Autores:
Diogo Amaral 93228 | Guilherme Pereira 93134 | João Rainho 92984 | José Costa 92996

Mentors:
Prof. Susana Sargento | Prof. Miguel Luis | Margarida Silva

System Requirements and Architecture

In this report we will present the system requirements specification beginning with a context of what is the work about, a brief description on the requirements gathering process, followed by the functional and non-functional requirements, context description, actors classification, use case diagrams, the system architecture, mockups of the dashboard and lastly the project calendar.

Context of our project

Nowadays aerial drones are becoming very popular as they also become more accessible, in case of usability and economics terms. Drones can have reduced size and good cost-benefit ratio which make them an excellent alternative for some specific objectives/missions. Some scenarios that benefit from the use of aerial drones are the surveillance of important individuals, monitor public-gathering restrictions, monitor forest areas or disaster areas, transfer lightweight high value objects, watch live video transmission, etc...
Taking all of that into account we decided to create an optimized relay to be integrated into a modular infrastructure, enabling the autonomous control and monitoring of a fleet of aerial drones in a mission context to manage many different situations.
The project is divided in three main parts, one onboard computer of the vehicle (drone), a ground station and a dashboard, everything will be explained in detail further up.

Requirements Elicitation

To gather all the requirements in need we started by meeting with our supervisors, professor Susana Sargento, professor Miguel Luís and Margarida Silva to evaluate the goals and helping define the system boundaries. Then we analysed the state-of-art, that is, researching for studies, technologies and studies similar to the one that we will develop. At last, a brainstorming session to get all the requirements collected and debate which ones were important or disposable.
All of that resulted as a big help for us to define the most important functional and non-functional requirements.

Functional Requirements

The following list presents functional requirements, i.e. features or functionalities that the system must have to allow the user to accomplish their tasks/objectives. We divided the requirements in three main modules, Dashboard, Ground station and Drone:

Dashboard:

- FD1: Should be possible to visualize real time video sent from the drones
- FD2: Should display information about the mission
- FD3: Must be able to display telemetry information from the drones
- FD4: Must be possible to adjust relay parameters to optimize drone position

Ground Station:

- FG1: Must be able to process network characteristics between the ground station and the drones
- FG2: Should be able to place relay drones in a optimal position taking into account communication and services parameters
- FG3: The system should be able to automatically adapt telemetry information sending ratio
- FG4: The system should be able to automatically adapt video parameters

Drone:

- FDR1: Must be able to monitor network characteristics between the ground station and the drones
- FDR2: Monitor the network characteristics between itself and the other drones and transfer that information to the ground station
- FDR2: Should be able to send video with different codecs and quality
- FDR3: Should be able to adapt telemetry sending ratio according to the ground station

Non-Functional Requirements

The following list presents non-functional requirements, i.e. define system attributes such as scalability, reliability, usability, security, maintainability, and performance. We divided the requirements in three main characteristics, Usability, Performance and Documentation:

Usability:
- NFU1: Provide a simple, complete and intuitive dashboard
- NFU2: The system must have a familiar visualization and interaction
- NFU3: To add new functionalities should be effortless on the system

Performance:
- NFP1: The network sensor must provide new data automatically
- NFP2: The camera must provide new information automatically
- NFP3: Relay should be sent almost immediately to the correct position

Documentation:
- NFD1: Must have a documentation easy to understand about the dashboard
- NFD2: Documentation about all the sensor and specific characteristics
- NFD3: Documentation with information about how to use the system
- NFD4: Documentation about how the whole system works together and plugins

Context and State of Art (SOA)

On this section we analysed studies, projects and technologies already developed about this topic, we considered the most relevante the next:

Drone-Based Wireless Relay Using Online Tensor Update:
(https://ieeexplore.ieee.org/document/7823731)

On this paper, drones are advocated to serve as mobile relays to forward data streams but some problems exist with that, for example, data transmission may suffer severe signal attenuation due to obstructions and it is also difficult to find an optimal location for drones due to the dynamic and unpredictable environments. To try to solve this problem the authors developed an algorithm that outperforms existing methods in achieving the trade-off between time cost and estimation accuracy.

Performance Improvement of Drone MIMO Relay Station Using Selection of Drone Placement:
(https://ieeexplore.ieee.org/document/8536637)

The objective of this paper is to evaluate multiple-input multiple-output (MIMO) transmission when a small autonomous unmanned aerial vehicle (drone) is used as a relay station. When using drones the propagation loss decreases but it has the issue of increased spatial correlation due to the direct wave. To solve this they introduce a propagation environment control method (PECM) that selects drones in optimal arrangement from multiple drones.

