Hub Connection Strings
        "HubConnectionString": "",
        "DeviceId": "",
    }
}
```

Steps to get **EventHubName**, **ConsumerGroup** and **EventHubConnectionString**:

1. Sign in to azure and navigate to your IoT hub (create if not already).
2. Select “*Built-in endpoints*” from the resource menu, under “*Hub setting*”.
3. Get the **EventHubName** from the section “*Event Hub-compatible name*” under “*Event Hub Details*”.
4. Get the **ConsumerGroup** from the section “*Consumer Groups*” under “*Event Hub Details*” (default name is $Default).
5. Get the **EventHubConnectionString** from the section “*Event Hub compatible endpoint*”.

Steps to get **StorageConnectionString** and **BlobContainerName:**

1. Create a storage account if not done already.
2. Navigate to your storage account.
3. Select "*Access Keys*" from the resource menu, under "*Security + networking*".
4. Get the **StorageConnectionString** from under "*Key1*".
5. Select "*containers*" from the resource menu, under "*Data storage*".
6. Create a container if not done already.
7. Get the **BlobContainerName** which is the name of your container.

Steps to get **HubConnectionString** and **DeviceId:**

1. Sign in to azure and navigate to your IoT hub from step 1
2. Select "*Shared access policies*" from the resource menu, under "*Security Settings*".
3. Select "*service*" as the policy name.
4. Get the **HubConnectionString** from the section "*Primary connection string*".
5. Select "*Devices*" from the resource menu, under "*Device management*".
6. Get the **DeviceId** of the device to be used (create if not already) which is the name of the device.

The connection string below needs to be acquired from your IoT Hub and should be stored in a file called ".env" in the farm folder. The file should contain the following:

```
1    IOTHUB_DEVICE_CONNECTION_STRING = ""
```

Steps to get **IOTHUB_DEVICE_CONNECTION_STRING:**

1. Sign in to azure and navigate to your IoT hub from step 1.
2. Select "*Devices*" from the resource menu, under "*Device management*".
3. Select the device from step 18.
4. Get the **IOTHUB_DEVICE_CONNECTION_STRING** from the section "*Primary connection string*".

Authentication

