

# Documentation Document - IoT Farm

Team #12 (BSV)



May 20, 2022

## Team Information

Name: **BSV**

Number: **#12**

Brookelyn Palfy - **1774601**

Spencer Kifell - **1937291**

Vathusan Vimalarajan - **1975607**

## Table of Contents

<b>Team Information</b>	<b>1</b>
<b>Project Description</b>	<b>3</b>
<b>IoT System</b>	<b>4</b>
Sensors	4
Plant Sensors	4
Geo-Location Sensors	4
Security Sensors	4
Cloud-to-Device	5
Contributions	6
<b>Mobile App</b>	<b>7</b>
App Purpose	7
App Functionality	7
App Snapshots	8
Future Work	11
Bonus Work	12
Authentication	12
Object Oriented Programming Design	14
Contributions	15
<b>Dummy Telemetry Data</b>	<b>16</b>

# Project Description

- The purpose of this project was to create an IoT solution for a fleet of container farms. A container farm is a shipping container greenhouse. Since these environments aren't necessarily natural, they require a lot more maintenance. This being said, we used a reTerminal (Raspberry Pi) with peripherals to continuously monitor any container farm's plants, geo-location and security data.
- The entire point of this process is that the container farm will be fully automated. That being said, we utilized Azure to store telemetry data and Azure device twins to store and control the state of actuators connected to the reTerminal.
- The hardware for each subsystem is read in Python and combined into a station to send one Telemetry payload to Azure for storage.
- A fleet manager or a farm technician can monitor any environmental changes in a cross platform mobile application developed using Xamarin Forms. The application gives the user access to specific functionalities based on their designated role when being registered. On the application, the user has the capabilities to control and monitor the status of many container farms.

# IoT System

## Sensors

### Plant Sensors

<b>Sensor</b>	<b>Base Hat port</b>
Temperature and humidity sensor	D26
Water level sensor	A4
Moisture sensor	A0
LEDs	D16
Fan/Relay	D5

### Geo-Location Sensors

<b>Sensor</b>	<b>Base Hat port</b>
GPS	UART

### Security Sensors

<b>Sensor</b>	<b>Base Hat port</b>
Noise sensor	A2
Motion Sensor	D24
Door Lock	PWM
Door Sensor	D22

## Cloud-to-Device

- Communication method chosen for controlling actuators is by using the **Device Twin**.
- We chose this method because we felt it to be the most simple choice for communicating the state of devices. With this choice, the app can set the desired properties and the station will receive the desired property, make the necessary changes and update the reported properties. The app can then reflect the reported properties.
- c.

Keys	Possible Values
"buzzerState"	"on" , "off"
"lightState"	"on" , "off"
"fanState"	"on" , "off"
"doorLockState"	"open" , "close"

- d.

Keys	Possible Values
"telemetryInterval"	Any integer value

- e. Example:

The device is controlled by setting the desired properties of the listed keys with the possible values listed above. In the example below, setting the value for "lightState" to "off" will turn off the LEDs (the reported properties will reflect (confirm) the changes have been made).

Desired Properties	Reported Properties
--------------------	---------------------

```

22     },
23     "properties": {
24       "desired": {
25         "power_level": 4,
26         "telemetryInterval": 5,
27         "buzzerState": "off",
28         "lightState": "off",
29         "fanState": "on",
30         "doorLockState": "closed",

```

```

"reported": {
  "serialNumber": "alohomora",
  "power_level": 4,
  "version": 32,
  "telemetryInterval": 5,
  "buzzerState": "off",
  "lightState": "off",
  "fanState": "on",
  "doorLockState": "closed",

```

## Contributions

Milestone	Brookelyn	Spencer	Vathusan
1	-User stories -App design document -Team contract	-Wireframe for UI -App design document -Team contract	-App design document -Team contract
2	-Plant subsystem hardware PR	-Geo-location subsystem hardware PR	-Security subsystem hardware PR
3	-Plant subsystem Telemetry and Desired Properties PR	-Geo-location subsystem Telemetry and Desired Properties PR	-Security subsystem Telemetry and Desired Properties PR
4	-Online Document -Controlling Actuator Devices -Device twin reported properties -Combined station	-Combined station -Updated telemetry	
5	-Functional Demonstration -Updated documentation	-Functional Demonstration - Updated documentation	-Functional Demonstration

