Drone Guard

Software Engineering B.Sc. Final Project

Software Design Document

Authors:

Yarin Mizrahi | Yali Sofer | Ido Betesh

Supervisor: Dr. Yigal Hoffner & Dr. Amit Resh

17/08/2022

Abstract

Lifeguards' ability to monitor and handle emergencies at the beach is a significant factor affecting the quality of their work, and they are expected to respond quickly to emergencies. The current means available to the lifeguards to carry out their duties are binoculars and a pair of eyes, which in some cases may lead to a situation where they will not be able to distinguish between an emergency situation that requires actual intervention, or a situation where a verbal warning is sufficient. In addition to all of the above, documenting the sequence of events that occur during an emergency is almost impossible given current technology. But such recordings are essential in case of disputes about the actions of the lifeguards and can also help in improving the training of the lifeguards.

DroneGuard is a tool that allows lifeguards to monitor the beaches using a drone in an intuitive way and without any prior knowledge of flying drones. The tool also records actions taken during an emergency, thus enabling a review and analysis of these situations.

Acknowledgements

We would like to thank Dr. Yigal Hoffner and Dr. Amit Resh for their constant nagging, harassing, badgering, cajoling, and other means of Chinese torture applied to us poor souls - throughout the duration of this project.

Table of Contents

1 INTRODUCTION & Problem Description

Today, LG's have binoculars and a megaphone, in order to scan the swimming area, and to communicate with the bathers to ensure they remain within the limits defined as safe. The LG work is demanding and not simple. It requires alertness to what is happening on a large surface area and a real-time response with as little procrastination as possible.

We are trying to give LGs a better way to monitor, zooming-in (navigating) and debriefing events. While doing so we face some challenges. The first difficulty is the fact that in most cases rescuers do not know how to fly a drone because the operations of flying a drone are fundamentally complicated.

There are almost no built-in solutions for simple and intuitive navigation of a drone without prior experience and knowledge out there. A drone can be an excellent tool for LGs. It allows for faster and more efficient monitoring that will allow improved focus ability for the user. Today there are a number of projects and companies that can offer solutions to LG's as drones with cameras and other aids.

1.1 DroneGuard Motivation

We are trying to solve two main problems:

1. The inability to monitor the bathers on the beaches and in the water in a large area, and the long response time to an emergency event. The drone will give us a better and faster way to monitor the shore and the water area with the ability to focus on the event faster and better.

2. An additional problem that comes with emergencies is the ability to replay and analyze them. Since the evidence available to the investigators are subjective, the testimony of LG's or bathers who have been at the scene can be relied upon, but there is no possibility of tracking the sequence of the events.
   In many cases the testimony may be wrong in some parts which may mislead the investigators. Using a digital system that stores data in real-time will allow quality and reliable investigations.

1.2 DroneGuard Goals

The main purpose of the system is to provide LG's with a tool that enables them to monitor the water and beach area, detect any problems, zoom in to swimmers that are in danger or potential danger, and communicate with the relevant swimmers **without** any pre-knowledge about flying drones.

The system will allow the LG's to investigate the videos of the incident in order to understand what happened and learn how to deal with emergency or potential dangers better.

The system will allow the LG's to patrol the beach in a convenient and intuitive way.
In addition to the main goal we have sub-goals regarding the functionality of the product:

- The system will have an intuitive interface for non-technical users
- The system will enable clear communication with the swimmer.
- Real-time display of the drone's view and the ability to move to a specific location by touching a specific area on the screen by the LG's (touch & go).
- In order to perform and implement the drone movement from point A to B, we will calculate the difference between these two points and the transformation we should output to the LG's screen with the minimum deviation possible.
- The system will allow the LG to control the drone using a GUI interface, without the need to know how to fly a drone.
- The system will provide LG a way to communicate with bathers in a more efficient and personal manner.
- The system will function as means of recording and monitoring emergency events for the purposes of interrogation.

1.3 Overview of the DroneGuard Approach

Our product will include:

- A controllable Drone with facilities for monitoring the area and communicating with swimmers.
- An application with a GUI that enables managing the drone easily without knowing how to fly one. We will do this by and the drone will be configured to fly clicking the home button. The LG will touch the real-time display and the drone should move to the selected location (touch & go). This will happen by calculating and converting the image pixel coordinates on the LG's screen to real-world coordinates in terms of direction, acceleration and speed with the minimum deviation.

1. The system will allow the users to control the drone and all its functionality from the web app.
2. The system will include a GUI- drone control and additional features.
3. The system will display all video streams live with no delay.
4. The hardware of the system will include a drone with a video camera, a microprocessor and a GPS.

1.4 Usage scenarios

1.4.1 Scenario 1 -

Yaron, 46, a LG on a beach in Ashdod. On a summer day the beach is crowded with bathers and it is difficult to keep track of all the happenings. Suddenly shouts are heard from the direction of the bathing area.
Yaron turns to the user interface to focus on this area. He navigates the drone to the area and examines the occurrence to determine how to operate. Realizing that this is a joke by a number of teenagers and not an emergency, he descends the height of the drone to watch the event carefully.
Yaron will return the drone to the OP to continue watching the shore.

1.4.2 Scenario 2 -

Avraham, 49 - last week a person drowned at his beach. The person’s family decided to sue Avraham and he needs some evidence that it is not his fault that the person died, he did the best he could. Avraham reaches the DB of the drone, where all footage is kept and uses the footage to protect him in court.
Dekel, 57, is the Vice President of Technological Renewal in the Tel Aviv Municipality. He was recently approached by a number of neighbors from the beach who complained that the rescue calls were interfering with them at noon.
Lifeguards complained that the bathers did not respond to them quickly enough until they realized they had been approached, thus reaching life-threatening situations. He decided to find a new way to help the LG's to monitor the beaches with less noise and public interferens.

1.5 DroneGuard audience

The audience for DroneGuard is the life-guards in the local beaches of the municipal councils and rescue forces.

1.6 Glossary

- **LG -** LifeGuard.
- **Drone** - Unmanned aircraft. Drone is a flying robot that can be remotely controlled or fly autonomously using software-controlled flight plans
- **Base (Docking Station)** - This is a predefined location that will be located next to the LG's cabin. This location is used for the drone to take-off and landing.
- **Observation point** - Observation point is a predefined location which will be the default location for observing the entire monitoring area and from there the LG will be able to zoom in and move to specific locations.
- **Flight-Commands** - After the LG presses on a specific point on the screen, the Translation algorithm will calculate how far the drone should fly to each direction (for example: 10 meters east, 30 meters north) and this flight command will be sent to the drone.
- **Monitor -** One of the three main purposes of our project, Monitoring refers to when the drone zooms in and moves closer to bathers for second look and communication purposes.
- **Navigate -** One of the three main purposes of our project. The navigation will be managed by an algorithm that translates the point touched on the screen to real life coordinates by calculating the pixels coordinates of the touched point with the real life coordinates and height received from the drone to a new real life coordinates and then send flight commands to the drone.
- **Debrief Events (Post Mortem)**- One of the three main purposes of our project. (Figure 2.3)
  When a team carefully deconstructs and analyzes a previous event. A thorough event debrief will help to identify what went right, what went wrong, and what could be better next time.
- **Water area** - The water area inside the predefined area.
- **Beach area** - The Beach area inside the predefined area.
- **Predefined area** - This is the LG's responsibility area and the drone flight borders.
- **Potential danger** - This term refers to any case where the drone descends and zooms in and gets closer in order to check if there is a potential danger.
- **Emergency** - This term is used for an event where bathers' lives are in danger.
- **Latency -** The amount of time it takes from the moment the LG touched the screen until the drone started moving.
- **Flight Commands** - A set of commands that can be sent to the drone in order to navigate him from point A to B e.g. takeoff, move right/left, rotate, etc.
- **Touch & Go -** Touch & Go is the method we use in order to navigate the drone by simply touching a point on the screen.

- **API -** (Acronyms) Application Programming Interface is a connection. between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software.
- **RP -** (Acronyms) is a small single-board computer.
- **OP -** (Acronyms) Observation Point.

2 LITERATURE SURVEY

2.1 Problem Survey

The LG work is demanding and not simple. It requires alertness to what is happening on a large surface area and a real-time response with as little procrastination as possible. We are trying to give LG's a better way to monitor, zooming-in (navigating) and debriefing events. While doing so we face some challenges.

The first difficulty is the fact that in most cases rescuers do not know how to fly a drone because the operations of flying a drone are fundamentally complicated. There are almost no built-in solutions for simple and intuitive navigation of a drone without prior experience and knowledge out there.

A drone can be an excellent tool for LG's. Allows for faster and more efficient monitoring that will allow improved focus ability for the user.
Today there are a number of projects and companies that can offer solutions to LG's as drones with cameras and other aids.

2.2 Solution Survey

Drone as a LG assistant, a way to improve the quality of the LG's work in maintaining order and safety on the beaches, and as a system for investigating safety and emergency incidents. There is a growing understanding around the world that using drones as a rescue tool in many areas and especially at the beach, has many benefits.

Existing Life Saving drones:

- **The Little Ripper** - "Lifeguard drone completes world-first ocean rescue" An Australian company assembled a drone that can carry various rescue equipment such as life jackets, electric defibrillators, survival kits, and more in order to save lives in beach areas.
- **The Pars** - robot is an Iranian drone designed to rescue people from drowning. Pars was developed by Amin Rigi and Amir Tahiri at RTS Lab, a lab working on novel robotic innovations for increasing life safety and living conditions. The drone is capable of carrying up to 3 life-rings. An experiment in the Caspian sea, 75 meters from the shore the drone was able to release life-rings under 30 seconds from its launch, compared to the LG who reached a drowning person within 90 seconds. Drone control, many projects examine the control of drones from different points of view to come up with the most intuitive way to control them. In our project, it is crucial that the drone control be as intuitive as possible so that LG with no pre-knowledge of flying drones can easily handle the GUI and avoid technical obstacles and fly the drone according to his needs.

Intuitive Drones control methods:

- Touchscreen Method of Piloting Drones - Intuitive drone touchscreen control by "finger drags".

Two researchers developed a touch-screen method to navigate and take pictures with drones. The method, called FlyCam, works with the concept of combining the drone and the camera movements. FlyCam uses one- and two-finger drags across a smartphone or tablet to control the drone as it accelerates or turns and takes images. The drone moves forward or backward along the camera's axis with single or double taps to the screen.

- Drone Intuitive Control - describes a way to control a drone through voice and gestures. In this project, the pilot can perform interactive control, which is activated by voice command.

2.3 Discussion and Conclusion

As seen in the previous sections, the use of drones as monitoring and life-saving tools has been gaining momentum in recent years. This is due to the recognition of the benefits that drones are equipped with, from the built-in tool and add-ons to the very different, new and unique angles of monitoring.

It allows us to quickly navigate to different and distant points from each other, whether used in an emergency, which can save valuable time. With all the positive aspects of drone use, the issue of intuition and simplicity in its navigation is still lacking. This is why our project will focus mainly on managing the simple and user-friendly interface combined with easy navigation capability using the touch screen of the tablet device, which is something that we haven't found a proper solution for in the market. This is an issue that is still in its infancy and does not seem to be being emphasized at the moment.

3 TECHNOLOGICAL SURVEY

3.1 Problem Survey

Our Problem Survey exposes several sub-problems. In this section we will describe all of the sub problems:

Drone:
For this project, We need a drone which will be able to carry the weight of the embedded system and the rest of the components which will be added to the drone. We also needed to make sure the drone we bought would allow us to connect the embedded system to send flight commands.

Embedded System:
We need an Embedded System for our project that will allow us to connect components Physically and also support wireless connections. We will need to use this system to calculate some of the transformation algorithms and also for streaming live video to the LG's tablet.