The authentication for this app used when logging in is being done with firebase. The connection strings used for this are stored in **Config > ResourceStrings.cs**.
The username and password needed to login are already available in the login page for easier accessibility.

App Functionality

Login Page

The login page has two tabs. One is for the fleet owner and the other is for the farm technician. Logging into either one will take you to the respective view with the information that should be shared with that user so that a technician will only see the telemetry of the farm sensors and be able to control the appropriate actuators and the fleet owner will be able to the the container and its security and geolocation information and control of these subsystems. As of right now, an authenticated account has access to both separate logins but in the near future the permissions will be restricted based on which account tries to login.

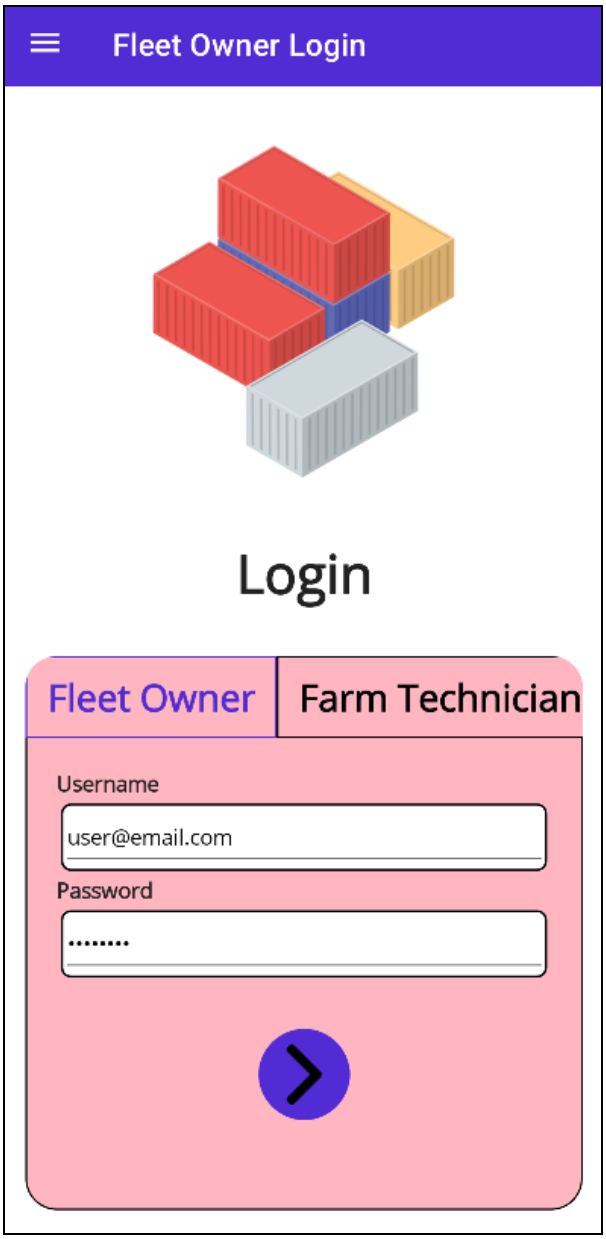

***Figure 1**: Fleet Owner login - The user can enter a username and password, followed by a button to login as a Fleet Owner. The user can also click on the "Farm Technician" title to change to the Farm technician login.*

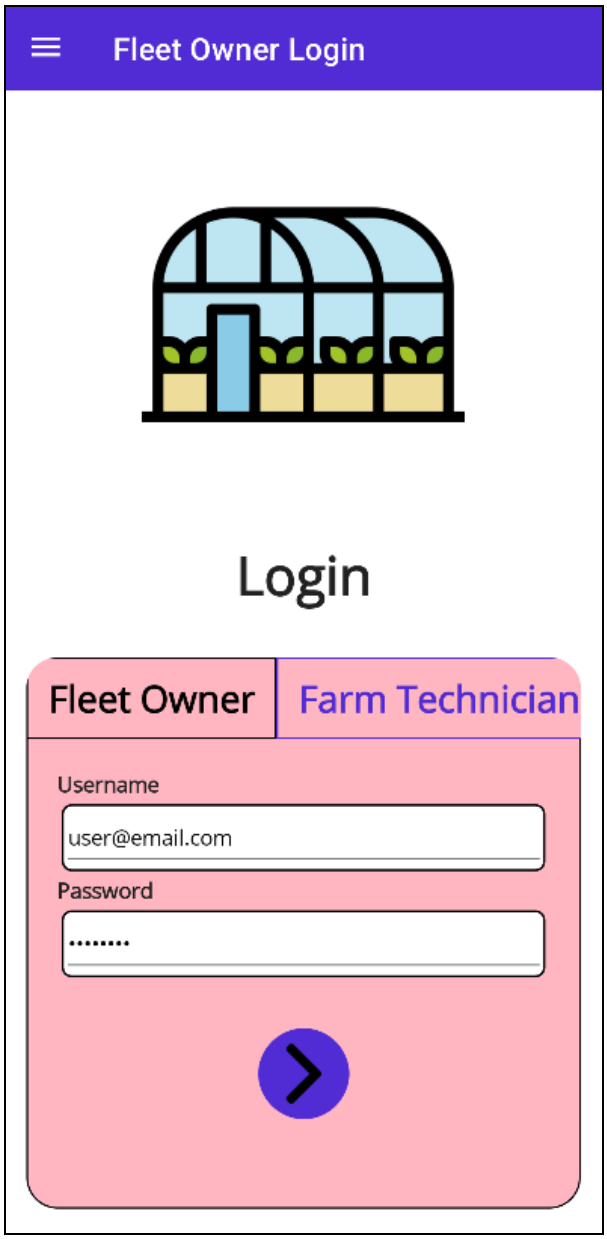

***Figure 2**: Farm Technician login - The user can enter a username and password, followed by a button to login as a Farm Technician. The user can also click on the "Fleet Owner" title to change to the Fleet Owner login.*

Technician Page

The information received from the plant subsystem will be displayed and will be updated as more recent information is received. The user will be able to control the state of the fan, light, and the door lock through this view and the state of the switch also indicates the state of the actuator in the subsystem. If the user toggles the switch and the subsystems for whatever reason are not able to change its state then the switch will revert back into its original state indicating that the change was not successful and the actuator is still on/off. The readings will also change their color depending on how close they are to the threshold specified in the settings page. Settings and logout functionality can be accessed from the flyout menu.

