GUST

Sprint Planning Document (Sprint 2)
Sprint Goal Backlog (Sprint 2)

February 18th - March 25th 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.
    - **Flight Screen:** The flight screen will allow users to select planned manuevers and execute them.
    - **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**
    - **Persistent Database**: Drone information and flight path persistence.
    - **Hosting:** The application will run locally from a docker container that also hosts the database and other necessary services (mavlink, gazebo, etc).
    - **Message Routing**: FastAPI will receive messages from frontend and route through mavlink to drone hardware. Also will route mavlink telemetry to the react front end.
    - **Auto Documentation**: All endpoints will have documentation which is handled through the FastAPI app.

- **Efficient API->Drone Communication**: The backend aims to be as efficient as possible to be able to handle multiple drone connections. Reducing overhead is the primary driver of this.
- **Mavlink Abstraction**: API Endpoints will be used to abstract away some detail with sending commands to the vehicles

Sprint 3 Planning

Sprint 3 Goals:

1. **Finalize Telemetry Screen with Backend**
2. **Finalize Planning Screen with Backend**
3. **Design & Implement Vehicle Connection in Backend**
4. **Test Multiple Vehicle Control on Hardware (HIL Testing)**

Sprint 3 Deliverables:

- **Test with Dragon Link Radio in Lab:**
  - **Assigned:** Nick, Cameron
  - Set up and connect to a drone to send mavlink commands over radio
- **Implementation of Telemetry Screen**
  - **Assigned:** Jacob
  - Finish implementation of map component to track multiple connected vehicles
  - Finish implementation of list component to display all connected vehicles
- **Design and Implementation of Planning Screen**
  - **Assigned:** Ricardo
  - Design and implement map planning component
  - Design and implement front end path creation
  - Design and implement swarm maneuver builder

Final Design of Telemetry Screen with Backend

- **Assigned:** Nick, James, Jacob
- Modify and integrate database access as needed

- **Final Design of Planning Screen with Backend**
  - **Assigned:** Nick, James, Ricardo
  - Finished designing the planning screen in the frontend

- **Design of Connection Management System**
  - **Assigned:** Nick, James
  - Design system to accept and manage vehicle connections

- **Testing of MAVLink with Backend Systems**
  - **Assigned:** Nick, Cameron
  - Test code from standalone vehicle control test script into backend