Communication:
We need to communicate wirelessly between the drone and the tablet(The user interface), and also transfer a large amount of data such as the live video stream from the embedded system and drone data to the tablet. Therefore we need to allow this kind of communication.

Navigation:
We need to be able to navigate the drone by calculating the data given from the drone (Altitude, Latitude, Longitude) and the image from the camera to a new real life coordinate and back to screen coordinates. As part of the navigation problem we need to transform the drone navigation control from the remote control to navigating by touching the screen in order to simplify the navigation task.

3.2 Solution Survey

The **Drone**

We have narrowed our options to three different drones which we considered as the best options. We made our decision based on the ability to sync to the drone system, max carry weight, price and max flight distance.

DJI Mavic:

This drone weighs 249g and can carry up to 195g. It has a maximum flight distance of 200 meters and costs 2100 ILS. We didn't choose this drone because after investigation we found out that its sdk is not supported as much as we thought it would be and therefore might end up causing integration problems.

Assembled Drone:

The assembled drone is a self-built drone, it's using an external API with a dedicated OS. It can carry a weight of up to 500 grams and has a maximum flight time of 20 minutes. It costs 2800 ILS. The reason we did not choose this drone in the end was due to the big risk of a one shot project in case we crash it. The second reason we didn't choose this drone is because of the need to deal with a lot of firmware which we are not familiar with and it might be a big time consumer.

DJI Tello:

This is an educational drone, It weighs 80 grams and can carry up to 45 grams. Its max flight distance is 100 meters and it costs 500 ILS. The reason we chose this drone was mostly the ability to use its sdk and use it to communicate and navigate the drone much easier than the other drones, also it gives us the ability to use backup drones in case of a crash.

The **Embedded System**:

- Arduino - This controller makes cyclic operations as long as it's connected to a power source. Its computing power is relatively low. The Arduino has no Operating System although it can easily integrate and connect to other components.
- RP zero WH - Linux based Computer, Its Operating System will be loaded from an SD card. It can connect wirelessly to Bluetooth and Wifi and also has physical connections for different components. It has high computing power and the ability to stream data.

The Communications:

In order to communicate between the App and the Embedded system we will have to consider the fact that we need to use wireless communication since there will be a big distance between the user and the drone when the system will be in use.
We considered three different communication methods:

- Bluetooth - It is a short-range wireless technology standard that is used for data exchange between devices over short distances using Ultra High Frequency radio waves.

- Wifi - A technology that allows data to be transferred between devices connected to it wirelessly via radio waves. The price of the components required to operate it is relatively cheap, and it is quite simple to connect them.

4 REQUIREMENTS AND SPECIFICATION

4.1 DroneGuard functional requirements

Monitoring	
Description/Functionality	Number
The drone will report its location.	3.1.1
The drone will share live stream video.	3.1.2
The system will allow the user to return the drone to its charging station.	3.1.3
The System will allow the LifeGuard to communicate with the bathers.	3.1.4
The drone will have predefined borders of monitoring.	3.1.5
The system will have a predefined Observation Point.	3.1.6

Emergencies situations	
Description/Functionality	Number
The drone will be able to return to its charging station when it recognizes a low battery situation.	3.2.1
The drone should be big enough to withstand the high winds of the sea, and carry the extra weight of the add-ons.	3.2.2
The system will allow the end-user to move the drone to a selected location by clicking on a specific point on the visual picture received from the drone.	3.2.3
The system will allow the user to control the drone on the horizontal & vertical axes.	3.2.4
The system will prevent the user from descending too low (lower than 4 meters) to avoid collision with bathers or other obstacles.	3.2.5
The application will enable the user to turn on the external audio devices.	3.2.6
The system will send coordinates/directions according to the area the user clicked on the screen.	3.2.7

Post Mortem	
Description/Functionality	Number
The system will allow the user to view all previous drone videos.	3.3.1
The system will allow the user to crop, edit and delete the videos.	3.3.2
The system will allow the user to have access to the Post Mortem videos at any time.	3.3.3
The system will save the drone footage automatically to the DB.	3.3.4

General	
Description/Functionality	Number
The system will allow the LG to login and identify himself.	3.3.1
The system will display statuses such as battery percentage and drone altitude.	3.3.2
The drone will report its location.	3.3.3
The system will save the drone footage automatically to the DB.	3.3.4

4.2 DroneGuard non-Functional requirements

Application	
Description/Functionality	Number
The data will be stored in the database for a period of time selected by the user.(With a time limitation of 90 days).	3.1.1

Drone	
Description/Functionality	Number
The drone will have the capability of carrying the add-ons equipment weight.	3.2.1
A camera, RP and battery will be installed on the drone.	3.2.2

Hardware	
Description/Functionality	Number
Video streams will have minimal latency.	3.3.1
The drone will include a GPS sensor that indicates its location.	3.3.2

4.3 Specification - System scope

Our system scope will include three main sections, Monitoring, Navigation and Post Mortem.

In the **monitoring section**, the scope of our system is to be able to monitor an initial predefined area using the drone and the ability to zoom in and out of specific locations within the pre-defined monitoring area.
The Monitoring will be done by flying the drone to a specific point the LG wants to examine by touching that area on the touchscreen.

The **Navigation section** relates to the actions that the drone should perform, such as moving in any direction, descending in order to zoom in a specific area requested by the LG (under a predefined limit), and more. All of that should be fluent in terms of transformations from the real-world coordinates to the pixels on the screen display and back.

The **Post Mortem section** has the ability to interrogate a recorded event if needed. The system will allow LG to access all of the drone videos and the ability to comment or delete them. Any LG will be able to see only his Beach records.

5 THE ARCHITECTURE

The DroneGuard system consists of three main entities:

1. **Drone** - attached with RP to which a camera, GPS, and audio devices are connected.
2. **Lifeguard** - has a tablet and can control and navigate the drone.
3. **Investigators** - have access to the DB with the drone footage and can debrief when necessary.

5.1 Roles

- LifeGuard - Navigates the drone with the DroneGuard application.
- Investigator - Investigate drone footage with the Debriefing System.

5.2 System Architecture and Drone Control

Architectural Description and Design: Activities and Data

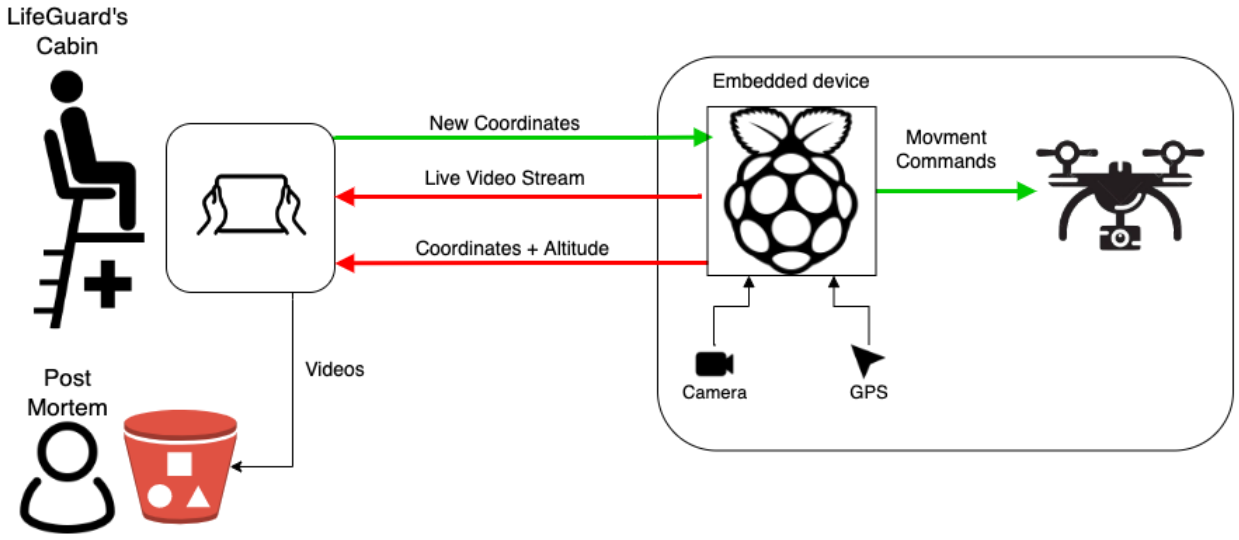

Figure 5.1: system architecture.

The system has three major parts:

1. The **Drone** with the microcontroller (RP) that controls it and also is responsible for the camera and the GPS controls, it also sends the commands to the drone.
2. The **LifeGuard system** consists of a Tablet that communicates with the RP and sends the data required to perform the relevant actions (takeoff, land, navigate..).
3. The **Post-Mortem system** will enable the LG to investigate videos that were recorded from the flight.

For the use of our system we had to add hardware components in order to control the drone.

- The RP (described as an embedded device in the diagram) is located on top of the drone responsible for several core functionalities:
    - **Video Stream** - Transmits a live video stream from the camera that is connected to the RP.
    - **GPS component** - Gets the Drone coordinates location and transmits them to the RP.
    - **Calculate drone commands** - gets the required movements from the application, calculates the commands, and sends them to the droneand as well

- When we had to connect the hardware all together we encountered communication issues since the Drone and the RP were able to connect only to **one** WiFI network. This caused some problems because as soon as we connected the drone to the WIFI network we were unable to connect it to LG's Tablet. In order to solve this issue we used a Wifi Range Extender which connected all of the components to it and this way we could connect all the components together.

- The LG's Tablet connects to the Range Extender(explained in section 7.2.3) and communicates with the RP.

5.2.1 Activities

- LifeGuard - Operated the drone from the laptop in the LifeGuards Cabin. The LG will monitor the beach area and the bathers.
  The LG can takeoff and land the drone in the required location.
  The LG moves the drone by indicating a specific point on the screen by clicking it.
  The LG is able to fix the drone position by arrow keys and navigate to another point in the same manner.
  The LG will be able to take off and land the drone, navigate it to the desired location, ascend and descend when approaching the target.

- Investigator - Has a separated system there can watch recorded flights, investigate them, comment and delete them.

5.3 Post-Mortem system

Once the drone lands on the ground, The video from the flight will be saved automatically to the RP. Later on the video will be uploaded to aws S3 bucket and will be available to watch on the Debriefing System.

Lifeguards will be able to log in to the Debriefing system and watch the videos which were taken by them and also comment on them only in their beach.

Users with Admin permission will also be able to edit or delete videos and also have access to watch videos taken by all of the LG's.

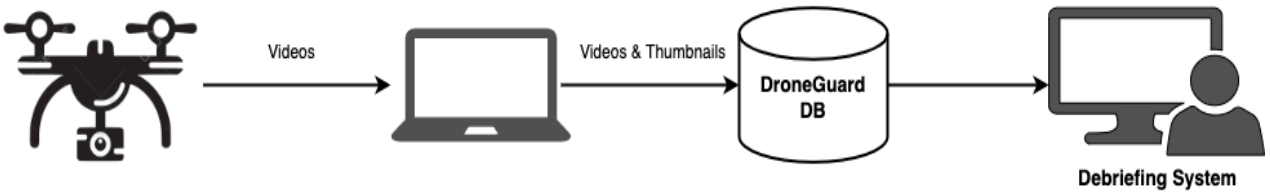

Figure 5.2: debriefing diagram.

5.4 The life-cycle view of a systems

5.4.1 Navigation & Monitoring system

Diagram 5.4.1 shows the life-cycle of the navigation flow, from the docking station to the desired observation point.