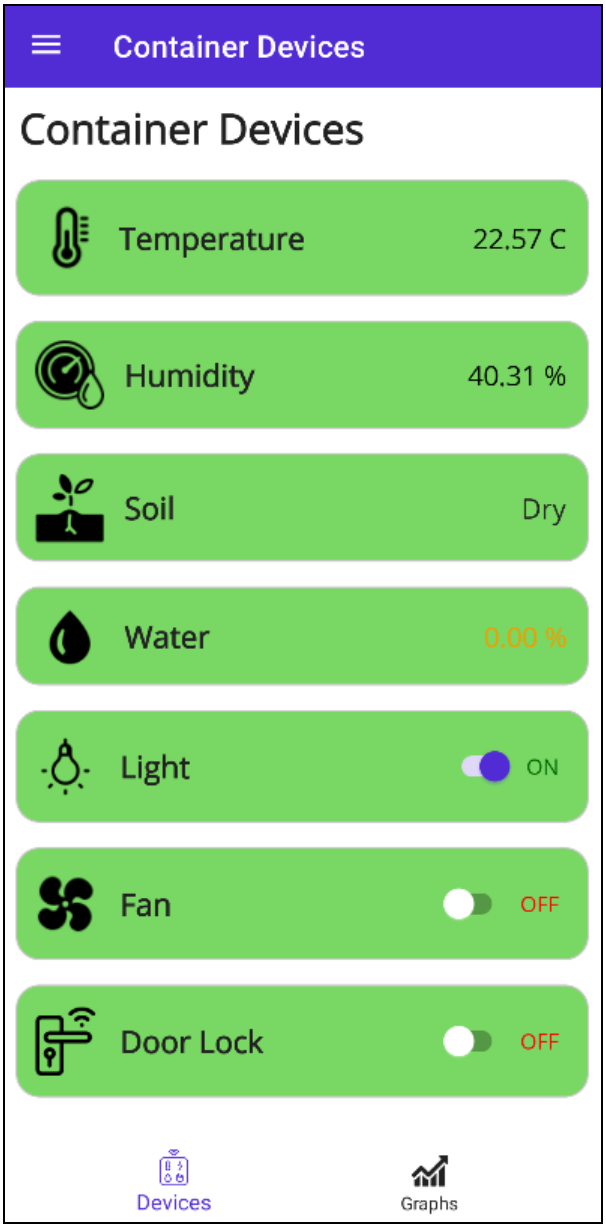

***Figure 3**: Technician Page - The user can see the Temperature, Humidity, Soil, and Water container sensor readings in the first four container devices. The user can also control the state of the light, fan, or door lock, by turning on/off the switch.*

Charts Page

The charts page can be accessed from the Charts tab, and displays a graph and detailed readings for the temperature and humidity values. On the “Temperature” and “Humidity” tabs, the farm technician can see a graph containing temperature/humidity readings over the last 24 hours. They can click on points of the graph to see the specific value at that point. They can also see a scrollable tab with more specific data over the last 24 hours, the temperature/humidity value and the recorded time.

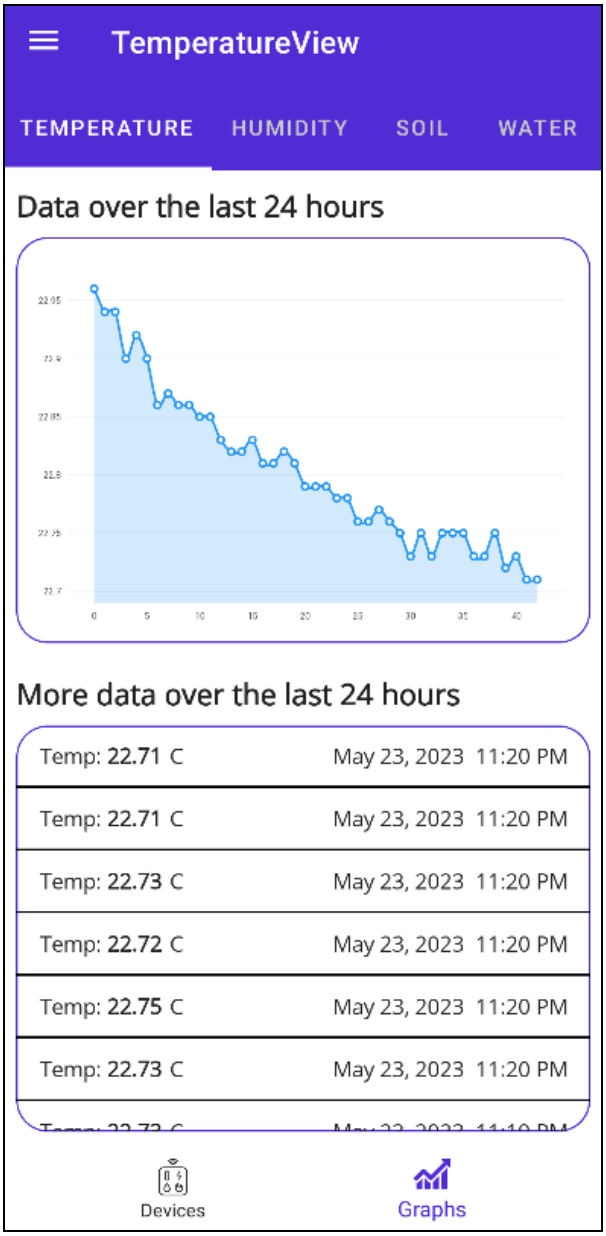

***Figure 4**: Charts temperature page - The farm technician can see a graph containing data temperature readings over the last 24 hours. They can also see a scrollable tab with more specific data over the last 24 hours, the temperature value and what time it was recorded at.*

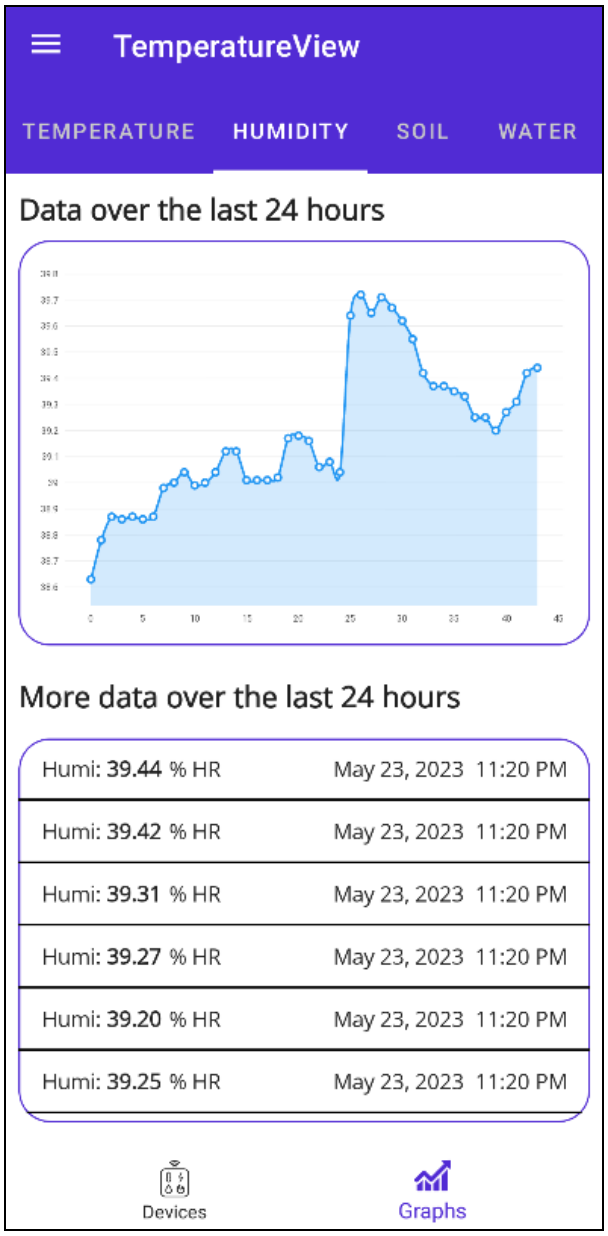