Actors

The target users of this project are those in the security force, civil protection and any other individual or organization that may need to perform tasks such as surveillance, disaster prevention, patrolling difficult access zones, etc.
The level of expertise required to use the system varies according to the needed features.
For example, less experienced users may easily retrieve information from the system, such as sensor information, video footage or overseeing a mission, as these are straightforward tasks. However, the user must have some knowledge in order to define the missions or set up the drones.

With that in mind the main actors are described in the following list:

- **Security Guard:** Represents security forces that use drones equipped with video cameras and other sensors to perform tasks such as monitoring public gathering restrictions or surveillance missions.

- **Civil Protection:** Make use of drones to perform missions that may help preventing disasters or aid in an emergency, such as monitoring forest areas or support search-and-rescue missions.

- **Drone Administrator:** Users capable of defining the missions, setting up the drones, handling other maintenance issues and overseeing the missions.

Use Case

The use case model consists in one package which is the web application(dashboard) that will provide the exchange of information between the users and the drones.

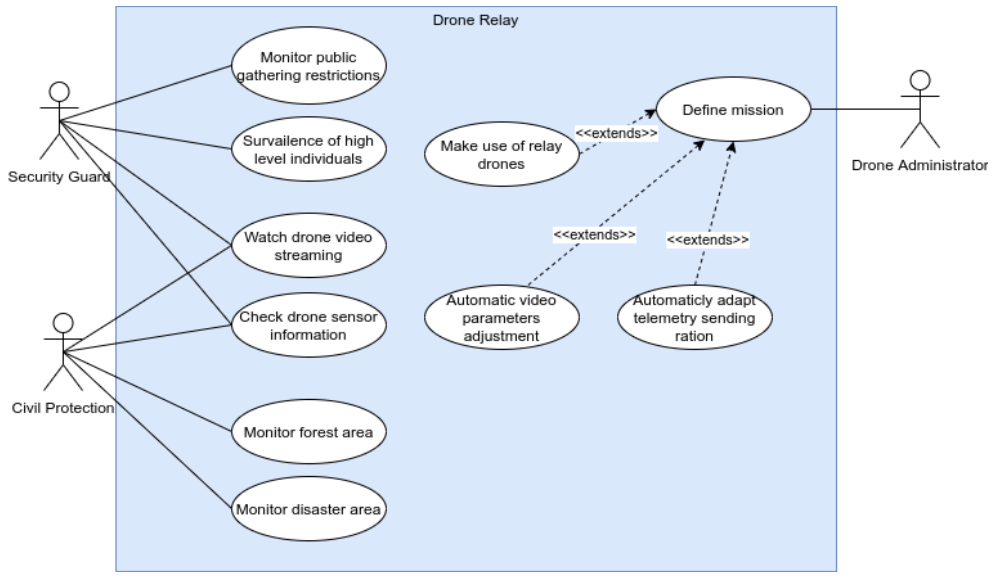

Drone Relay
Monitor public gathering restrictions
Survailence of high level individuals
Watch drone video streaming
Check drone sensor information
Monitor forest area
Monitor disaster area
Make use of relay drones
Define mission
Automatic video parameters adjustment
Automaticly adapt telemetry sending ration
<<extends>>
<<extends>>
<<extends>>
Security Guard
Civil Protection
Drone Administrator

Figure 1: Use case

Drone Administrator

- **Define mission:** program the mission which the other users will be able to use and provide ways for them to provide the information for the program to execute (eg: area of the search).

- **Automatic video parameters adjustment:** configure how the video parameters change for the given network.

- **Automatically adapt telemetry sending ratio:** configure how the telemetry will change accordingly with the given network.

Security Guard

- **Monitor public gathering restrictions:** allow the security guard to check if public gathering restrictions are being followed by, making it easier and faster for the officer to find and attend such events.

- **Surveillance of high level individuals:** provide easy access for officers to follow and watch high level interest individuals.

Civil Protection

- **Monitor forest area:** allow the civil protection user in charge to check a forest area after a fire or a similar event.

- **Monitor disaster area:** allow the civil protection user in charge to check a specific area after/during a disaster or a similar event.

User

- **Watch drone video streaming:** allow the user to get access to the drone live footage or in passed missions.

- **Check drone sensor information:** show current sensor information and last readings from the current mission as well as past missions.

System architecture (Deployment Diagram)

This Figure (3) illustrates a simplified deployment diagram of this project. The aim of this diagram is to capture main components of the deployment, using a simple, easy to understand language.
This diagram is further detailed in Figure 2, that elaborates on the internal interconnects of each main component.