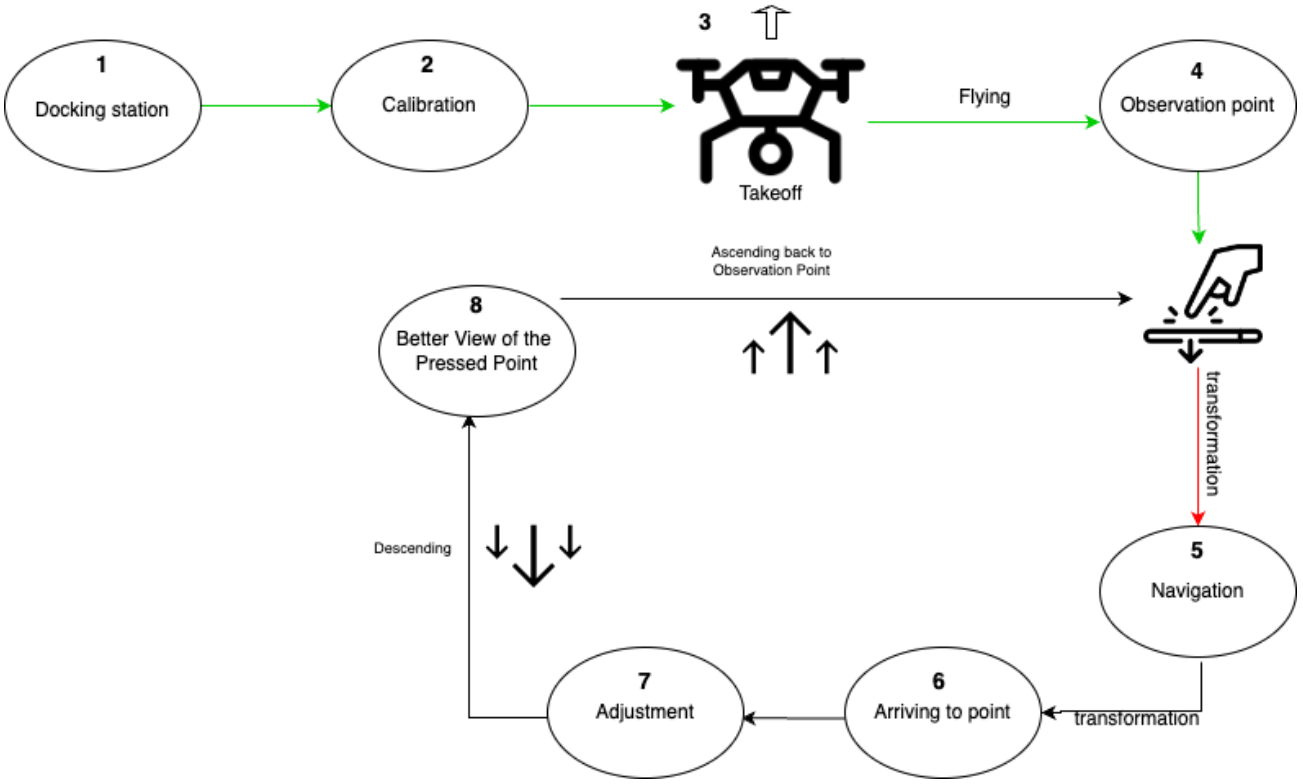

Figure 5.4.1: navigation flow diagram.

The Navigation-flow & Monitoring-flow steps shown in Figure 5.4.1:

1. Initial drone location in Docking station.
2. Initial system calibration.
3. Drone takes off from the docking station by the LG command.
4. Drone arrives at the observation point and stays there.
5. LG watches the beach, touches the point on the screen in order to move there. For this action to take place, we need to calculate the transformation from the real-world coordinates to display pixels and vice versa. (See details regarding transformations calculation at Algorithmic description).
6. The transformations get calculated ⇒ the drone navigates itself to the desired point.

7. LG decides whether the drone is in the correct position or it needs some adjustments.
8. When reaching the point in the LG ability to descend, then the LG can choose another point in the screen return to step 5.

5.4.2 Debriefing system

Debriefing flow steps:

- Debriefing system landing page.
- Scroll between all recordings.
- Click specific recording.
- User options are to play/pause the recording, comment on it and delete.
- Users can either export drone logs containing data such as drone height, battery level, GPS coordinates, temperature, etc from the specific recording.
- Go back in order to watch all the recordings.

6 Mathematical Basis: Real -> Virtual -> Real Transformations

In order to deal with our main issue, the transformation from pixels to the real world and vice versa have to deal with:

The ability to convert the image obtained from the camera and display it on a screen in a pixel view, so that indicating a location on the screen can be converted back to meters in the real world. The relationship between the dimensions of the real world and the view in pixels is maintained.

In this section, we describe the relationships between the real and virtual views and how to convert from one to the other.

Technical concepts:

- **Focal Length** - the distance between the lens and the image sensor when the subject is in focus, usually stated in millimeters.
  determines the optical feature of a camera lens.
  Our camera Focal length is 3.29mm.
- **Field of view (FOV)** - the maximum area of a sample that a camera can image. It is related to two things, the focal length of the lens and the sensor size.
  Objects that are outside the FOV range are not documented in the photo.
  Our camera FOV is 72.4 degrees.
- **Sensor Image Area** - determines how many pixels are captured in the camera lens.
  Our camera's Sensor Image Area is 3.63 X 2.72mm.
- **Drone Orientation -** We will keep the drone orientation facing North. This way will also assist us because when our drone is statically facing north we can always assume that [**North = Forward**, **East = Right**, **South = Backwards**, **West = Left**].
  We get the drone's current direction from a compass component that is connected to the RP on top of the drone so that we can always rotate it northwards by default.
  While the drone

  Navigation Discussion:
  In order for the navigation to work as expected when the LG moves the drone right, left forward, or backward, we must set its default orientation northward. To do so, we make sure that after every touch on the screen by the LG the drone sets its orientation north before it executes its next navigation command.
  By the time of the reset northward, the LG is restricted from touching the tablet until the drone faces north. Therefore we let the LG be concern-free when it comes to compass directions.

  The following are the markings of the variables for the purpose of the calculations:

  Virtual world parameters(**pixels**) -

  - L_S - tablet screen length in pixels (pixels length)
  - W_S - tablet screen width in pixels (pixels width)

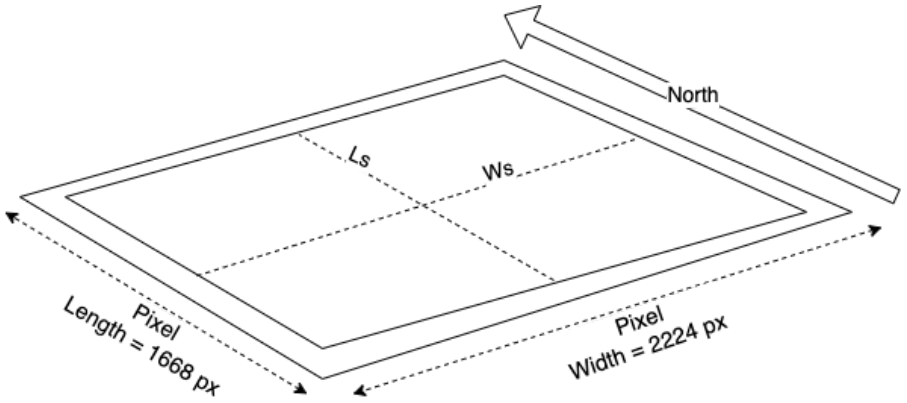

Figure 6.1.1: virtual world diagram(screen).

Real world parameters(meters) -

- **h** - Drone height (from surface),
  We get drone height from the built-in barometer.
- W_R - Distance (real width)
- L_R - Distance (real length)

Camera's parameters -

- **f** - focal length
- Sen_L - Sensor image area (length)
- Sen_W - Sensor image area (width)
- **α** - Horizontal lens degree
- - Vertical lens degree

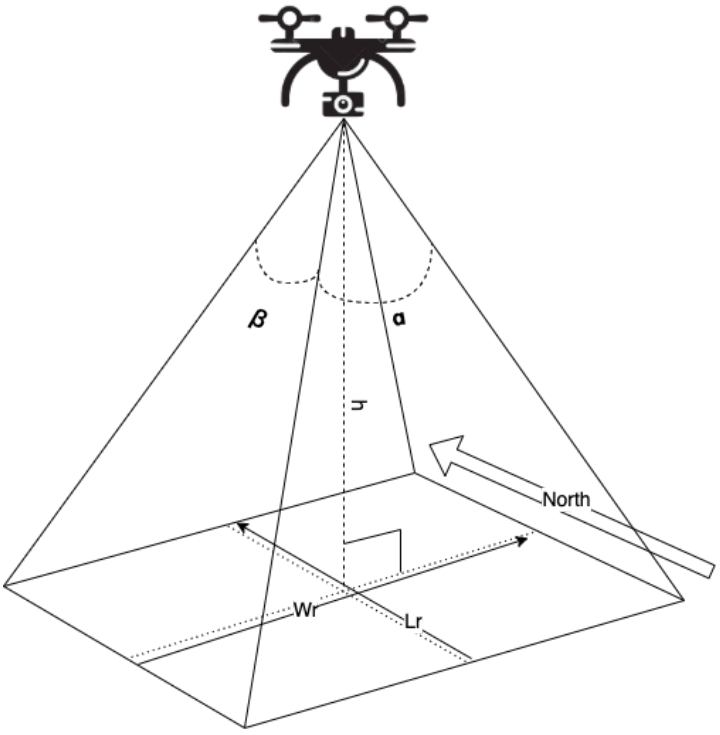

Figure 6.1.2: real world diagram.

Real to Virtual transformation Calculations :

- First We will calculate our α & β lens degrees from our basic given camera's parameters(f, Senw,Senl).

$$\text{Width: } \alpha = 2 \cdot arctan\left(\frac{Senw}{2 \cdot f}\right) \Rightarrow 2 \cdot arctan\left(\frac{3.68}{3.6 \cdot 2}\right) \approx 54.14°$$

$$\text{Length: } \beta = 2 \cdot arctan\left(\frac{Senl}{2 \cdot f}\right) \Rightarrow 2 \cdot arctan\left(\frac{2.76}{3.6 \cdot 2}\right) \approx 41.94°$$

- We use the α & β degrees and the drone height (h) in order to calculate the size of both Width and Length captured **real** dimensions in meters (real world width - W_R & real world length - L_R):

W_R - Real World Width: $tan\left(\frac{\alpha}{2}\right) = \frac{W_R}{2h} \Rightarrow W_R = 2h \cdot tan\left(\frac{\alpha}{2}\right)$

L_R - Real World Length: $tan\left(\frac{\beta}{2}\right) = \frac{L_R}{2h} \Rightarrow L_R = 2h \cdot tan\left(\frac{\beta}{2}\right)$

Example -

$W_R = 2 \cdot 10 \cdot tan\left(\frac{54.14}{2}\right)$ = 10m~

$L_R = 2 \cdot 10 \cdot tan\left(\frac{41.94}{2}\right)$ = 7.5m~

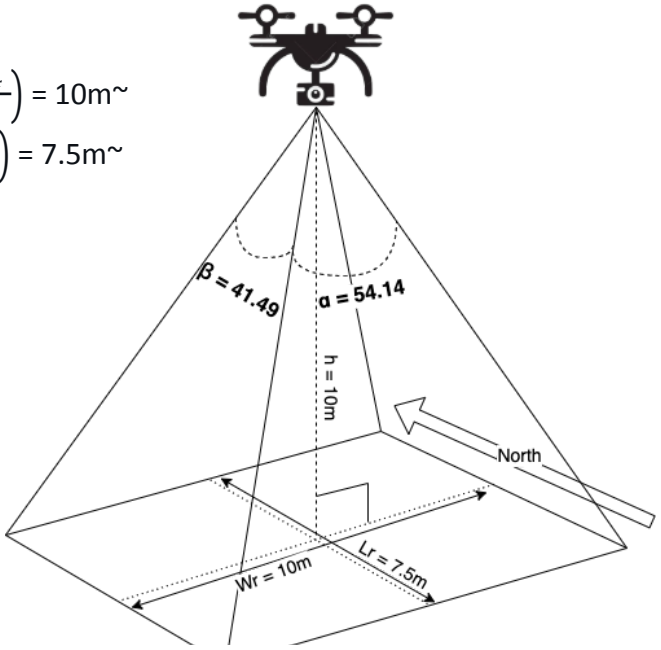

Figure 6.1.3: real-world diagram(values) .