***Figure 5**: Charts humidity page - The farm technician can see a graph containing data humidity readings over the last 24 hours. They can also see a scrollable tab with more specific data over the last 24 hours, the humidity value and what time it was recorded at.*

Fleet Owner Page

The fleet owner page can be accessed by the fleet owner and it displays information about the container(s) being tracked. The geolocation information and security information can be accessed by their respective tabs. The geolocation tab offers information regarding the location of the container; the address retrieved from the gps is displayed as well as other data such as the vibration being detected, the pitch, and angle of the container. A switch can also be toggled to turn a buzzer on or off. By pressing the button.

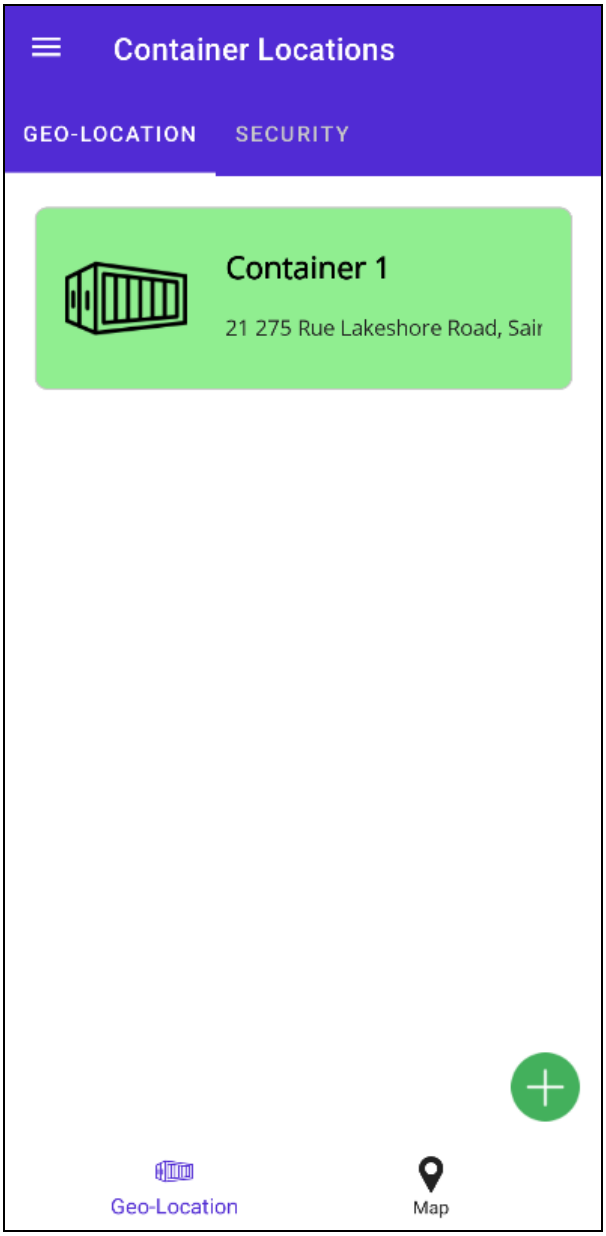

***Figure 6**: Fleet Owner Geo-Location page - The fleet owner can see the list of containers they own, showing the container name and address. They can also see an "add" button to add a new container, which is a feature coming soon!*

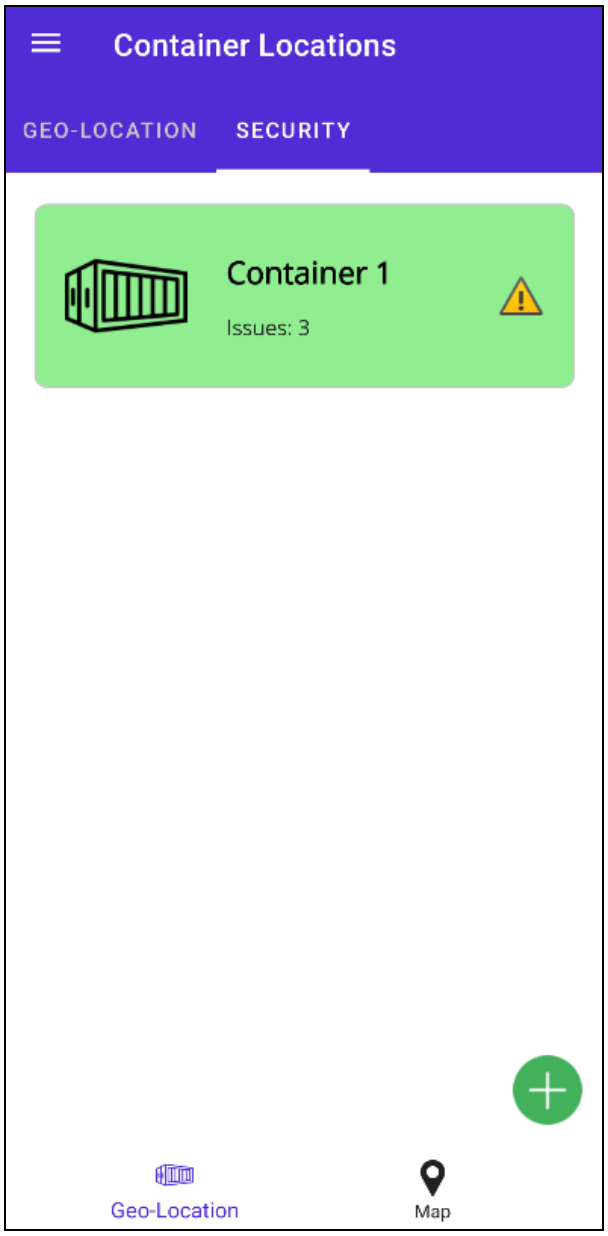

***Figure 7**: Fleet Owner Security page - The fleet owner can see the list of containers they own, showing the container name, number of issues wrong with the container, and a status icon. They can also see an "add" button to add a new container, which is a feature*

coming soon!