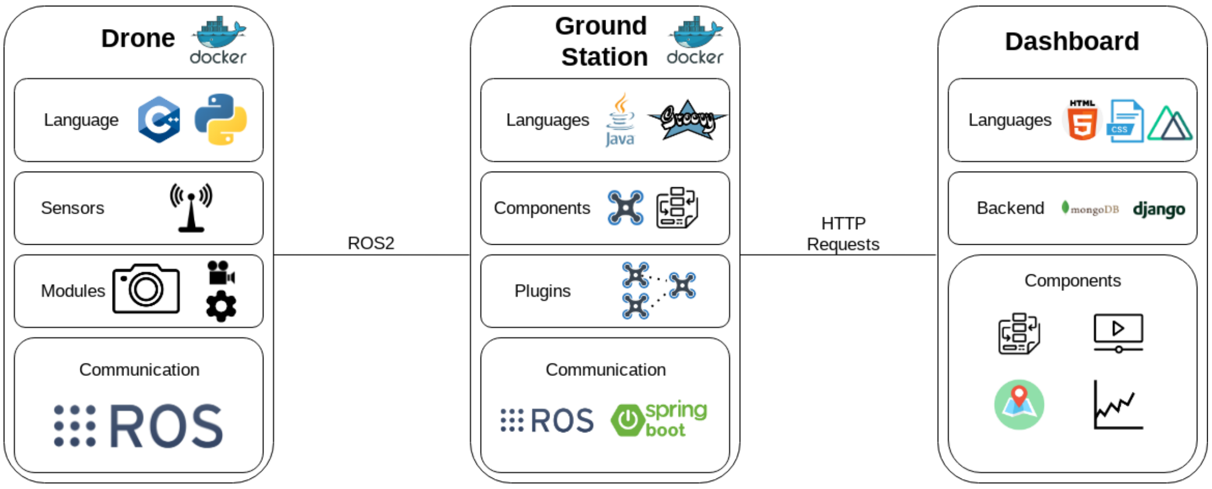


Figure 3: Simplified deployment architecture

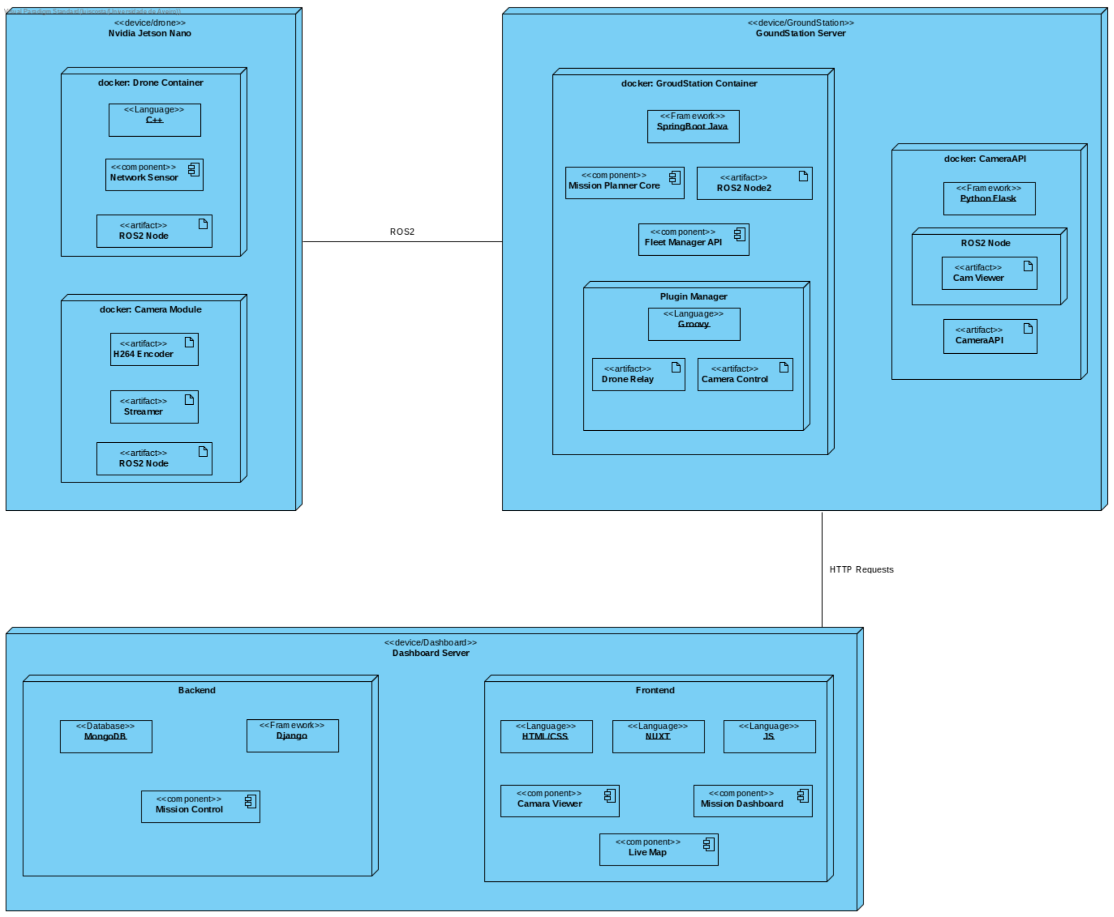


Figure 2: Deployment architecture

The system architecture presented in the figures above, show the three main components of this project:

- **Drone (Nvidia Jetson Nano) :** This module consists of 2 docker containers, the first is responsible for the control of the drone main board with the use of mavsdk, a C++ library, and the second responsible for the camera capture, transmission and dynamic update of the video stream quality.
  The communication is provided via a ROS2 node, that communicates (via subscriptions and publishes) to the ground station. The network sensor runs via a python script that transmits information about the network signal and its status.

- **Ground Station:** This module consists of a docker container which contains a Flask Server and a SpringBoot Server.
  The SpringBootServer consists of a previously developed fleet-manager that is used to control, organize and coordinate drone missions, the system allows for the development and use of plugins. The standard plugins created are the drone relay, which using artificial intelligence automatically positions relay drones during missions, so that the mission drones never lose connection to the ground station and optimize the number of relay drones.
  The flask server is used to transmit video feed and control quality from multiple drones to be displayed in the dashboard.

- **Dashboard:** This module consists of a developed frontend and a backend, that the user will use to interact with the mission, sensors and camera system.
  The backend uses mongodb as a database and djago to connect to the ground station for control and management.
  The frontend is made with Nuxt, which is a framework that extends vuejs made to create interactive SPA’s, and allows the upload and tracking of missions, connect to multiple drones and ground stations, displaying a map for easy interpretation.
  This dashboard also contains a section dedicated to the display and control of the drone camera quality settings, and multiple charts that illustrate the continuous status of the attached drone sensors.

System architecture (Domain model)

The domain model presented in Figure 4 describes the entities, roles and relationships of the system. This model will help to better understand the problem.
To begin with, there is a Drone administrator which, as described in the previous chapters, is capable of setting up the missions and configuring the ground station.The ground station then through the configuration will make use of one or more drones to coordinate the missions defined by the drone administrator.
As for the users represented by the security guard and civil protection,they will oversee the mission in order to, for example, retrieve information from the system, such as sensor information or video footage.

