Animal Locator Drone

Drone software for locating animals with a focus on dogs.

20240309 Proposed design based on simulators

- Initial Author: Ziad Arafat

1 Overview

1. In this design we will focus on implementing compoments in a simulation environment rather than having to worry about implementing things around a particular drone setup.
2. Things such as drone connectivity, availablity of a secondary board etc. won't need to be a consideration.
3. However, in order to help make this design portable and adaptable to a wide range of drone setups we will make it modular and non monolithic in nature.
    1. This means we will break up functionality into multiple running services
        1. Dare I say "microservices"
4. Some real challenges we have identified with designing this system.
    1. What is the necessary flow of logic when it comes to identifying dogs and further investigating the found dog?
    2. How do we efficiently stream video data from the cameras and deliver it to multiple endpoints that need it?
    3. Which endpoints need said video and which do not?
    4. In the context of having a physical drone we encountered these challenges which we hope to not deal with in simulated setups.
        1. How do we connect the end user to the drone?
        2. How do we stream video data and to whom given the limitations of connectivity?
5. Updated model for defining and refining the design documentation.
    1. RADIO model
        1. https://www.greatfrontend.com/system-design/framework

2 R A D I O

1 Requirements Exploration

1. We will maintain a list of functional requirements and non-functional requirements here.
    1. **Functional Requirements** (What does it do for the user?)
        1. **Primary Requirements** (These are essential for an MVP)
            1. The system will help the user quickly locate their lost dog.
                1. The drone can autonomously follow waypoints set by the user.
                2. During flight the user will receive real-time detections of dogs on the ground
                    1. A "detection" constitutes the following
                        1. A photo or multiple photos of the detected dog.
                        2. The precise location of the dog.
                        3. A graphical map representation of the location.

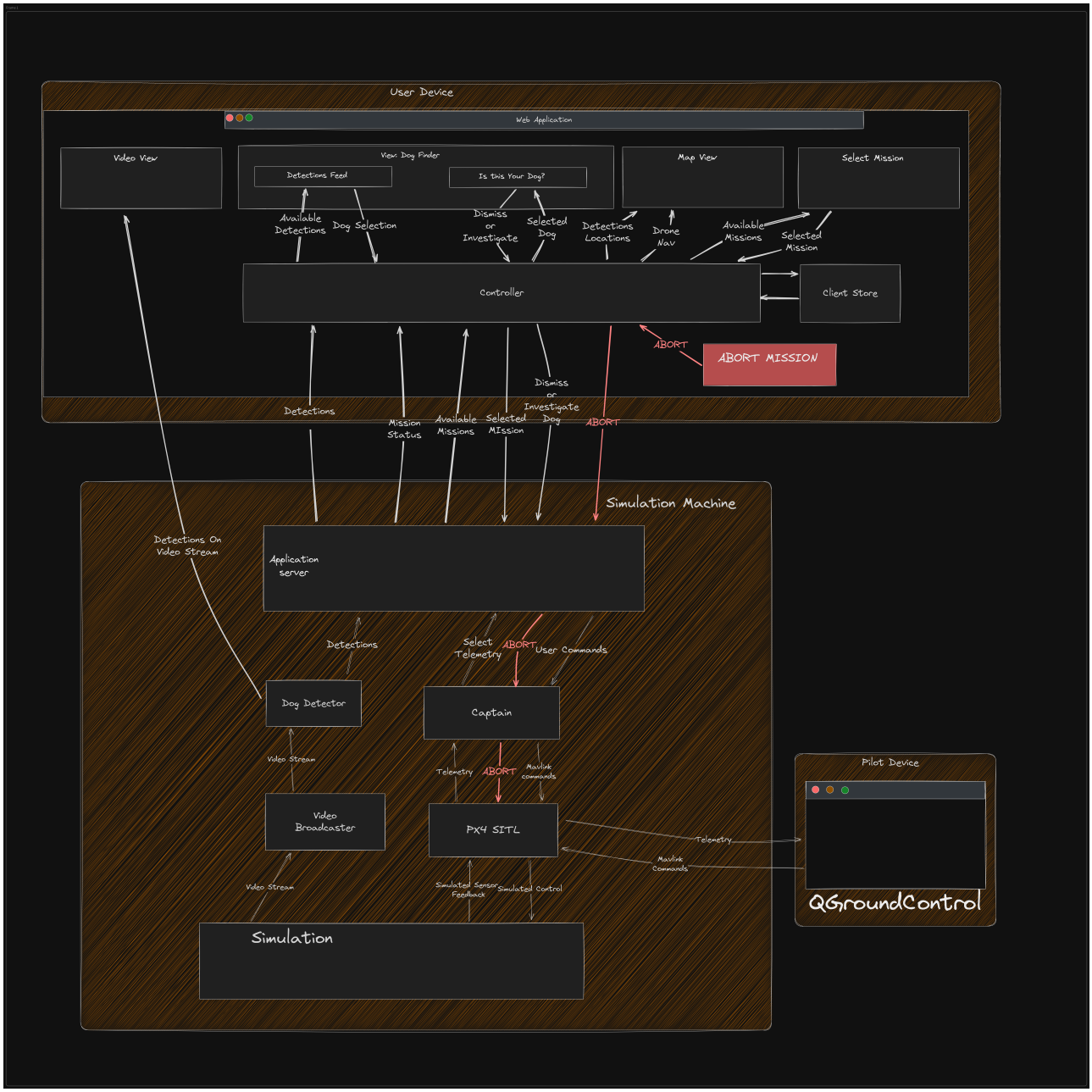
4. Directions to navigate in person to the location (like google maps).
2. The user will be able to browse through detections of dogs
3. For each detection
    1. The user can indicate if it should be ignored or investigated further.
3. For each detection that is marked for further investigation.
    1. The drone will autonomously track the dog and attempt to get a better view of the dog.
    2. As the drone is doing this
        1. The user can cancel at any time to continue the search for other dogs.
        2. The user can indicate that this is the correct dog and begin efforts of recovery.
4. When the user has begun recovery efforts the drone can be used to continue tracking the location of the found dog until recovered.