Geo-Location Information Page

The Geo-Location information page can be accessed by clicking on a container in the Fleet Owner page, and is used to display information related to the Geo-Location container subsystem. The fleet owner can see two sections, “Sensors”, and “Controllers”.

Under the “Sensors” section, the fleet owner can see the address of the container, the pitch, roll angle, and vibration levels of the container. Under the “Controllers” section, the fleet owner can control the state of the container buzzer by turning on/off the switch.

The fleet owner can also click on the ‘Show on map’ button to navigate to the Maps tab and which will show the precise location of the container on the map.

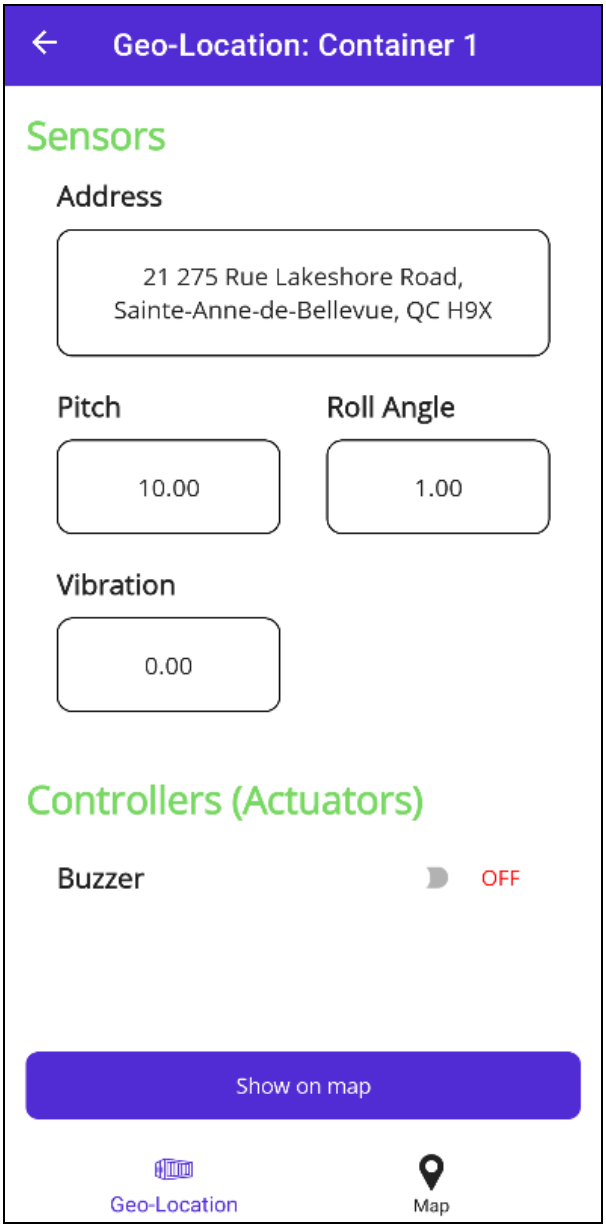

***Figure 8**: Geo-Location information page - The fleet owner can see an interactable map containing pins indicating their container locations.*

Security Information Page

The Security information page can be accessed by clicking on a container in the Fleet Owner page, and is used to display information related to the Security container subsystem. The fleet owner can see two sections, “Sensors”, and “Controllers”.