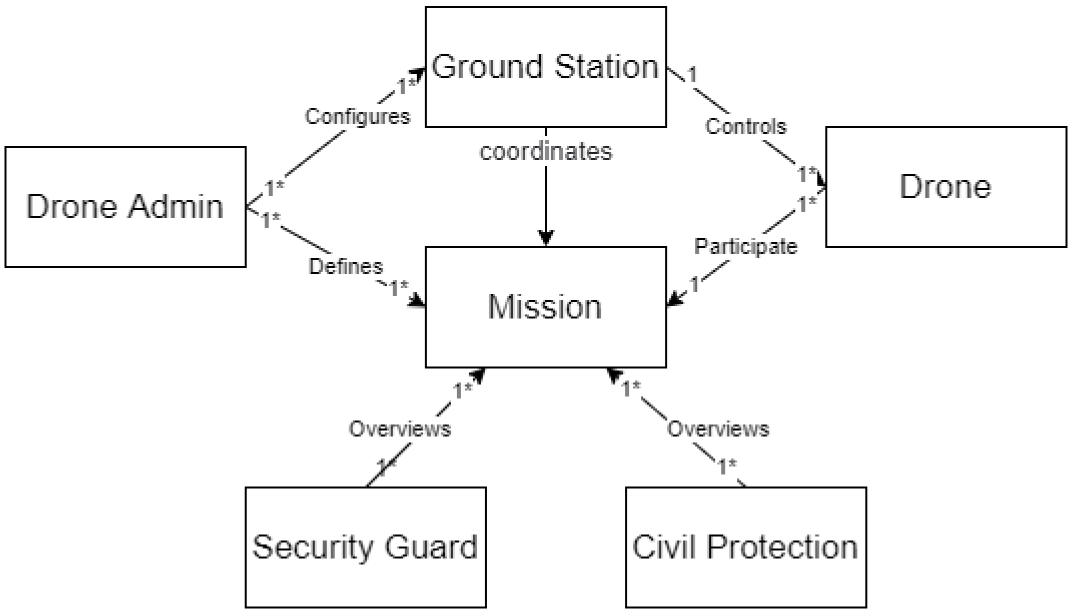
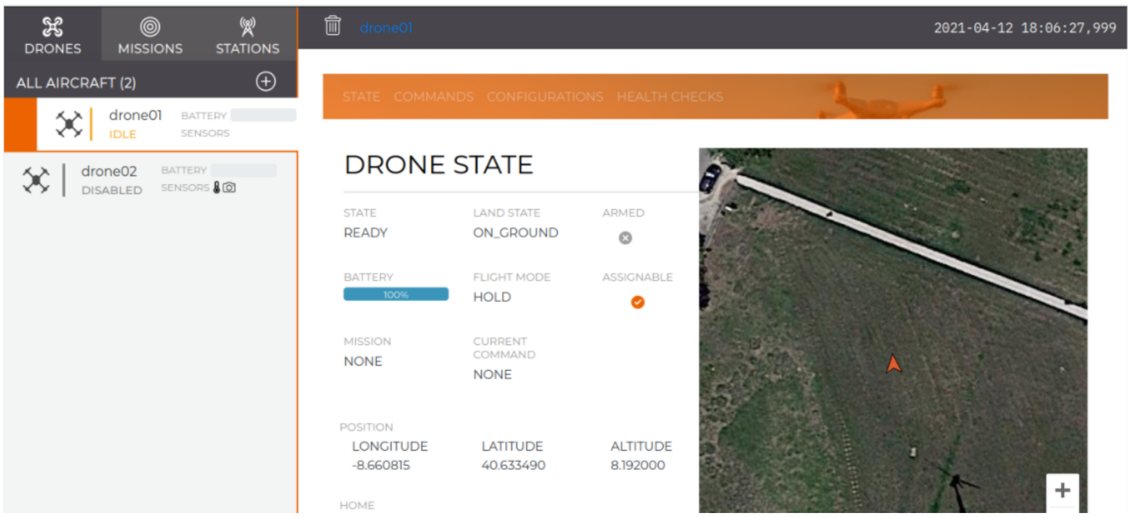


Figure 4: Domain Model

Dashboard and Mockups

The dashboard for this project will be the contact point between the user and the other modules so the intuitiveness of this dashboard will be highly important.
An already existing dashboard will be the base for this project and an overview of the existing dashboard can be seen in Figures: 5,6,9. Although the dashboard will have to be in general altered to accommodate the necessary features for this project some mockups have been made for some of the new functionalities. The mockups of the tab in the dashboard that will allow the users to see the drone live camera feed can be seen in Figure 7 and the Tab that allows the user to check all the sensor charts can be seen in Figure 8.