# Mobile App

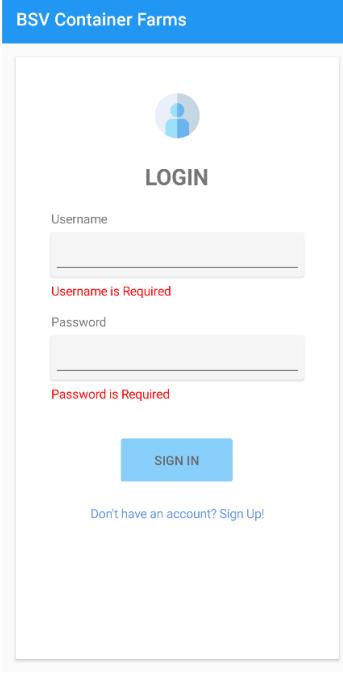
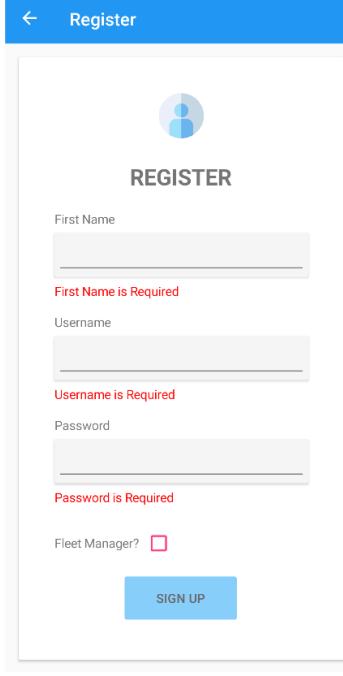
## App Purpose

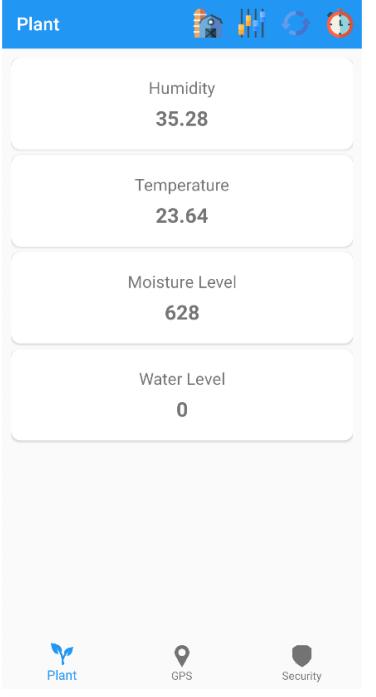
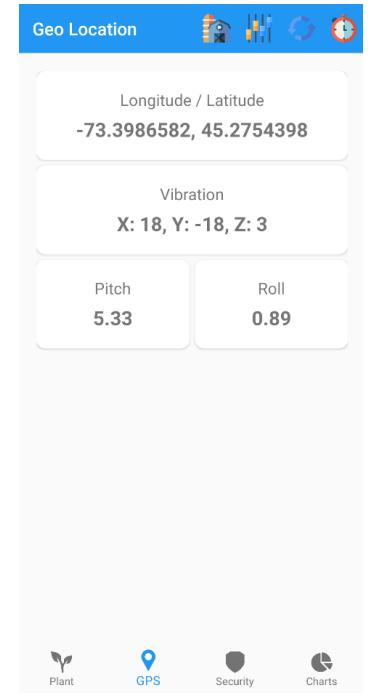
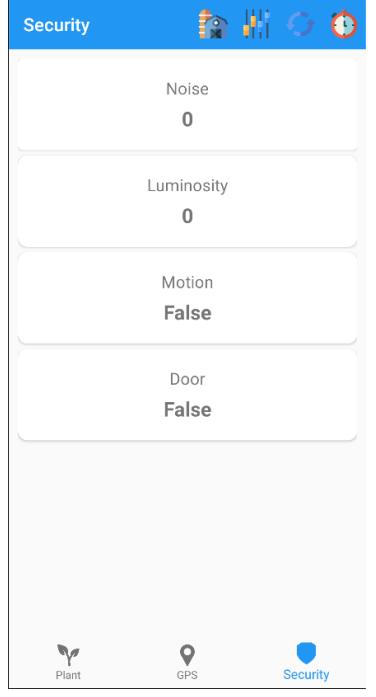
- The purpose of this application is to communicate with farm container devices and display collected data to the user. The user can also control certain actuators with the application. The purpose of this is to create an application that farmers and managers of a container farm can use to monitor and update all necessary information/states of the container farm.