Under the “Sensors” section, the fleet owner can see the noise, luminosity, and motion levels of the container, along with the door state. Under the “Controllers” section, the fleet owner can control the state of the container buzzer and door lock by turning on/off the switch.

The fleet owner can also click on the ‘Show on map’ button to navigate to the Maps tab and which will show the precise location of the container on the map.

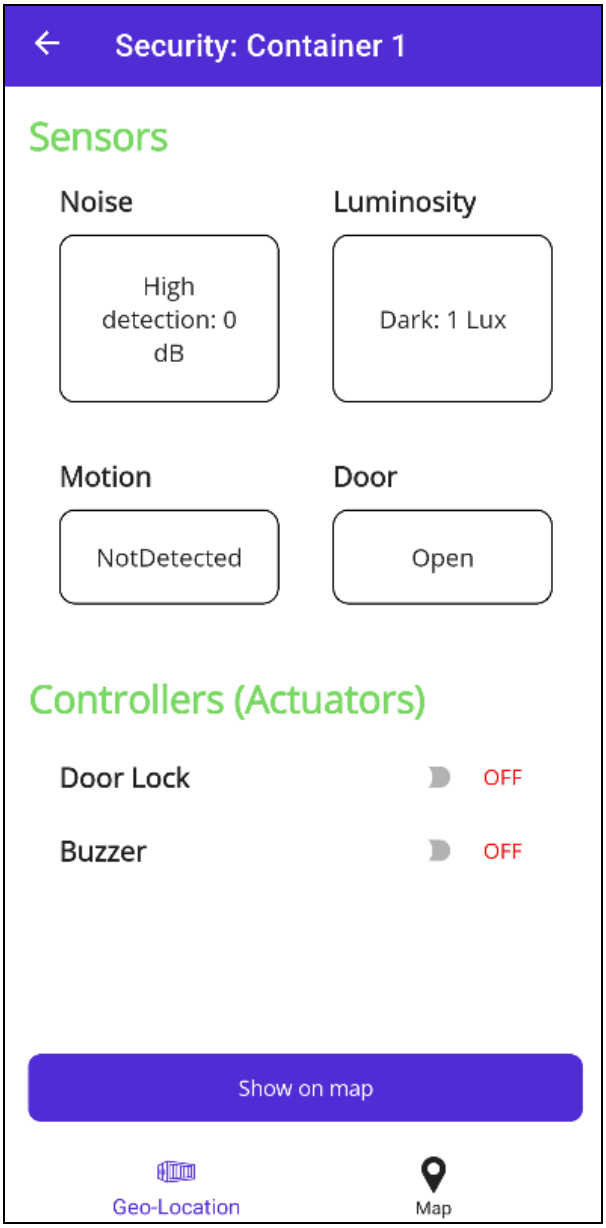

***Figure 9**: Security Information page - The fleet owner can see an interactable map containing pins indicating their container locations.*

Map Page

The Map page can be accessed by clicking the Map tab, and displays a map containing pins indicating all the fleet owner's container locations. The fleet owner can interact with the map to zoom in/out to find specific container locations.