drone01 IDLE BATTERY SENSORS
drone02 DISABLED BATTERY SENSORS

DRONE STATE

STATE	LAND STATE	ARMED
READY	ON_GROUND	✗
BATTERY	FLIGHT MODE	ASSIGNABLE
100%	HOLD	✓
MISSION	CURRENT COMMAND	
NONE	NONE	

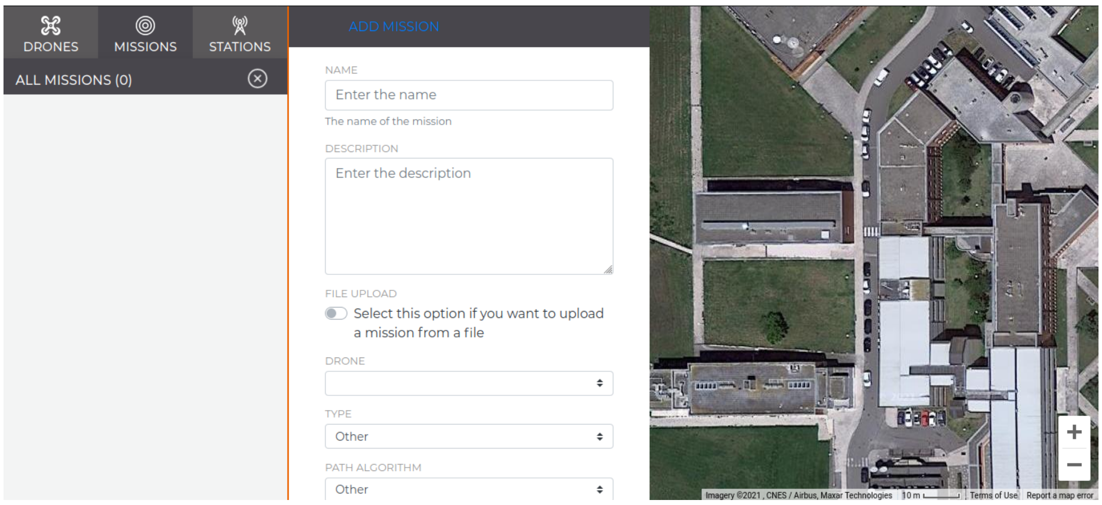
POSITION

LONGITUDE	LATITUDE	ALTITUDE
-8.660815	40.633490	8.192000

HOME

Figure 5: Telemetry data

This figure (5) illustrates the multiple tabs associated with each drone, the selected “Status” tab shows drone telemetry data, this includes the state of the drone, its coordinates, location on the map, battery status and the information related to the drone control and mission. Other tabs are also illustrated in this figure, the commands tab that is used to manually command the drone, and the configurations and health check tab, that show information related to each topic.


DRONES
MISSIONS
STATIONS
ALL MISSIONS (0)
ADD MISSION
+
−
Imagery ©2021 , CNES / Airbus, Maxar Technologies 10 m Terms of Use Report a map error

NAME
Enter the name
The name of the mission
DESCRIPTION
Enter the description
FILE UPLOAD
Select this option if you want to upload a mission from a file
DRONE
TYPE
Other
PATH ALGORITHM
Other

Figure 6: Upload mission

This figure (6) illustrates the process of uploading or creating a mission to be performed by the drones.
The common parameters required for these processes are the name and description of the mission. If the user chooses to upload a mission file, the mission can be added instantly, otherwise, the user can select further parameters, and click on the map to design the mission plan.

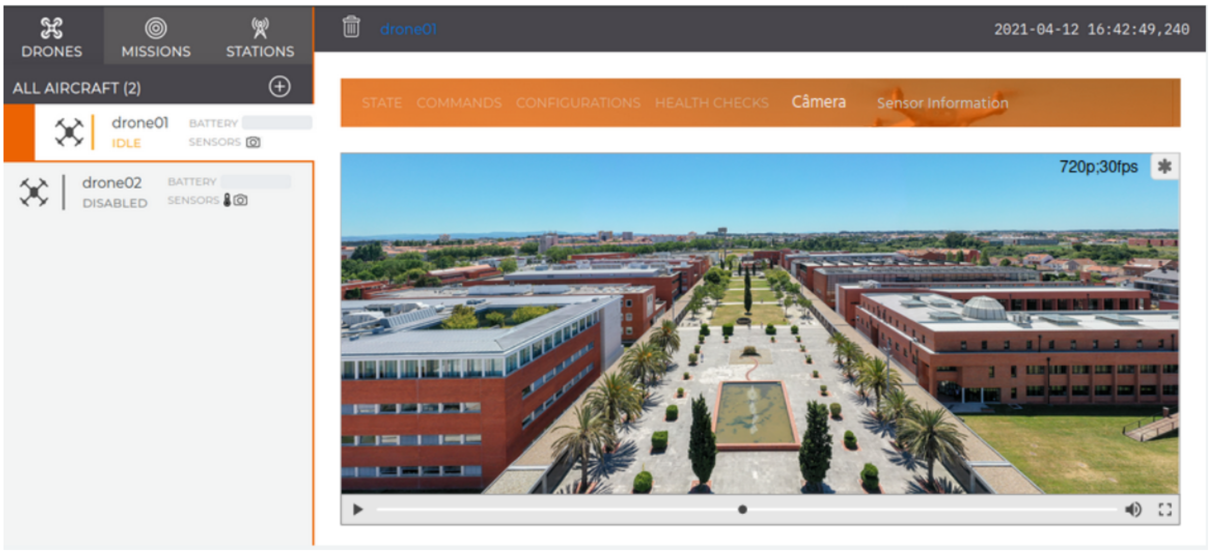


Figure 7: Camera live feed

This figure (7) shows a mockup of the camera menu, which will be used to display the live drone camera feed, with optional manual ou automatic quality control, based on network characteristics, by selecting the wheel icon located in top right part of the video container.

