



Sprint Planning Document (Sprint 1)

Sprint Goal Backlog (Sprint 1)

January 18th - February 18th 2025

Ricardo R-C, Jacob Stock, Nick Warren, James Montgomery, Cameron Acree

High-level Project Overview

Project Mission:

- GUST is a ground station control software package that can manage multiple vehicle connections simultaneously and control them using MavLink. It will be able to plan paths for the vehicles to follow individually or as a swarm. It will also support real time sensor streaming

Problems We Are Solving:

- No ground control software currently exists that can manage swarms effectively.
- Researchers need the ability to plan maneuvers for teams and swarms of uncrewed vehicles

Project Overview (High-Level Features):

- **Front End Application:**
 - **Telemetry Screen:** The telemetry screen allows the user to select a single or multiple vehicles and view their positions and telemetry data such as altitude, velocity, roll, pitch, and yaw
 - **Path Design Screen:** The path design screen will allow users to design a path for a single vehicle and combine multiple paths for multiple vehicles into a single swarm maneuver.
 - **Connection Widget:** Users will be able to easily connect to different vehicles and manage the connections. This widget will display connection strength to any of the vehicles
 - **Live Sensor Streaming:** Users will be able to view live streamed data collected from sensors on any chose vehicle
 - **Emergency Protocol:** In case of an emergency, the application will be able to send a safe land command to any vehicle with one button click
 - **Tele-Operation:** Users will have the ability to fly any selected vehicle with their keyboard
- **Backend Services**
 - **Realtime Database:** Data for vehicle telemetry and pathing
 - **Hosting:** The application will run locally from a docker container that also hosts the database.

Sprint 1 Planning

Sprint 1 Goals:

1. Design Front End & Implement Telemetry Screen
2. Database Design for Telemetry
3. Backend Design for Telemetry
4. Vehicle Control Verification & Test in Simulation
5. Vehicle Control Verification & Test on Hardware

Sprint 1 Deliverables:

- **Research the tools to be used in this project:**
 - **Assigned:** Ricardo
 - Research and read through all of the necessary documentation for the tools to be used throughout the project.
- **Setup the Backend:**
 - **Assigned:** Nick
 - Setup and test the FastAPI backend
 - Create endpoints for telemetry
- **Design and Implement the Data Base:**
 - **Assigned:** James
 - Design and implement the data base using PostgreSQL to support telemetry
- **Design Setup the Front End**
 - **Assigned:** Jacob, Ricardo
 - Design the layout of the front end of the application and handle the setup of the base react application
- **Implement the Telemetry Screen Front End**
 - **Assigned:** Jacob, Ricardo
 - Install the necessary React components
 - Implement and style the telemetry screen
- **Integrate Backend and Data Base**
 - **Assigned:** Nick, James
 - Make all the telemetry api calls match the appropriate modification functions for the database
 - Write tests to evaluate the functionality of the endpoints

- **Create MavLink Test Script for Simulation Flight**
 - **Assigned:** Cameron
 - Build and test a script to interface with an emulated flight controller and flight simulator that can command vehicles to fly to waypoints in a certain order
- **Test MavLink on Vehicles**
 - **Assigned:** Cameron, Ricardo
 - Test the flight control script on actual vehicles in the LAGER laboratory.