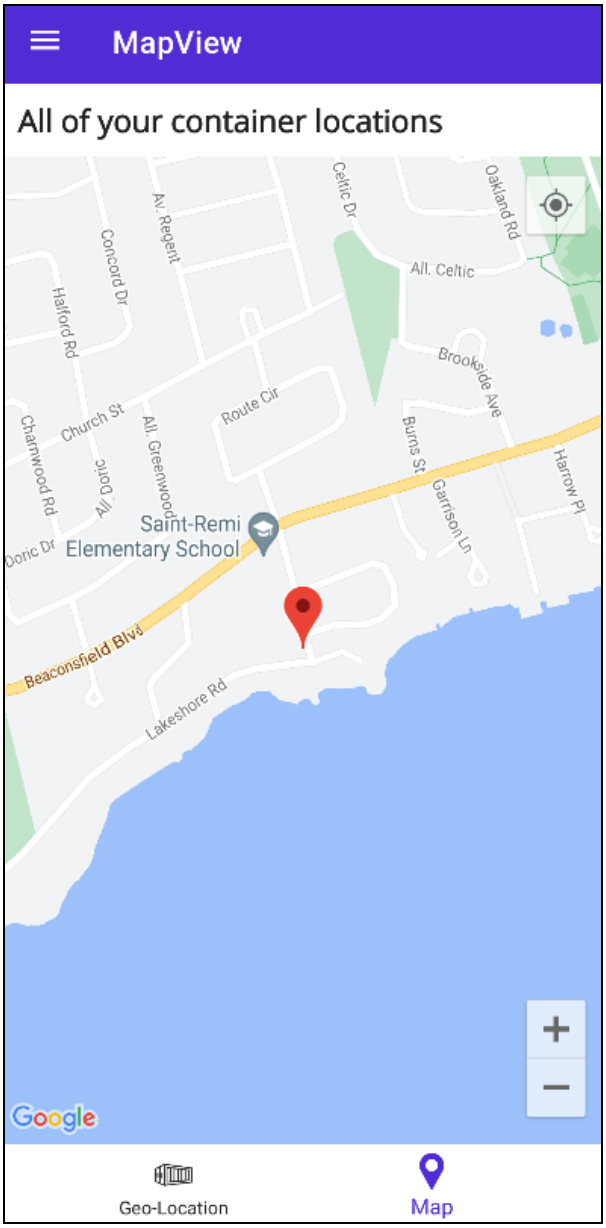

***Figure 10**: Map page - The fleet owner can see an interactable map containing pins indicating their container locations.*

Settings Page

The settings page can be accessed by the flyout menu, and is used to change the container devices thresholds and telemetry interval. The user can change the temperature, humidity, and water levels high and low thresholds. These thresholds are

used to change the text color of the matching container devices. The user can also change the telemetry interval which will change how many seconds it takes before new messages are received to the app.

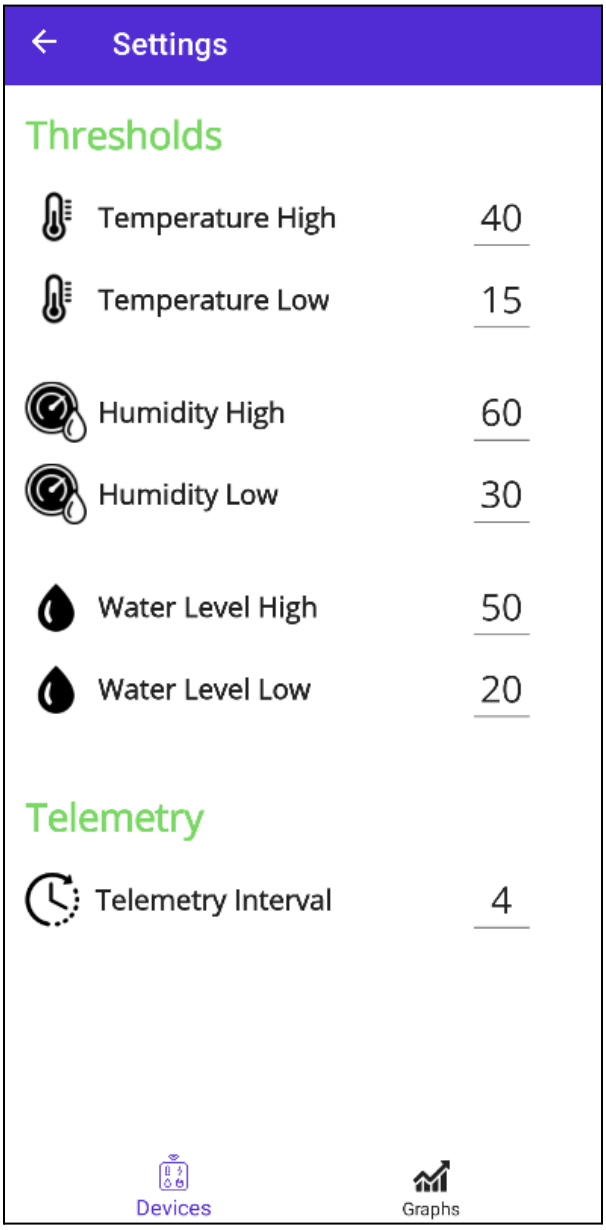

***Figure 11**: Settings page - The user can change the container devices thresholds and the message telemetry interval.*

Flyout Menu

The fleet owner page or farm technician can click on the hamburger menu (the three lined icon on the top left of the app) to access the flyout menu containing two options. One, “Logout”, so the fleet owner page or farm technician can log out of the app, and two, “Settings”, so the fleet owner page or farm technician can navigate to the settings page.