- Now we will calculate the pixels to meter ratio using W_R, W_S, L_S, L_R from previous calculations as follows:

con_W - Width Conversion (width ratio of pixels to meter): $\frac{W_R}{W_S}$

con_L - Length Conversion (height ratio of pixels to meter): $\frac{L_R}{L_S}$

Example -

$con_W - Width\ Conversion = \frac{10}{2224} = 0.0045$

$con_L - Length\ Conversion = \frac{7.5}{1668} = 0.0045$

Virtual to Real transformation Calculations :

- Now that we know the number of pixels per meter, all that is left is to calculate the distance between the middle point and the selected point(screen touch) by using two variables that represent the real distance from the center of the screen to the destination point in meters.- which are the real distance from the center of the

screen to the destination point in meters. To do that we can either use the Pythagorean theorem of subtraction between the points with attention to the four cardinal directions.
And then we will get the real world distance from the screen distance.

New_point = The point on the screen where the LG pressed on.

Middle_point = The middle of the screen. $(\frac{W_S}{2}, \frac{L_S}{2})$

- **Move_x** and **Move_y -** they are the real distance from the center of the screen to the destination point in meters on both axes.

$$Move\_x = (New\_point - Middle\_point) * con_W$$

$$Move\_y = (New\_point - Middle\_point) * con_L$$

Example -

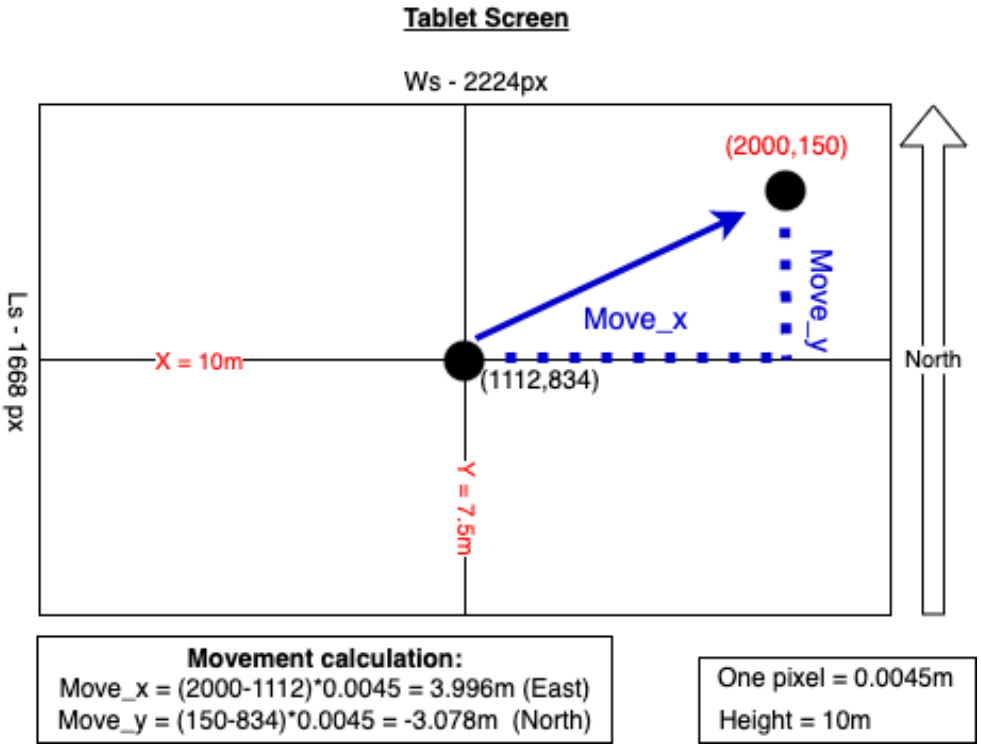

Figure 6.1.4: Movement calculation example

Illustration - Coordinates Transformation
Step I: LG touches the screen in order to navigate the drone to a specific point.
Step II: coordinates transformation calculation by the server (which runs on the RP), commands are sent to the drone, and the drone reaches the point.

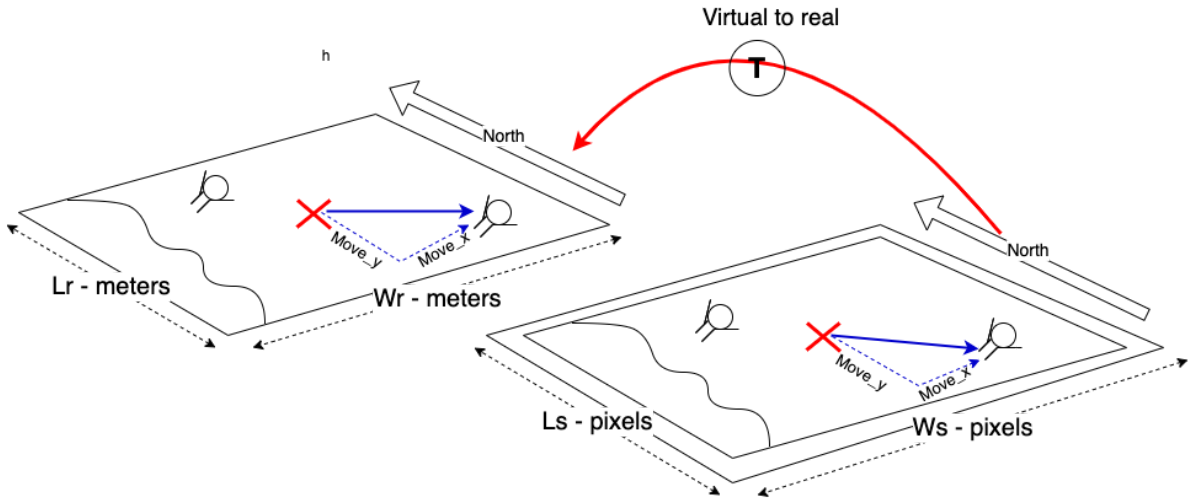

Figure 6.1.5: Virtual to Real transformation.

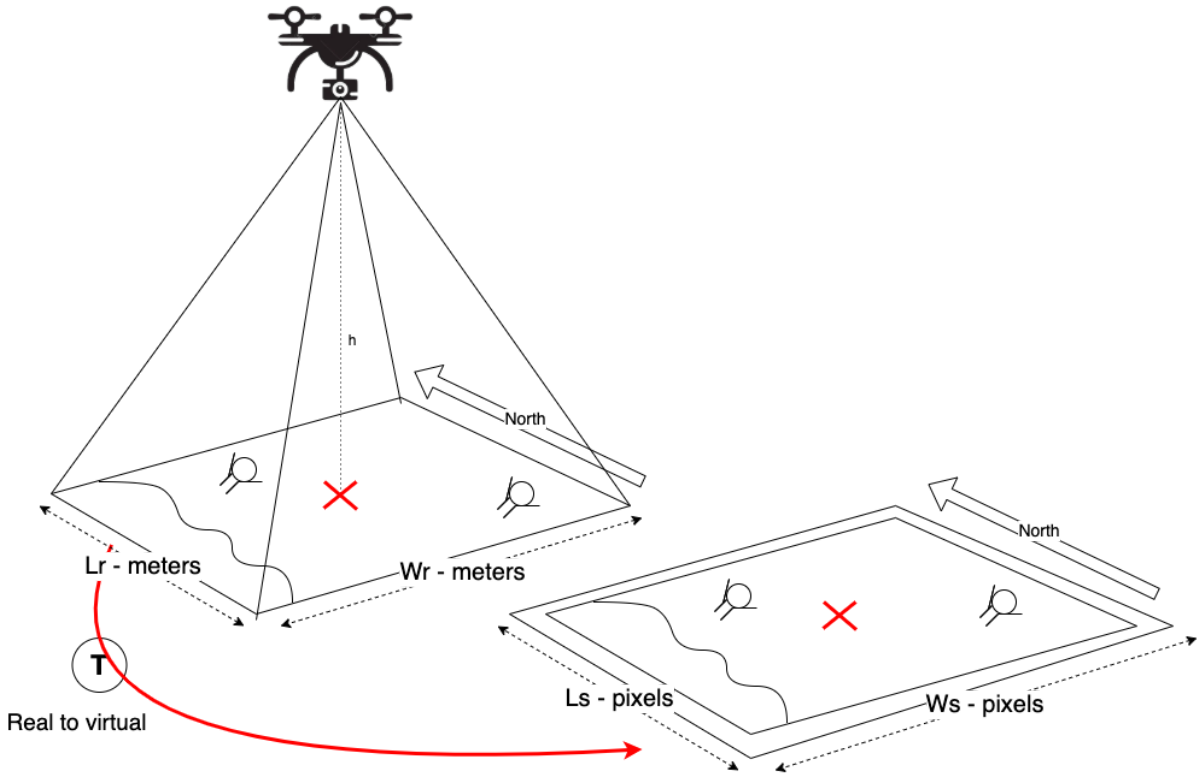

FIgure 6.1.6: Real to Virtual transformation.

Real to Virtual transformation algorithmic overview:

$$\alpha = 2 \cdot arctan\left(\frac{Sw}{2 \cdot f}\right)$$

$$\beta = 2 \cdot arctan\left(\frac{Sl}{2 \cdot f}\right)$$

$$W_R = 2h \cdot tan\left(\frac{\alpha}{2}\right)$$

$$L_R = 2h \cdot tan\left(\frac{\beta}{2}\right)$$

$$con_W = \frac{W_R}{W_S}$$

$$con_L = \frac{L_R}{L_S}$$

Virtual to Real transformation algorithmic overview:

middle point = $(\frac{W_S}{2}, \frac{L_S}{2})$

$$Move\_x = (New\_point - Middle\_point) * con_W$$

$$Move\_y = (New\_point - Middle\_point) * con_L$$

7 SYSTEM DESIGN

7.1 System configuration and components

The components of the system are the drone, the RP with the attached camera and GPS module. The RP has the control server(responsible for communicating with the drone) and the camera server(controls the management of the camera).