## App Functionality

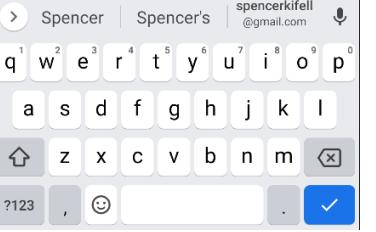
- **Sign up:** The application allows you to create an account with the option to be fleet manager or farm technician which will determine your access to views and controls.
- **Login:** After you have signed up, you are able to login to the application with your selected username and password which will be authenticated.
- **Manage Container farms:** Fleet Managers can then add, edit and delete container farms. Select the container farm to view additional data and controls. Farm technicians only have access to container farms when they have been added to one by a fleet manager.
- **Add Users to Container Farms:** Fleet managers can add farm technicians and other fleet managers to container farms they created. All your container farms will be displayed in the container farm view where you can select the container farm that you want to work with.
- **View Measurements:** Once you have selected a container farm, you can view data collected by the plant, security and geo-location sensors by using the bottom navigation to visit each tabbed page. To get this data, you must refresh the page using the refresh button in the top menu bar. Refreshing the page gets the data from azure and displays it to the user.
- **Control Actuators :** You now also have the option to control plant, security and geo-location actuators from the application by clicking the controls icon in the top menu bar from each subsystems page.
- **View Charts Page:** In the bottom navigation you also have access to a charts page which allows you to select a container farm property and the number of data points you want displayed and it will generate a graph for the user to see how the property has behaved over time.
- **Modify Telemetry Interval:** The user can modify the telemetry interval by clicking the clock icon in the top menu bar and saving a desired interval.
- **Map View:** Lastly, the user can long press the longitude latitude measurement to access a map with the current location of the container farm.

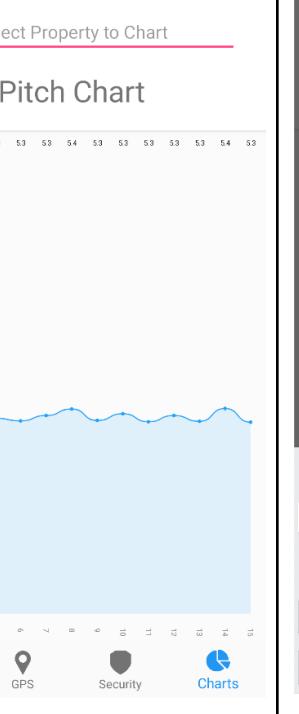
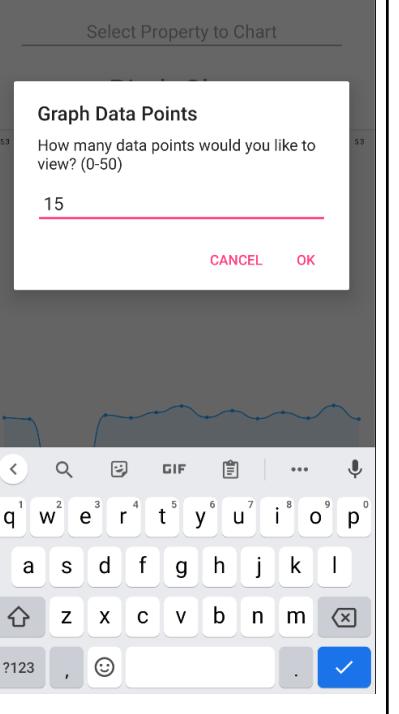
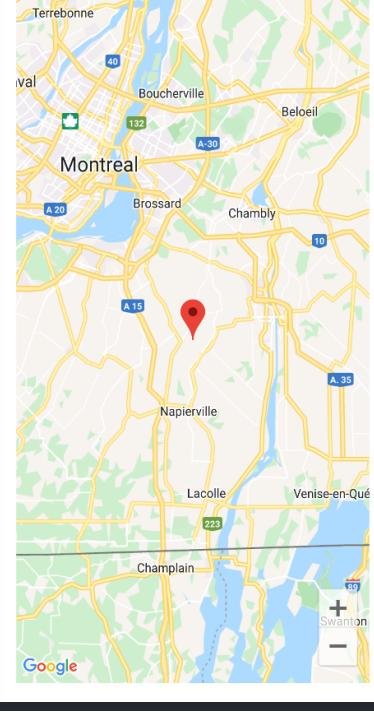
## App Snapshots