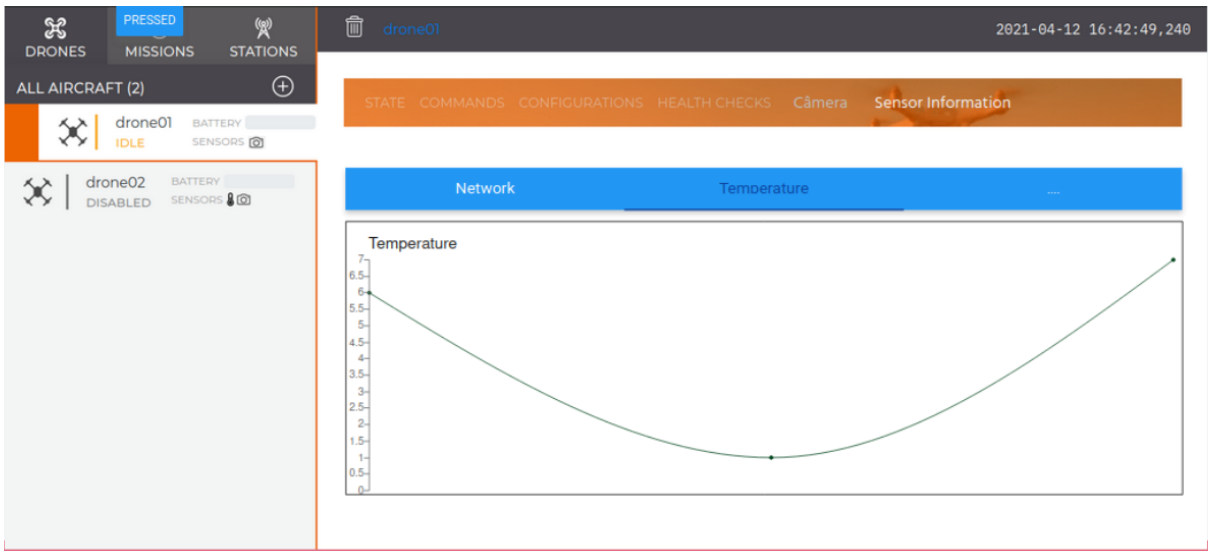


Figure 8: Sensor charts

This figure (8) shows a mockup of the sensors information menu. It’s composed of multiple tabs that will show the information of each sensor throughout the mission, generating charts for this purpose.