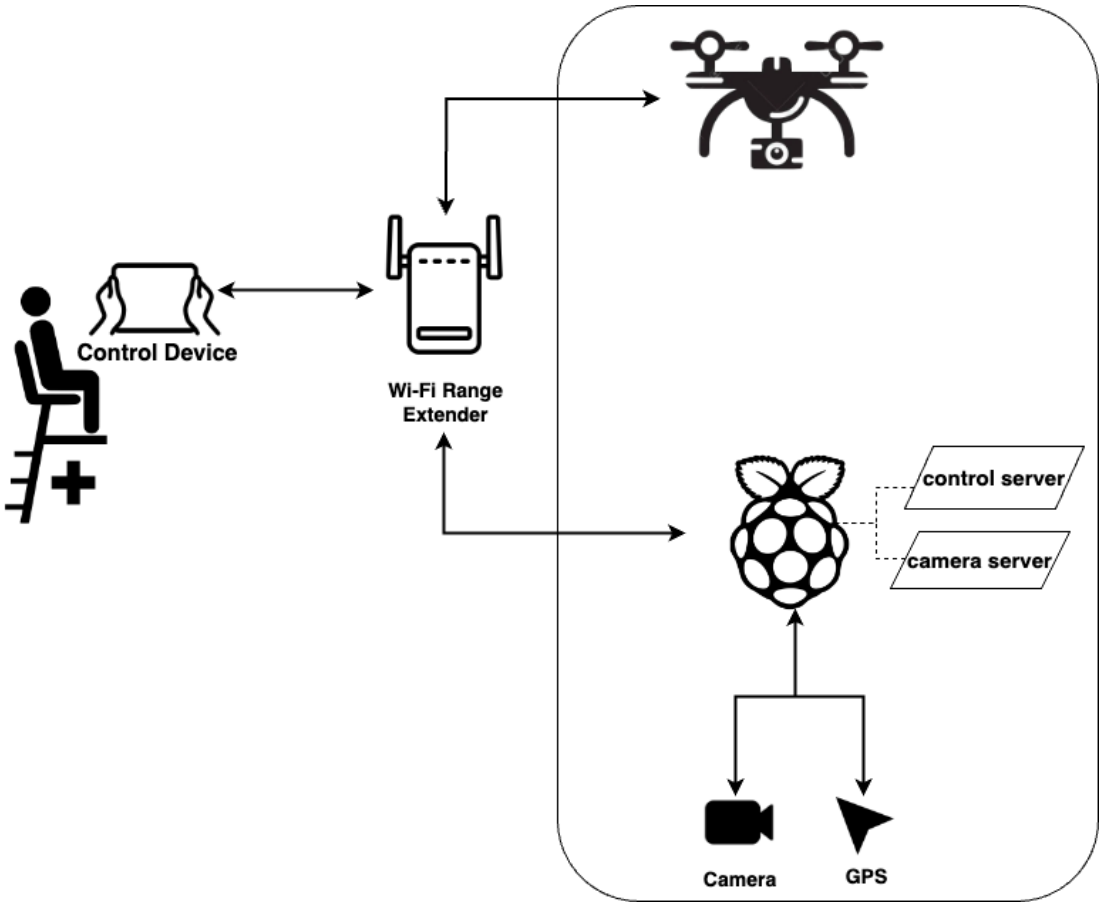

Figure 7.2.3 System configuration showing the components of the system and connection between them.

Range Extender - In order to have two main functionalities in the system:

1. Extend the signal from the laptop that runs the LG app to the drone that receives the commands.

2. Our drone has a limitation so it could only connect to one device that can interact. Since we needed to connect both the RP and the LG laptop with the control app we needed to figure out a way to include them both - we managed to achieve that by using a range extender that creates an access point to which all components connect.

On drone booting, we configured it to auto-connect the range extender network then the rest of the components are connected to the same network as well and can communicate with each other.

Camera - We used the **Mini camera for Pi Zero** as our camera for the drone.
We chose a different camera(not the built in one in the drone) because the angle of the one in the drone did not fit our requirement and it could not be moved.
The Mini camera was light weight with easy to use integration and this is why we chose it as our camera for the project.
We connected the camera to the RP in the angle that fitted our needs, due to the fact that the camera was connected to the RP we could easly records the video session and control the on\off operations.

GPS - We chose the **GT-U7 GPS Module** because it was the most light weighted we found, it was also RP compatible and reasonably fair to calibrate and get signals from. We connected the GPS to the RP in a manner we could get the information that we needed.

Software - on both navigation and post-mortem apps we used NodeJS for the backend and ReactJS for the frontend.
Whereas the post-mortem app communicates in a "regular" way - over HTTP requests and basic API, the navigation app communicates over web socket in UDP protocol - the reason to do so is that the navigation commands should be transferred fast as possible to the drone so the app is capable of sending and receiving data in real-time from and to the drone.
Video stream - In order to get the video stream with the minimum delay possible we examined several options:

- Sending the stream with FFMPG library.
- Separate Node server responsible only for the video streaming.
- Separate Python server responsible only for the video streaming.

Eventually, we decided to stream the video using a separate Python server which has the least delay time and is easy to integrate with the navigation app.

7.2 Data components

We used a non-relational database (MongoDB) since it is free and easy to integrate with our system.

DroneGuard does not have a complex data model.
With LG's main application, data management is pretty straightforward and involves three entities:

User - LG and Investigator credentials.


```
_id: ObjectId("62081047e28f2cdddf4e094b")
email: "qwe@gmail.com"
password: "$2a$10$sC.FPx9fcMwWRA4Ho8N0qOE6A7ZOlWre2OFqUxpgkJkCvhdQJaz8W"
name: "tomer"
userType: "Lifeguard"
createdAt: 2022-02-12T19:53:43.469+00:00
updatedAt: 2022-02-12T19:53:43.469+00:00
__v: 0
```


Beach - Beaches and their details, helps to distinguish between the different LGs work places, whereas each LG is capable of watching only records from his shifts(specific beach).


```
_id: ObjectId("62190b1caa4957bc5319e0dc")
name: "Gordon"
city: "Tel Aviv"
__v: 0
createdAt: 2022-02-25T17:00:12.567+00:00
updatedAt: 2022-02-25T17:00:12.567+00:00
```


Record - Contains the URL and thumbnail of the recording as well as the beach on which it was recorded.


```
_id: ObjectId("6286199c8fab51b36b21c823")
url: "https://drone-guard-debriefing.s3.eu-west-1.amazonaws.com/test_0705202..."
user: ObjectId("620b9e2daa47cb38f0040357")
beach: "Frishman"
thumbnailUrl: "https://drone-guard-debriefing.s3.eu-west-1.amazonaws.com/test_0705202..."
createdAt: 2022-05-19T10:19:08.833+00:00
updatedAt: 2022-05-19T10:19:08.833+00:00
__v: 0
```

7.3 Interactions

7.3.1 Debriefing system - the user can use the debriefing system to investigate and re-watch an event in a post-mortem way with the following actions: the user can watch a video only from his beach, add a comment to the video or delete it.

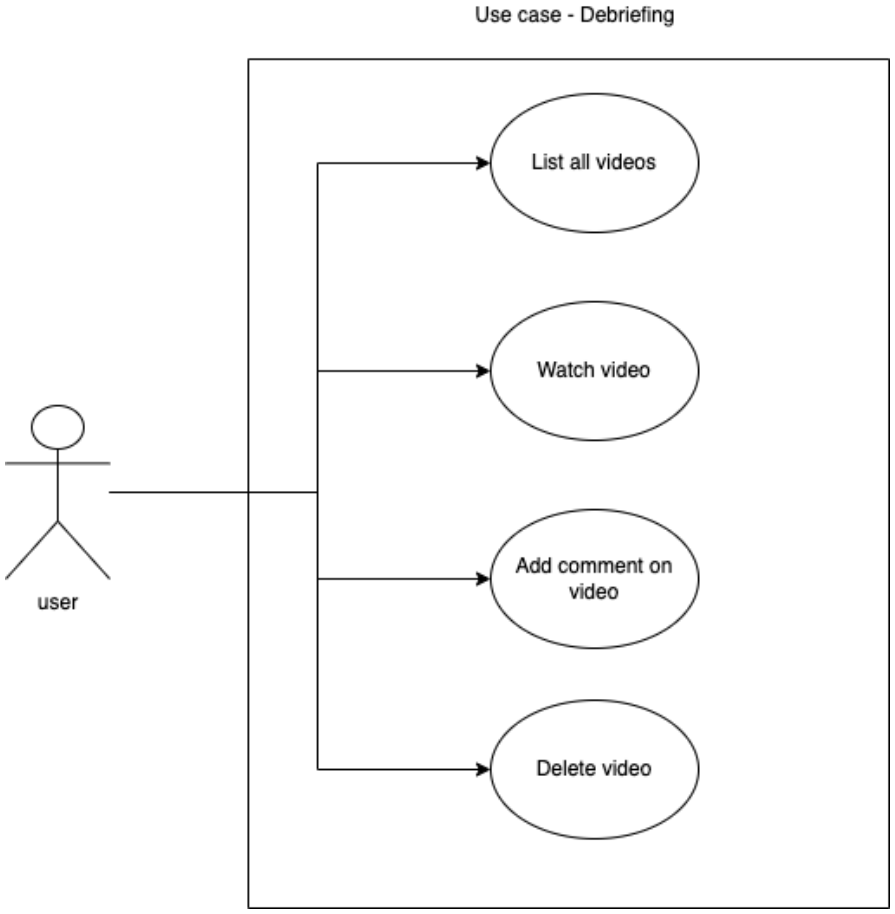

Figure 7.3.1: use-case debriefing diagram.

7.3.2 The Post-Mortem system - The flow of the debriefing system starts with seeing all the videos available to the user from his beach in the main page, the can choose a specific video the can watch it or move to other video..

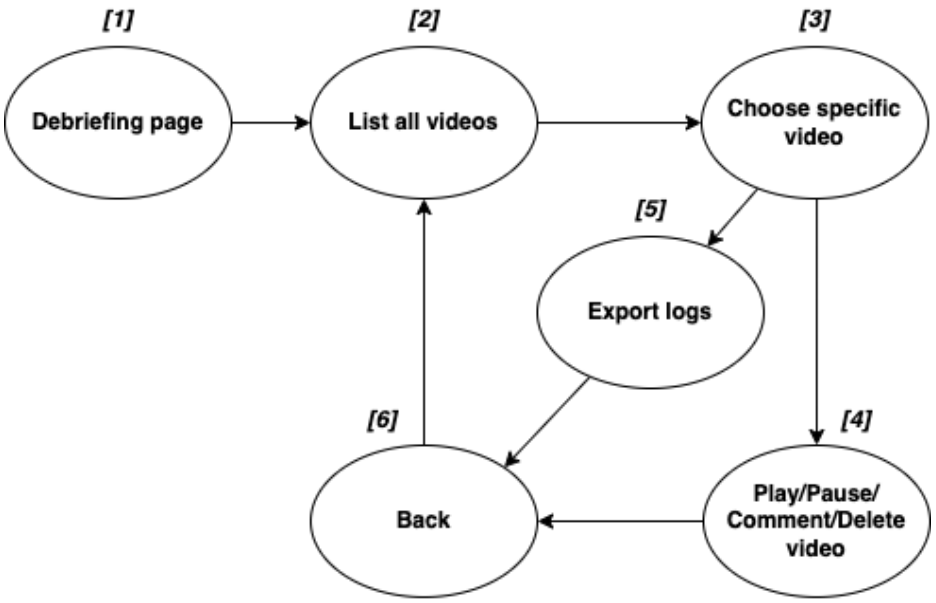

Figure 7.3.2: debriefing flow diagram.

7.3.3 Monitoring and navigating systems - the user can do several things in the manner of navigating and monitoring, he can press on the screen to move to a specific point, he can adjust the location of the drone.