 <p><b>BSV Container Farms</b></p> <p><b>LOGIN</b></p> <p>Username <input type="text"/></p> <p>Username is Required</p> <p>Password <input type="password"/></p> <p>Password is Required</p> <p><b>SIGN IN</b></p> <p>Don't have an account? <a href="#">Sign Up!</a></p>	 <p><b>REGISTER</b></p> <p>First Name <input type="text"/></p> <p>First Name is Required</p> <p>Username <input type="text"/></p> <p>Username is Required</p> <p>Password <input type="password"/></p> <p>Password is Required</p> <p>Fleet Manager? <input type="checkbox"/></p> <p><b>SIGN UP</b></p>	 <p><b>Container Farms</b></p> <p><b>Brookelyn Farm</b> Brookelyn's Farm at home</p> <p><b>Spencer's Farm</b> Spencer's farm that never works.</p> <p><b>+</b></p>
<p><b>Login Page</b></p>	<p><b>Sign Up</b></p>	<p><b>Container Farms</b></p>

 <p><b>Plant</b></p> <p>Humidity <b>35.28</b></p> <p>Temperature <b>23.64</b></p> <p>Moisture Level <b>628</b></p> <p>Water Level <b>0</b></p> <p><b>Plant</b> <b>GPS</b> <b>Security</b></p>	 <p><b>Geo Location</b></p> <p>Longitude / Latitude <b>-73.3986582, 45.2754398</b></p> <p>Vibration <b>X: 18, Y: -18, Z: 3</b></p> <p>Pitch <b>5.33</b></p> <p>Roll <b>0.89</b></p> <p><b>Plant</b> <b>GPS</b> <b>Security</b> <b>Charts</b></p>	 <p><b>Security</b></p> <p>Noise <b>0</b></p> <p>Luminosity <b>0</b></p> <p>Motion <b>False</b></p> <p>Door <b>False</b></p> <p><b>Plant</b> <b>GPS</b> <b>Security</b></p>
<p><b>Plant Measurements</b></p>	<p><b>GPS Measurements</b></p>	<p><b>Security Measurements</b></p>

<b>Plant Controls</b>	<b>Geo Location Controls</b>	<b>Security Controls</b>
 Plant  GPS  Security	 Plant  GPS  Security	 Plant  GPS  Security

<b>Telemetry Interval</b>	<b>User Access to Container</b>	<b>Adding User to Container</b>
<b>Telemetry Interval</b> Enter Telemetry Interval in Seconds <input type="text" value="20"/> <b>SAVE</b>	<b>User Access to Container</b>  admin Fleet Manager	<b>Adding User to Container</b>  admin Fleet Manager <b>Add User to Container Farm</b> What's the username of the user you would like to add? <input type="text" value="Spencer"/> <b>CANCEL</b> <b>OK</b> 

 <p>Charts</p> <p>Select Property to Chart</p> <h2>Pitch Chart</h2> <p>53 53 44 44 53 53 53 54 53 53 53 53 54 53</p> <p>Plant GPS Security Charts</p>	 <p>Charts</p> <p>Select Property to Chart</p> <p>Graph Data Points</p> <p>How many data points would you like to view? (0-50)</p> <p>15</p> <p>CANCEL OK</p> <p>q w e r t y u i o p a s d f g h j k l z x c v b n m 123 , . ✓</p>	 <p>← Current Location</p> <p>Terrebonne Boucherville Beloeil Montreal Brossard Chamby Napierville Lacolle Venise-en-Québec Champlain Swanton</p> <p>Google</p>
<p>Charts Page</p>	<p>Changing amount of data points within a chart.</p>	<p>Current Location Map</p>