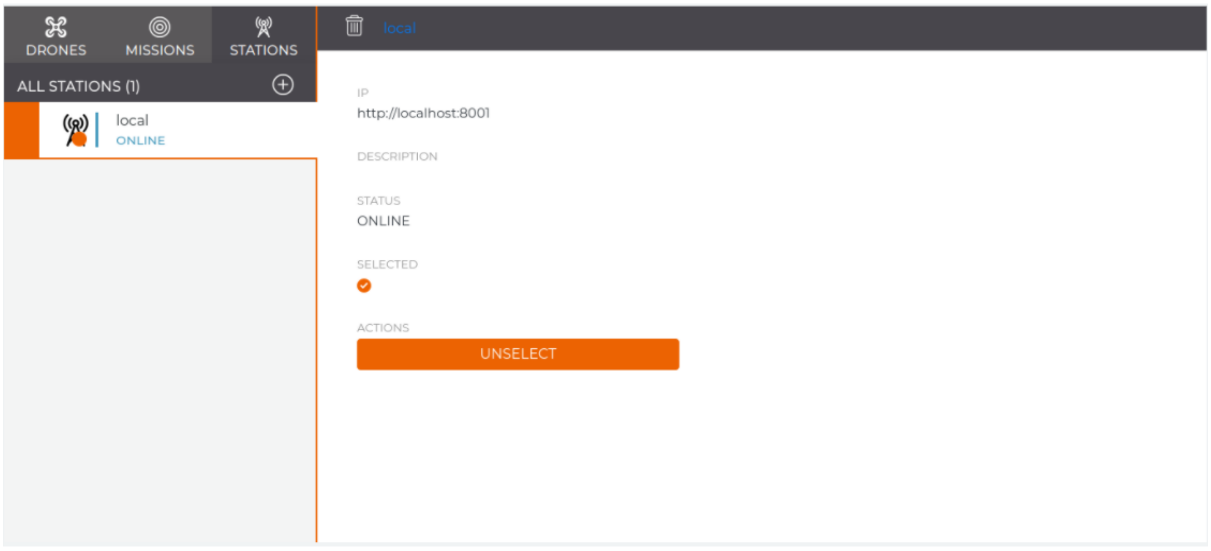
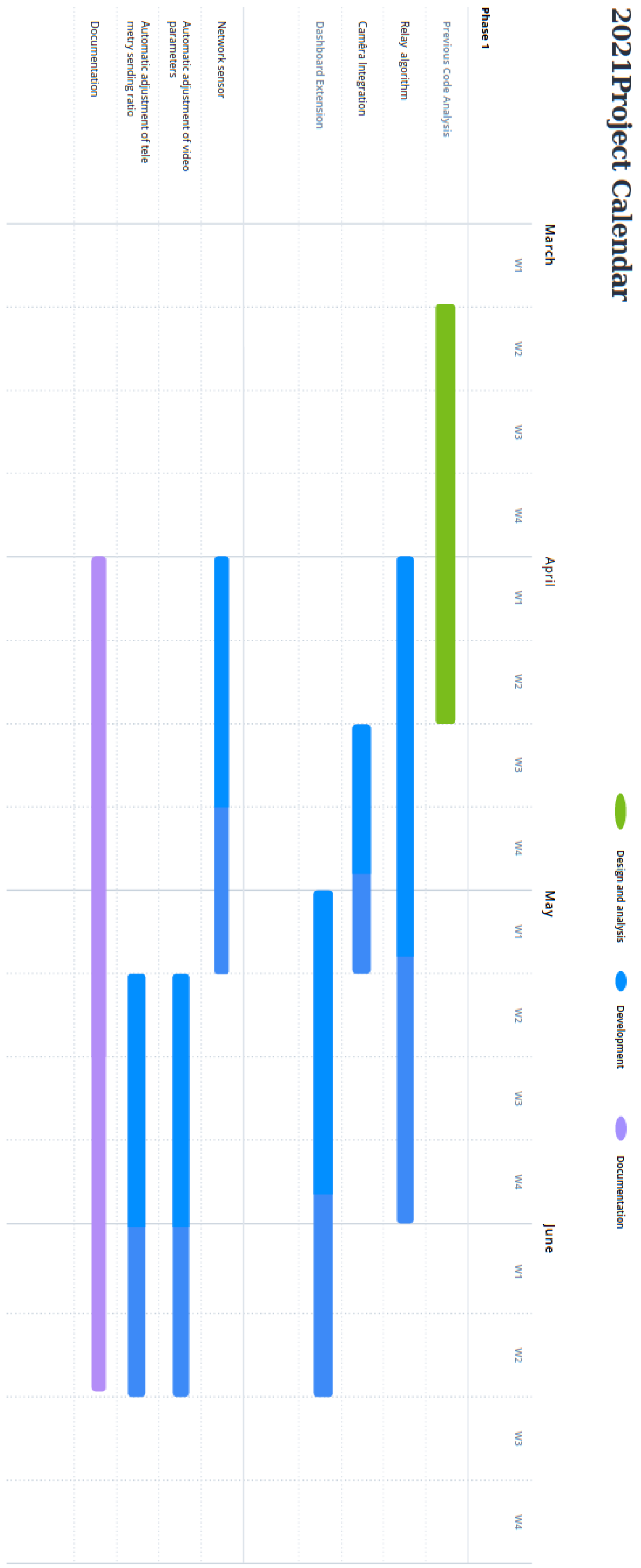


Figure 9: Ground Station Selection

This figure (9) illustrates the insertion and configuration of a ground station that the dashboard uses to communicate. The IP parameter needs to be set to the ground station IP, and the select button selects which ground station is going to be used.

Project Calendar



The above project calendar was designed taking into account the difficulty and the priority of the various tasks that need to be complete in order to make this project successful.
A brief description of the tasks can be seen below:

- **Previous code analysis:** As this project is being developed on a foundation of a mission planning project, an analysis on the work that has already been done plays a fundamental role on the project. This task allows us to better understand and use the technologies needed to develop the other tasks.

- **Relay algorithm:** This task consists in developing an algorithm that takes into account some parameters such as network characteristics will place the real drones in an optimal position.

- **Camera integration:** This task consists in integrating the capture of video footage from a camera placed on the drone to be displayed in multiple devices, using the dashboard that has been previously mentioned.

- **DashBoard Extension:** The objective of this task is to change the dashboard previously developed to add the functionalities and goals described on the section of functional requirements.

- **Network sensor:** The objective of this task is to develop a sensor that will run on the drones capable of monitoring the network characteristics between the drones, the ground station and forward that information to the ground station.

- **Automatic adjustment of video parameters:** This assignment has the goal to transmit video with different qualities automatically, taking into account the quality of the network at the specific time.

- **Automatic adjustment of telemetry sending ratio:** This task consists of transmitting different amounts of telemetry data automatically, taking into account the quality of the network at the specific time.

- **Documentation:** This task will be done throughout almost the entire project as the documentation is an essential tool used to communicate the work done and how the project works with the entities involved in the project.