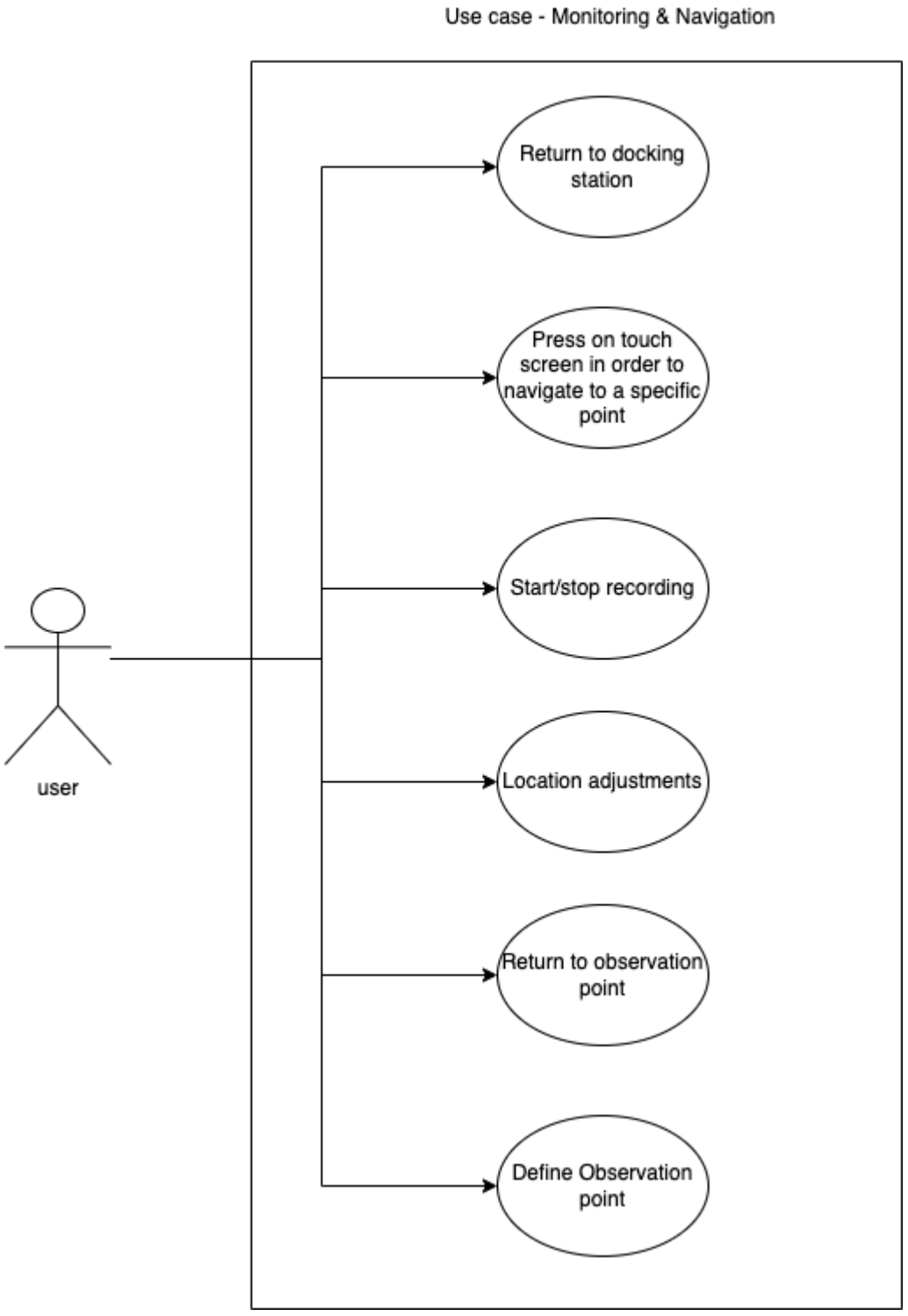

Figure 7.3.2: use-case monitoring and navigation diagram.

7.3.4 State diagram

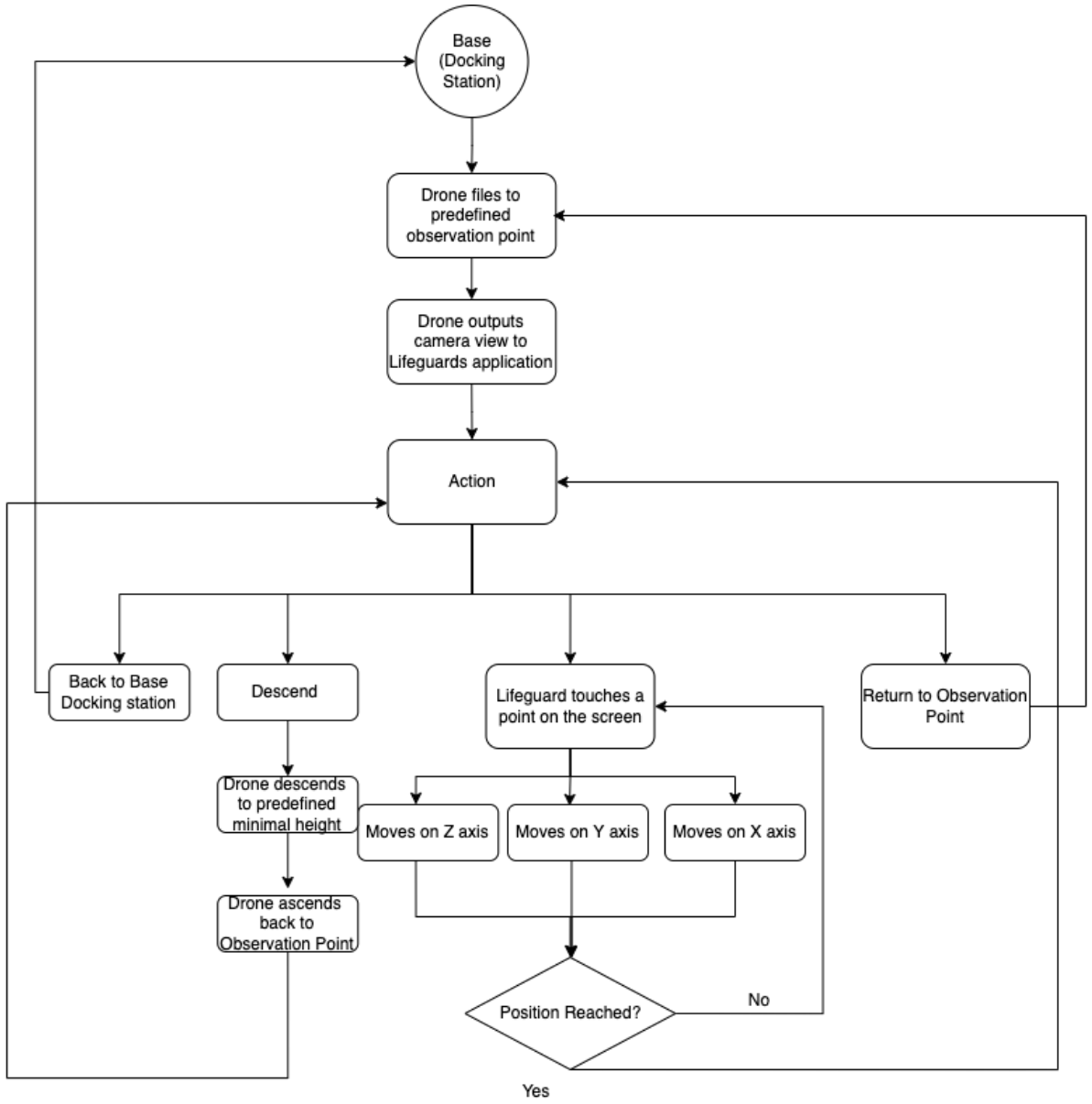

Figure 7.2.1.1: system state diagram.

7.3.5 Navigation sequence diagram

The navigation sequence diagram shows what happens when the system is turning on and when the LG is pressing a point in the screen and wants to navigate there.

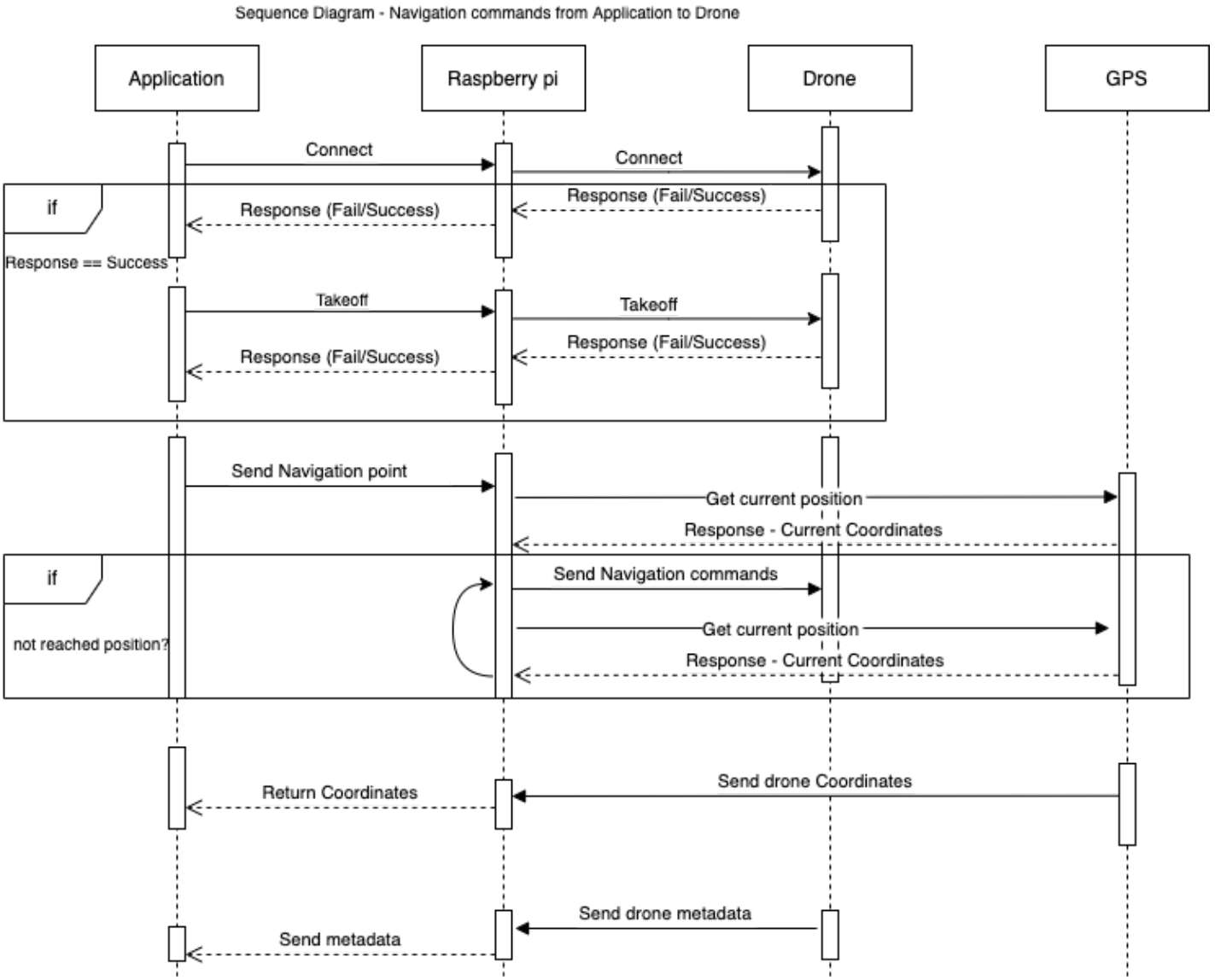

Figure 7.2.2.1: navigation sequence diagram.

Application diagram shows what happens when connecting to the camera and to the raspberry pi when connecting.