## Future Work

- A demo view for guests to see how the application works, this way they could decide if they want to purchase the service or not.
- The user would have the option to install additional hardware. (i.e. extra fans, sensors)
- Implement a settings fragment.
  - Having themes (changing the color scheme on the fly).
  - Multiple languages (be able to toggle that app between english and french.)
  - If this app was to roll out on a global basis, we should offer a variety of different units for measurements. (ie. C -> F, mL -> L)
- Use of storage in IoT Hub
  - Storing historical data to present to users with a detailed history.
- Add more users, so that others can monitor the state of the farm without having any controls.
- Azure active directory for authentication instead of doing it locally.
- Using a salt to properly hash the passwords and any other sensitive data being stored.
- Survey others using our application to get feedback on user experience and design and implement any suggestions that will improve the application.
- Create a detailed user manual to describe the intended use of each feature and how it works.
- Utilize IoT hub storage so the user can better track data over long periods of time.

## Bonus Work

### Authentication

- We implemented local authentication for our application. This being said, we used the SQLite ORM (however, not the same package we used in class, this one supported multi-relational databases where children models were stored.)
- We didn't use Azure Active Directory or Auth for this particular project, mainly because we didn't believe we had enough time to get our hands dirty with cloud authentication. As mentioned in our future work section, the plan would be ideally to look into the Azure Active Directory for authentication since Microsoft technologies and products integrate into each other very efficiently.
- Nuget packages used:
  - Sqlite-net (1.6.292)
    - Used as the general ORM for the multi-relational database.
  - SQLiteNetExtension (2.1.0)
    - Used as an extension to the ORM (Data annotations)
- Beyond authentication, roles were also assigned to users, that being said dependent on the user's role, they had access to different aspects of the app.
- Something regarding authentication that we also plan on looking into is using a salt and hashing the user's passwords/personal connection strings.

### Dataset for the Container Farm Table

Table: ContainerFarm								
	ID	Name	Description	EventHubConnectionString	EventHubName	DeviceId	TelemetryInterval	IoTHubConnectionString
1	10	Brooklyn Farm	Brooklyn's Farm at home	Endpoint=sb://...	iothub-ehub-co-w22-...	co-w22-bpalf...	0.0	HostName=co-w22-bpalfy...
2	11	Spencer's Farm	Spencer's farm that never works.	Endpoint=sb://...	iothub-ehub-co-w22-...	skifellDevice	0.0	HostName=co-w22-skifell...

### Dataset for the User Table

Table: User					
	ID	Role	FirstName	Username	Password
1	1	1	Spencer	admin	admin
2	2	0	Brooklyn	user	user