2. **Non-functional requirements** (Constraints and general operational requirements)
    1. The UI needs to be user friendly.
    2. Drones need to be operated safely and in accordance to laws and FAA regulations.
    3. One intended use case is for there to be a community drone that can be operated for a community to help locate lost dogs.
        1. Therefore it's important to consider the cost and easy of setup for this use case.

2 Architecture and high level design

1. Key Components and their relations
    1. Simulation Machine (Powerful machine that will run the simulated drone and it's other services)
        1. Simulation
            1. This is used to simulate the physical drone and the environment it interacts with. This is likely going to be Airsim/Colosseum or Gazebo.
        2. PX4 SITL
            1. PX4 is the firmware used on many drones and it can run Software-In-The-Loop mode which means it is simulated in software.
            2. This can connect to a simulation platform.
        3. Video Broadcaster
            1. Takes a video stream as input and will process it and allow other clients to subscribe to the broadcast.
        4. Dog detector
            1. This service will read in video data and make predictions on where there are dogs or other important features in the video. It can do this in real time.
        5. Captain
            1. The captain is responsible for guiding the drone based on user inputs and other factors.
        6. Application server
            1. The application server provides an interface between the user device and other services on the simulated drone. It does this by serving a web application and a

web API.

2. User device (A device used to access the UI application and command/monitor the mission.)
    1. Web Application
        1. This web application will be served from the application server and run in a web browser.
3. Pilot Device (specialized device that is meant for an experienced pilot to monitor and control the drone as needed.)
    1. QGroundControl
        1. This is the standard software used for configuring and commanding PX4 flight stacks.
2. 

