Goals:

1. Set up the backend to server telemetry data
2. Set up the database to hold telemetry data
3. Design the front end and implement the telemetry screen
4. Test vehicle control via MavLink library in MavProxy
5. Test MavLink vehicle control on hardware

Sprint Backlog:

- Set up DB in docker container [done, ~1hr]
- Design DB schema [done, ~5hrs]
- Set up SQLAlchemy [done, ~1hr]
- Implement database structure and define access [done, ~5hrs]
- FastAPI installed on docker container [done, ~1 hr]
- Basic get and post requests using FastAPI [done, ~3hrs]
- FastAPI calls to get and set drone data [done, ~3hrs]
- Backend integration with DB [In Progress, ~2hrs]
- MavLink installed and vehicle control research [done, ~3hrs]
- MavProxy setup and flight controller emulator [done, ~2hrs]
- MavLink standalone script that flies to waypoint in sim [done, ~2hrs]
- MavLink integrated with backend [In Progress, ~4hrs]
- React base app created and functioning [done, ~1hr]
- Drone select drop down screen 1 [done, ~2hrs]
- Map API selected and component implemented [done, ~3hrs]
- Map svg overlay to display position of selected vehicle [In Progress, ~3hrs]
- React app gauges for altitude and velocities [done, ~2hrs]
- React app “navball” for roll, pitch, yaw [In Progress, ~2hrs]
- Front end screen 1 integrated with backend [In Progress, ~2hrs]
- MavLink test on hardware [In Progress, ~6hrs]