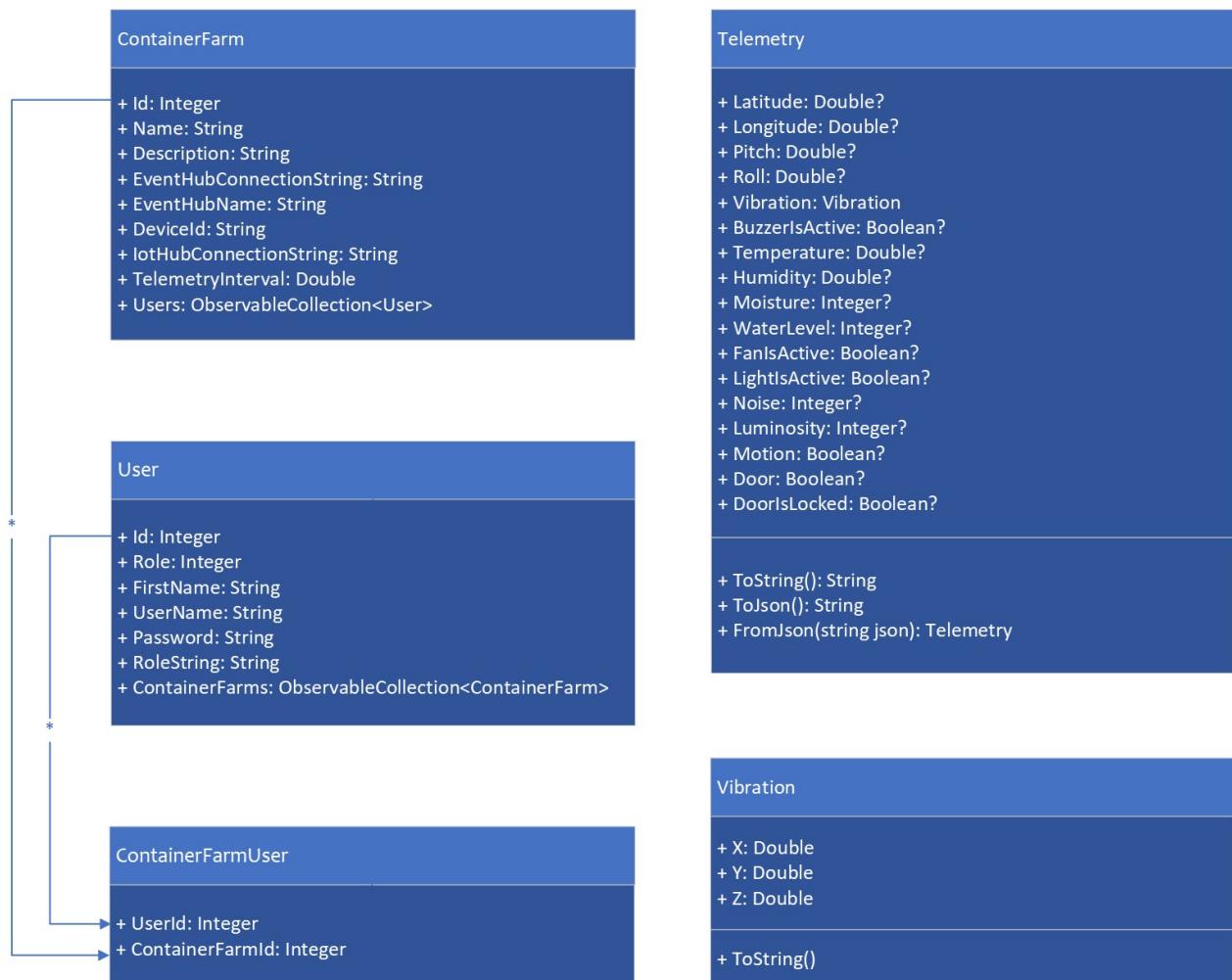
Dataset for the ContainerFarmUser Table (Many to many relationship)

Table: ContainerFarmUser

UserId	ContainerFarmId
1	9
2	10
3	11
4	11

# Object Oriented Programming Design

- Models:
  - User
    - This model is used to store the user's data.
  - Container Farm
    - This model is used to store the container farm's data.
  - Container Farm User
    - This model is used to represent the many to many relationship between users and container farms.
  - Telemetry
    - This model represents telemetry data for specific containers.
  - Vibration
    - This model represents vibration objects to be contained within a telemetry object/model.



## Contributions

Milestone	Brookelyn	Spencer	Vathusan
1	-Design Document -User Stories -Team contract	-Screen Analysis -Screen Design -Design Document -Team contract	-Design Document -Team contract
2	-Project organization -Model Creation -Page Design and navigation -Comments and documentation -User stories finalized	-Project organization -Model Creation -Data Repositories -Comments and documentation -video submission	-Model Creation -Comments and documentation
3	<i>-Focused on connected objects</i>	-Connection to Azure IoT Hub -Update data models -Display Data -Defensive Programming	-Defensive Programming
4	-IoT Hub communication -Documentation Update -Final Demo	-User Roles -Authentication -Update Jira board -Final Demo -Documentation Update -App advanced views -Documentation Update	-App advanced views -Final Demo

## Dummy Telemetry Data

```
{  
    "BuzzerIsActive": true,  
    "Door": false,  
    "DoorIsLocked": true,  
    "FanIsActive": true,  
    "Humidity": 50,  
    "Latitude": -46.6,  
    "LightIsActive": true,  
    "Longitude": -23.5,  
    "Luminosity": 0,  
    "Moisture": 20,  
    "Motion": true,  
    "Noise": 20,  
    "Pitch": 0,  
    "Roll": 0,  
    "Temperature": 25,  
    "Vibration": {"X": 25, "Y": 12, "Z": 76},  
    "WaterLevel": 10  
}
```