

Project Management/Tracking Procedures

Our team plans on using an Agile project management style with the addition of a gantt chart to help keep track of more long term goals. We believe this to be the most effective way to manage our team due to the fact that each aspect of our project is closely connected to one another. In order for the AWS team to complete their weekly goals the IoT team must also be making strides in goals related to sending data to AWS. The same goes for the front end requiring AWS data to be used for visualization and display. Having quick weekly meetings and Agile-like sprints will allow us to ensure that team work is happening and that nobody is being seriously bottlenecked by anybody else.

The use of a gantt chart on top of our standard Agile development will be useful to ensure we aren't spending more time than we have on specific aspects of the project. Each of the components of our system will require constant tweaking and development, however there are several major milestones we plan on tracking with our gantt chart. These milestones include full connection between our IoT base station and AWS, full connection between AWS and our frontend application, and different data visualization goals.

The software we plan on using to manage these goals is a combination of Github for code management and version control. As well as a shared KanBan board hosted on [Mattermost](#) to help keep our teams working efficiently and following our Agile development goal.

Task Decomposition

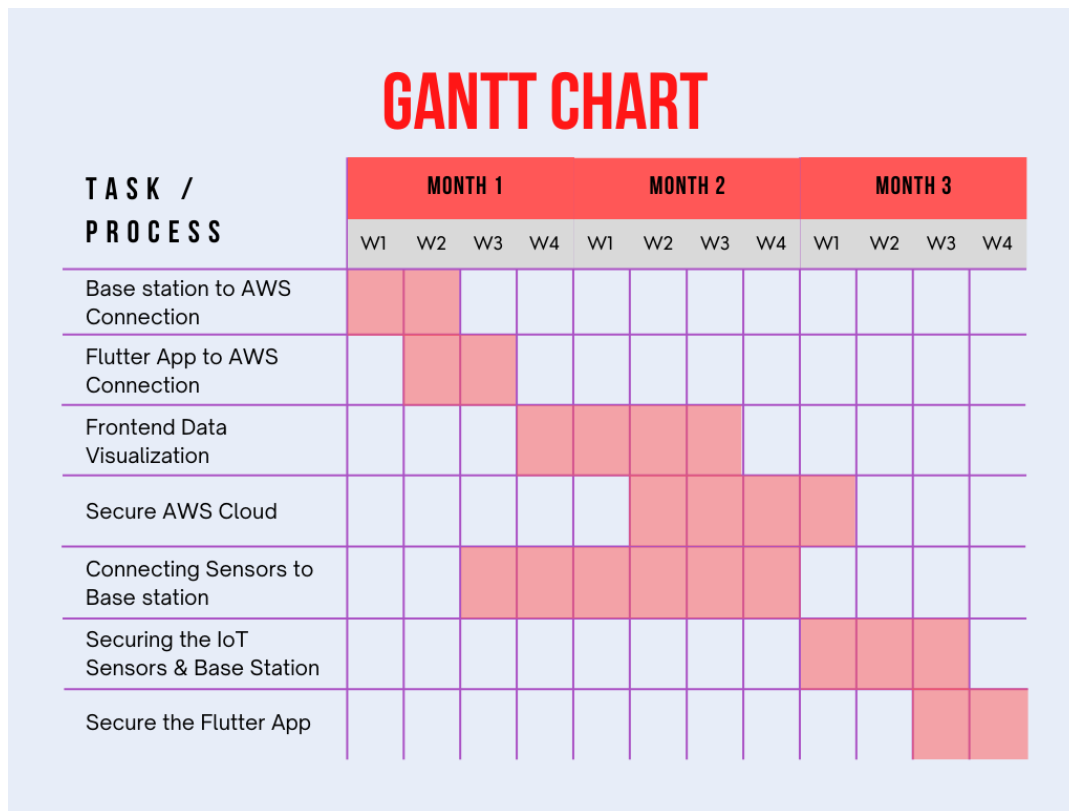
- IoT Base Station connection to AWS
 - Being able to send and receive data securely
 - Storing data securely and making it easily accessible
 - Ensure access to user specific data
- Flutter App connection to AWS
 - Being able to send and receive data securely
 - Allow transmission of data to legitimate users
- Frontend data visualization
 - Correctly displaying our data in a way that Farmers can understand and use
- Secure the AWS Cloud
 - Limited permissions
- Connect the IoT Sensors to the Base Station
 - Setting up LoRaWAN protocol to connect the sensors and base station
- Securing the IoT sensors and Base Station
- Secure the Flutter App

Project Proposed Milestone, Metrics, and Evaluation Criteria

- **Round trip data connection from IoT sensor to frontend**
 - Metric: Live updates when new sensor readings come in
 - Metric: Delay of no greater than 5 minutes
- **Secure login system for frontend**
 - Metric: Offensive security testing is unable to find flaws in login page
- **AWS data logging**

- Metric: Data stored in the cloud
- Metric: Data separated by user/group
- **Data analysis implemented in the cloud**
 - Metric: Data is being fed into AWS machine learning algorithm
 - Metric: Accurate assessments of the input data is returned from the algorithm
- **Front end data graphing**
 - Metric: Users are able to see useful visualizations of their data
- **Dark mode**
 - Metric: Users are able to choose between light and dark mode inside of app

Project Timeline/Schedule



Risks and Risk Management/Mitigation

Base station connection to AWS risks:

1. Cannot connect our base station to AWS - 0.2

Flutter App connection to AWS risks:

1. Cannot connect our flutter app to AWS - 0.2

Frontend Data Visualization risks:

1. Cannot use the data that is received - 0.1

Secure AWS Cloud:

1. AWS shuts down - 0.01

Connection Sensors to the Base Station:

1. The servers and base station are incompatible with each other - 0.3

Securing the IoT Sensors and the Base Station:

1. The sensors are proprietary so we cannot do any securing - 0.5

Risk Mitigation Plan: If the sensor ends up being proprietary and is unable to be secured by us, we will have to drop this task. It would be too far into the project to find a different sensor and connect it to our system. So, if this were to occur, we would add a task to replace this task where we do our sensor data validation and security in the cloud instead.

Secure the Flutter App:

1. The flutter app won't be able to be secured because it will not be set up properly - 0.2

Personal Effort Requirement

	David Wolfe	Thomas Ruminski	Joseph Hunter	Rian Lamarque	Eli Hanson	Syed Al-Hussain	Total Hours:
Base Station to AWS Connection	8	2	0	18	8	0	32
Flutter App to AWS Connection	0	0	10	4	8	4	26
Frontend Data Visualization	0	0	16	4	4	10	34
Secure AWS Cloud	0	0	0	12	16	0	28
Connect Sensors to Base Station	4	8	0	0	0	0	12
Secure IoT Sensors	10	12	0	0	0	0	22
Secure Base Station	14	14	0	0	0	0	28
Secure Flutter Application	2	0	10	0	0	8	20
Total Hours:	38	36	36	38	36	22	206

Other Resource Requirements

SenseCAP S2104 - LoRaWAN® Wireless Soil Moisture and Temperature Sensor - x2

- Cost: \$130- 2 of these needed \$260
- Battery Type: Standard D-size
- LoRaWAN Device Class A (least power consuming)
- No required modifications

Raspberry Pi with a LoRaWAN Transceiver

- RaspberryPi - x1
 - Cost: TBD
 - Want: built-in WiFi capability
- LoRaWAN Transceiver: RAK3172 Breakout Board - x1
 - Cost: \$15
 - Breakout board format is convenient
- Required Modifications: Bridge LoRaWAN input to WiFi, TCP/IP output