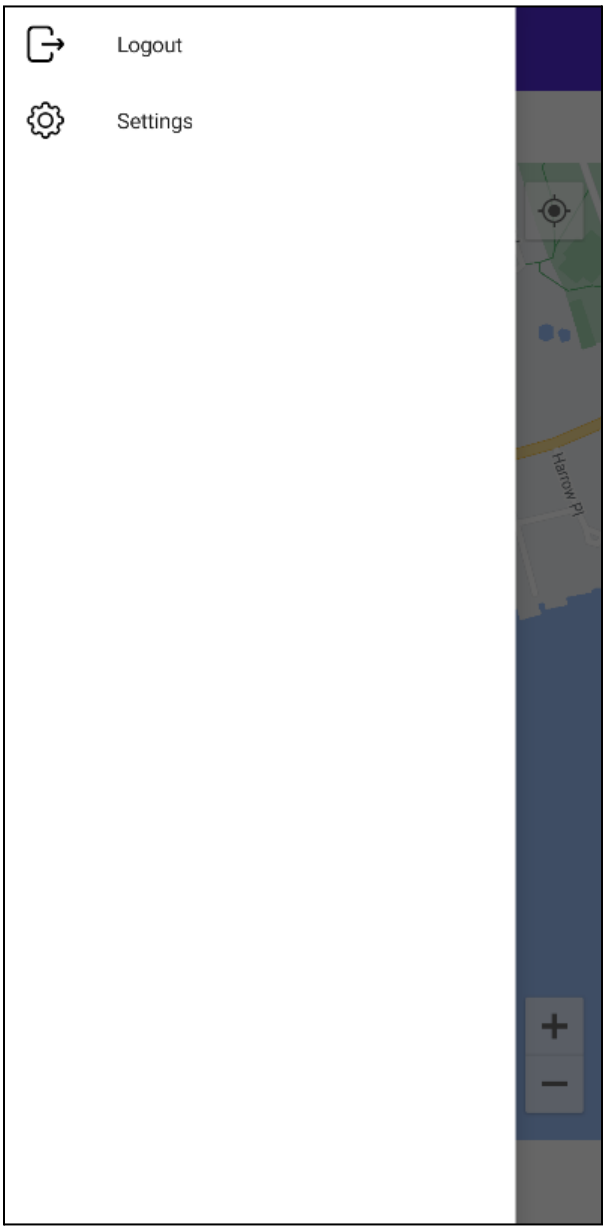

***Figure 12**: Flyout Menu - The fleet owner or farm*

technician can see two menu options, “Logout” to logout of the app, and “Settings” to navigate to the Settings Page.

Future Work

Right now the same account signs in and has access to both fleet owner and technician functionality and we plan on implementing a way to allow access depending on the permission assigned to the user. The ability to register an account in the app is also planned. The UUID and permission level will be stored in a database such as a Firebase DB and when login authentication with firebase is successful the users permission level will be checked using the database.

There is only one container being tracked and in the future the ability to add extra containers is also planned so that a fleet owner can track more than one container. The fleet owner will be able to click the floating action button in the fleet owner view which will take them to a page allowing them to enter the information that will be used to identify and connect to additional containers.

The colors palette is also something that needs to be adjusted as while the colors associated with the threshold and the container and sensor list items are nice, they are not colorblind friendly. Some research is required to ensure the app is as accessible as possible. The ability to allow the user to change the colors they want to see, or have multiple themes accommodating color blindness are planned for the future.

Charts and graph tracking data over the past 24 hours are only available for the temperature readings and humidity readings. This feature is planned to be expanded to include the other sensor’s historical data.

When the device collecting the sensor data loses connection to the IoTHub, or the internet the data gathered until the device reconnects is lost. In the future the plan is to

have a buffer collecting the data and sending it once the device reconnects so that there is no gap in the data.

The App does not notify the user if the data has not been updated for an extended period. In the future the app will have an indicator in the form of an icon in the navbar that lets the user know when the data is dated. This icon can be clicked to let the user know when the last update was so they can try to troubleshoot.

The TwinActuatorService class manipulates the repo which violates the MVVC design pattern. Reformatting that class and moving those functionalities to the appropriate classes is planned for the future.

The flyout menu is still visible on the login screen and is planned in the future to be hidden until the user is logged in. The Settings page will also be different for both different types of logins so that they can set the thresholds of the systems they are tracking.

Contributions

Braeden	Payal	Sarim
App frontend, backend, subsystem backend (Geolocation).	App frontend, backend, subsystem backend (Security).	App frontend, backend, subsystem backend (Plant).

The work was essentially equally divided. At the start of the project we had divided by saying Braeden would do the frontend, Payal would do the App backend, and Sarim would work on the Python script handling the subsystem.

In the end, everyone contributed to every part of the project. We worked on our respective subsystems in the python script, and when it came to the app we

implemented our subsystems in the app, the necessary views and the C# code associated with them.

Certain things such as the connection from the App to the IoT Hub were handled by Braeden and Payal, and sending the readings in the form of D2C messages from the Raspberry Pi was handled by Sarim.

In short, the division of work was quite equally distributed and everyone relatively helped and worked on a lot of the same tasks. We were in constant contact and took little to no time helping the others if they needed help.