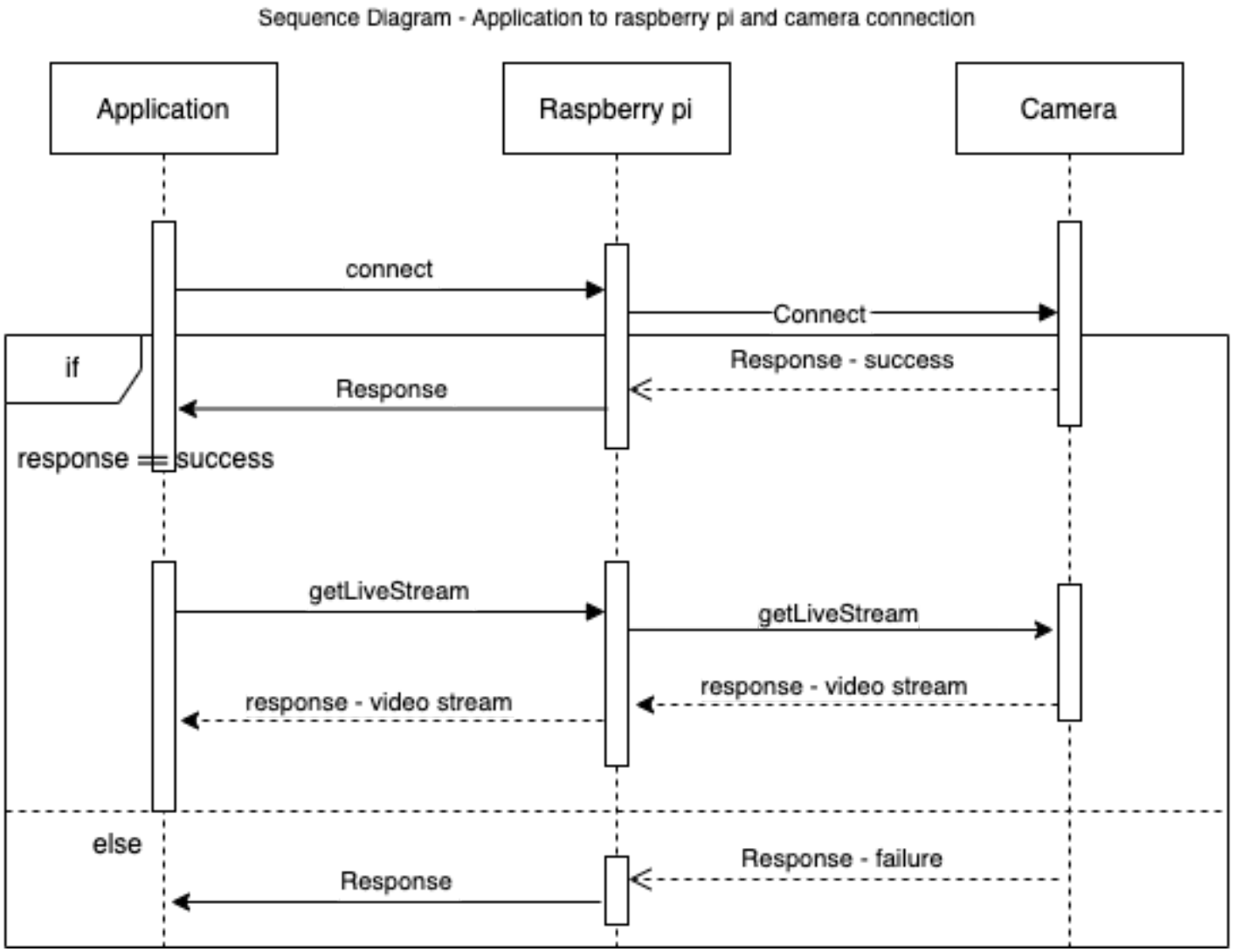

Figure 7.2.2.2: application sequence diagram.

8 IMPLEMENTATION

8.1 Interfaces (GUI)

8.1.1 LifeGuard Application

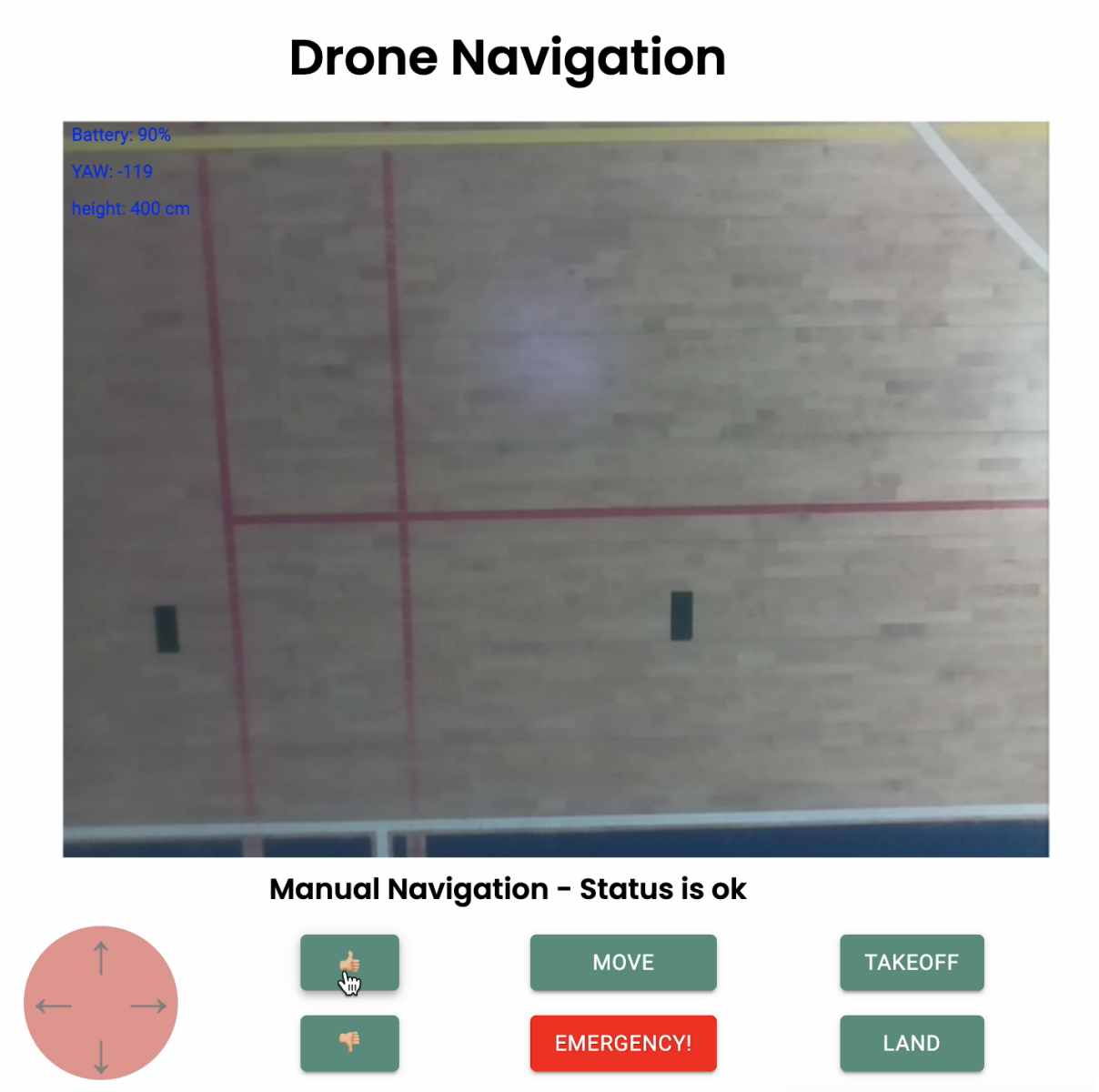

Figure 8.1.1: navigation application.

8.1.2 Debriefing System

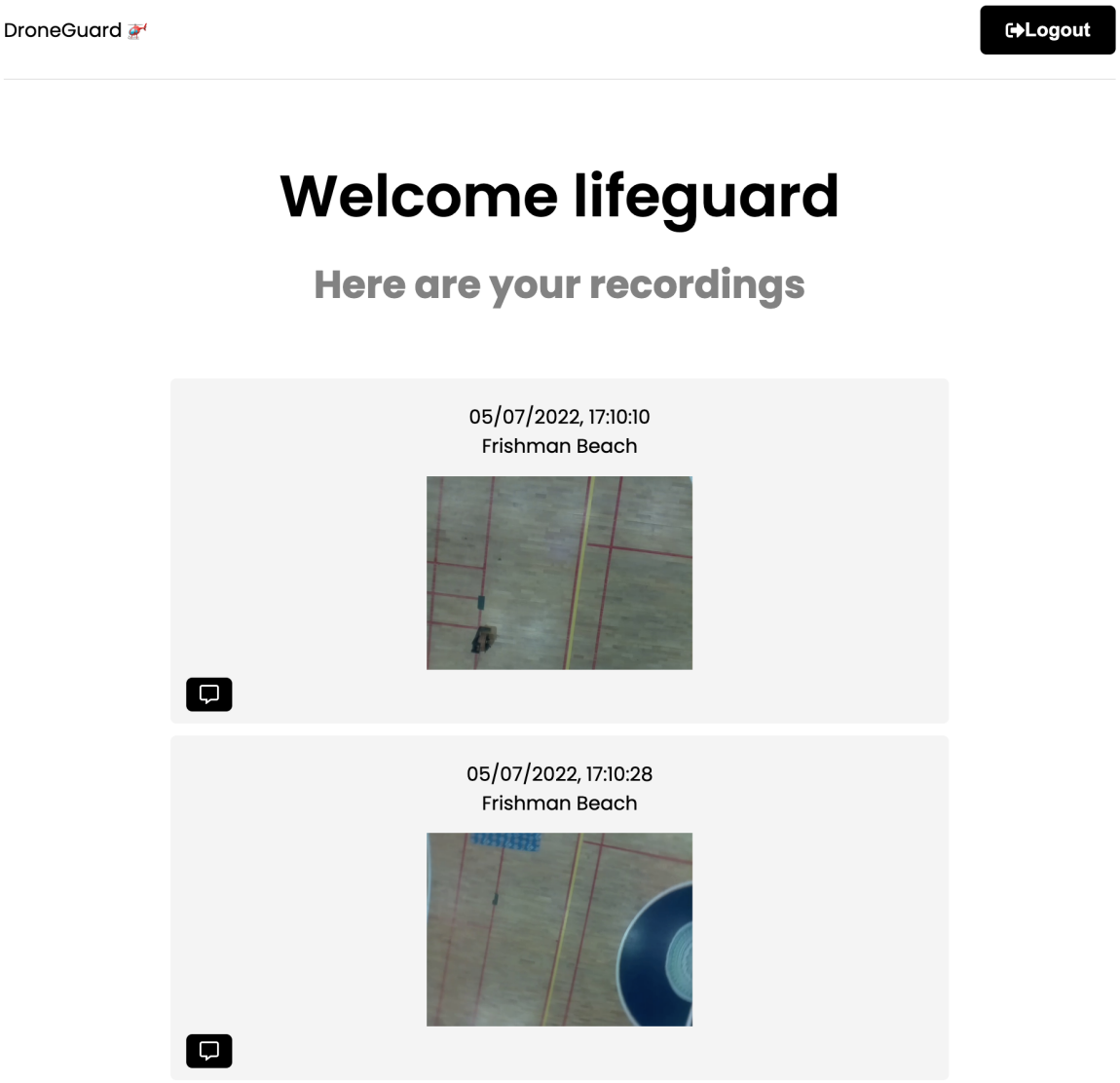

Figure 8.1.2: debriefing application.

8.2 Drone

Figure 8.2.1: drone back side.

Figure 8.2.2: drone front side.

8.3 Development environment

The DroneGuard project was developed in the Visual Studio Code environment. Visual Studio Code is a cross-platform IDE, providing an editor for several programming languages such as Python, React and JavaScript with on-the-fly code analysis, error prevention and automated refactoring for JavaScript code, supporting AWS connection.

8.4 Programming languages

In our project, we used mostly NodeJs for the core services. The UI we did with React. We also used Python for the camera streaming. In order to automate DroneGuard flows, we used Bash scripts.

8.5 Limitations of the system

Due to hardware limitations the drone system has some physical limitations.
We can only fly the drone when there is no wind at all. Due to the fact that the drone is lightweight, and we are putting a lot of weight on the drone, whenever there is wind the drone starts to spin in the air, and sometime might even now respond to the controller.
The use of this minimalistic drone limits the flying time to couple of minutes each time before the drone lands automatically, with the heavy weight of the drone the battery runs much faster than usual.
Another limitation is the use of wifi to connect all the systems component, we are using an extender for the wifi, the extender must be plugged into a plug in order to work, so the user is limited to use the system only when there is a plug nearby..
Another weight limitation is that while all the functionality works we cannot fly the drone with the gsp module due to being overweight that's not allowing the drone to take off from the ground.