3 R

First Proposed Design

1. Proposed design diagram
    1. https://excalidraw.com/#room=7e2e5fed98110a848b6a,OxINdgfSKA8y0GQ-5m-grA

2. The overall design is as follows.
    1. The goal is for a user to be able to use a drone that will fly over a large area of ground autonomously and try to locate dogs in real time.
    2. Drone
        1. The drone will be equipped with sensors and computers to be able to autonomously fly over a path and detect dogs on the ground.
        2. Flight Controller
            1. The flight controller will manage autonomous flight and following preset waypoints.
        3. Secondary Board
            1. A secondary compute board on the drone will take care of processing and transmitting data including.
                1. Processing input video/image data from cameras
                2. Looking for pets in video/image data using AI
                3. Transmitting data back to the user
                4. Processing user input from the user to command other components.
            2. Web App
                1. The web app will provide an interface for the user app to communicate with the drone and also receive data from the drone.
            3. Database
                1. The database will store any persistent data the system needs.
            4. dog detector service
                1. This component will take images/videos as input and provide useful data about found dogs.
            5. Flight Service
                1. The flight service will talk to the Flight Controller to command missions and receive telemetry data needed for the user.
    3. User device
        1. Phone or tablet running an app that will interface with the web app to allow the user to
            1. Select a flight mission and begin missions.
            2. Customize settings for searching for dogs.
            3. See the current status of the mission
            4. See images of dogs and generally what the drone sees
        2. Views
            1. Dog Finder
                1. Subviews
                    1. Feed of detected dogs
                        1. Allows you to see pictures of found dogs and browse through them.
                        2. Selecting one will display it in "is this your dog"
                    2. "Is this your dog" card
                        1. Includes a picture of the dog
                        2. location of the dog on map
                        3. Option to select if it should be further investigated or ignored.
            2. Video View

            1. A live stream showing what the drone sees and detections on screen in real time.
        3. Map View
            1. An interactive map
                1. Location of drone
                2. Locations of detected dogs.
        4. Select Mission
            1. View at the beginning to select which mission the drone should fly.
            2. Also displays a map showing the route of the mission.
        5. ABORT MISSION
            1. Button to abort the current mission.
            2. Always visible
    3. Controller
        1. Processes data produced by the user as well as data retrieved from the backend server.
        2. Caters the data to be best viewed by the individual "Views"
        3. Stores and retrieves data from the client's local data store. "Client Store"
    4. Client Store
        1. Used to store data relevant to the current mission in the RAM of the client.
4. Ground Pilot Device
    1. The ground pilot device will allow a pilot to monitor and take emergency manual control of the drone.
    2. This is a requirement by the FAA for all autonomous drones and a critical safety measure.
    3. Communicates directly with PX4 SITL via mavlink.

Implementation

1. Drone platform
    1. Simulated drone
        1. For the drone platform itself we will start with a simulated drone using the colosseum simulator and possible Gazebo.
            1. https://github.com/CodexLabsLLC/Colosseum
            2. https://microsoft.github.io/AirSim/
            3. https://gazebosim.org/home
        2. Flight Controller
            1. For the flight controller we will use PX4 in SITL mode
                1. https://px4.io
    2. Physical Drone
        1. For the physical drone we plan to use a basic drone frame kit with a classic pixhawk flight controller capable of running PX4
            1. Currently we are looking at the Holybro X500 v2
                1. https://www.getfpv.com/holybro-x500-v2-arf-kit.html?utm_source=google&utm_medium=cpc&utm_campaign=DM+-+NB+-+PMax+-+Shop+-+SM+-+ALL+|+Full+Funnel&utm_content=pmax_x&utm_keyword=&utm_matchtype=&campaign_id=17881616054&network=x&device=c&gc_id=17881616054&

gad_source=1&gclid=CjwKCAiAq4KuBhA6EiwArMAw1KnzoqzcRcEUzLdYfhrSO0TbaQLhCj82WURUPezR6SqhG48fqDwV6RoCIL4QAvD_BwE

        2. We have 2 pixhawk controllers available already.
    2. Controller
        1. We are currently looking into a Herelink controller from ETV otherwise we will need to do more research
2. Secondary Board Components
    1. The Board itself
        1. Currently we are looking into either a Raspberry pi or to borrow an Nvidia Jetson Nano board from ETV
    2. Web App
        1. See https://github.com/animal-locator-drone/dogfinder-backend/wiki/Proposed-Design-and-Implementation
    3. Database
        1. PostgreSQL or SQLite but we are undecided.
    4. Dog detector service
        1. No implementation plan yet
        2. Ideas
            1. OpenCV
            2. YOLOv8
            3. Python
    5. Flight Service
        1. No implementation plan yet
        2. Ideas
            1. PyMavSDK
            2. Python
            3. FastAPI
3. User Device
    1. Device itself
        1. No plan yet but anticipate a laptop or tablet or phone
    2. Frontend App
        1. See: https://github.com/animal-locator-drone/dogfinder-app/wiki/Proposed-Solution