8.6 Installation and configuration issues

During the assembling of the Drone and the RP, we bought a compass component in order to receive the drone heading direction at all times. While we tried to connect and configure the compass we encountered an issue which eventually caused us to move on without the compass.

8.7 Data Security

Data is a crucial part of many systems and it is better to be on the safe side when handling it. In your system, DroneGuard, we can ensure data security in three sensitive points of the system:

1. **Data encryption** - helps protect private information, sensitive data, and can enhance the security of communication between our client app and the server. In essence, when our data is encrypted, even if an unauthorized entity gains access to it, he will not be able to read it.

2. Secure shell (**SSH**) for remote login - we need to connect the RP remotely in order to implement and execute our code. SSH is a reliable and secure way for such action.

8.8 Risk Management

A potential risk is that the hardware components that we will choose won't work properly together or won't satisfy our needs for the project. We'll need to investigate them carefully and do a lot of research about the best components for our project.

The delivery time of the hardware component, due to the worldwide shipping issues and the lack of chips, there is an option that the component will arrive later than expected. We will need to do the ordering as quickly as possible or buy it in Israel(though prices will be much higher..).
The drone can crash while testing its functionality when it tries to land for reasons such as strong wind, human mistake, or software error. We'll do our best to test the drone carefully.
Assemble the drone (we will buy the parts separately - reduces costs) which is a knowledge we do not have. We will deal with this with the help of a drone expert, the hardware lab at Shenkar and guides from the Internet.
Another big risk is the voice communication between the LG and the drone. We will do research if it's possible and if it is, we will check whether we can buy the required equipment with our limited budget.

Software:

- Issues can occur due to a lack of API's documentation or incoherence. In this case, we'll try to have a reference from a working project on GitHub (or any other open source code platform) and figure out its use.
- A complex action to perform is when the LG touches the real-time display and the drone should move to that location (touch & go).

Interpersonal:

- Since all of us have jobs, planning and availability can be difficult to manage and overcome. We'll use a detailed schedule (using a dedicated management system such as Monday) in which we divide the work so that missions won't be missed.

8.9 Exceptions Management

During the development process we gave error handling and exception handling a lot of thinking, due to the fact that the system is supposed to be lifesaving and it works closely with people on the ground it is important to be as careful as possible.
If an error occurs in the system the default behavior of the drone is to stop and wait where it is.
We also added an 'Emergency' button that will cause the drone to immediately stop its engines and drop to the ground.
In the code every exception will lead to a log with tracing of the issue.

8.10 Versions Control

We used Git to track the history of DroneGuard's source code.
We used GitHub to host the code.
Each step of the development is separated into a branch, so that after testing and approval the changes are pushed to GitHub.
DroneGuard source code: https://github.com/idobetesh/DroneGuard

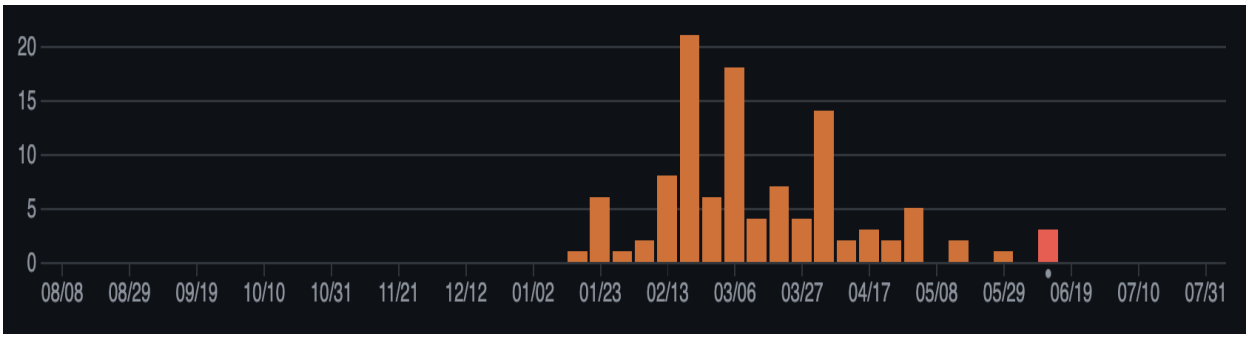

Figure 8.10: commit rate during project diagram.

8.11 Project Management

For Project management we used Trello in order to manage all of our tasks. We categorized the tasks into different categories in order to maintain our tasks in an organized way. For the tasks status we created our own customized statuses and kept updating the schedule.

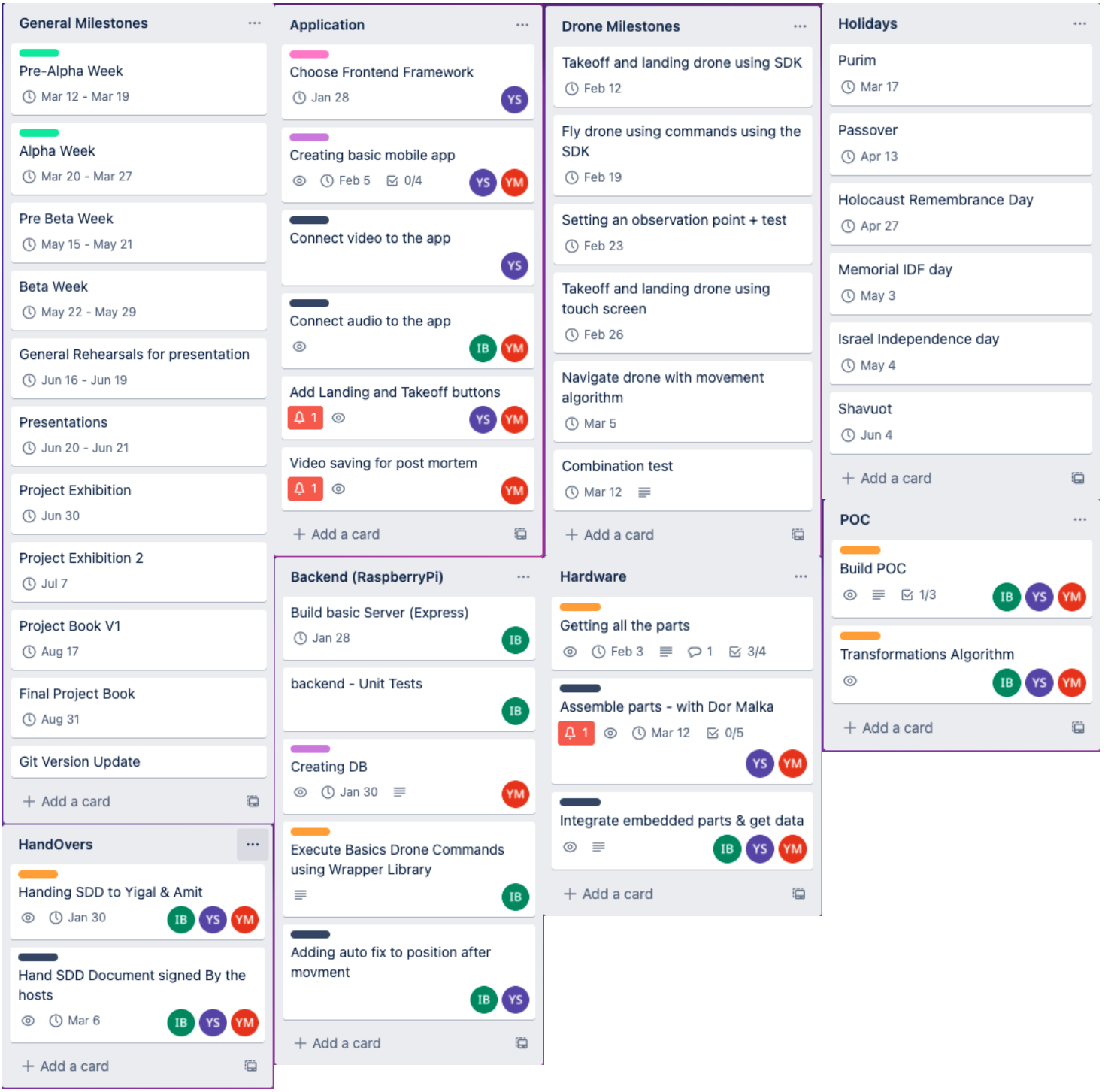

Figure 8.11: assignments during project .

8.12 DroneGuard - Source Code

DroneGuard App: https://github.com/idobetesh/DroneGuard/tree/master/droneguard-app
Drone Control Server: https://github.com/idobetesh/DroneGuard/tree/master/control-server
Debriefing System- Client:
https://github.com/idobetesh/DroneGuard/tree/master/debriefing-service/client
Debriefing System - Server:
https://github.com/idobetesh/DroneGuard/tree/master/debriefing-service/server
Debriefing System - Lambdas:
https://github.com/idobetesh/DroneGuard/tree/master/debriefing-service/lambdas
Hardware Configurations: https://github.com/idobetesh/DroneGuard/tree/master/hardware
DroneGuard Scripts: https://github.com/idobetesh/DroneGuard/tree/master/scripts

9 SYSTEM VALIDATION

- **Interfacing with the drone's flight controller and transmitting navigation commands**:
  - Sending navigation commands to the drone and checking the return value from the drone.
  - Checking the entire period in which we performed experiments and flights of the drone, including correcting and improving the navigation algorithm in order to reach the destination point with maximum speed and accuracy.
  - Checking that the commands were indeed carried out properly by the drone.
  - Flying the drone by basic predefined commands:

```
[
  { direction: 'right', distance: 235 },
  { direction: 'forward', distance: 123 },
  { direction: 'down', distance: 200 },
  { direction: 'up', distance: 200 }
]
```

Figure 9.1: pre-defined commands.

- **Algorithm Validation**:
  - By calculating conversions from the virtual -> real world and vice versa -
    - Conversion to real scale -
    - In calculating this algorithm, we use data from the camera on the drone (angle, camera resolution) and the height of the drone and calculate the scale perceived by the camera in meters.
    - Based on the camera resolution, the user interface will display a screen from which the navigation will take place, the screen will be on a scale of 1:4 for the camera resolution. Using the calculation from the previous section, a conversion can be made so that we get a scale of pixels per one meter.
    - Conversion from the virtual world to the real world -
      - By clicking on a point on the screen, a sampling of points (x,y) is performed which are sent to the algorithm.
      - The algorithm calculates the distance of the selected point from the center of the screen by using the conversion from the previous section and by using the Pythagorean theorem and finding the excess (the excess is the axis where the drone will fly and its calculation will give us the distance in meters that the drone must move).
      - Using the tangent we calculate the angle at which the drone must turn.

- **POC:**
  - In order to test the feasibility of the project, it was necessary to prove that the algorithm we built actually works and therefore a proof of feasibility was needed. In order to prove the feasibility, we used an image taken by a drone, with real data provided from that image (height, camera angle, resolution, coordinate of the place where the image was taken).
  - By clicking on a point in the image, we modeled coordinates (X, Y) and sent them to the algorithm that eventually outputs the coordinate of the clicked point. We searched for this coordinate on google maps and thus we could verify the accuracy of the algorithm.

- **Validation of System Safety**:
  - During the development of the system, we added validations to prevent human errors while using the drone. We performed this validation by limiting the drone in certain situations and by performing multiple flights while checking the edge cases.
  - We made the flights optimal and in sub-optimal conditions in order to see the behavior of the drone and give them an answer (such as an emergency button so that the drone does not cause damage).
  - We added restrictions during the tests in order to make sure that the use of the drone by an inexperienced person will not harm those around (such as maintaining a minimum flight height of the drone).

- **Users Management**:
  - On the main server of DroneGuard, the user details (rescuer/admin) are kept. There is a separation between the two so that in the debriefing system a LG is allowed to see the videos taken during his shift and is not allowed to delete them. On the other hand, an admin is exposed to all the videos and may delete them if necessary.

- **Camera Output Delay**:
  - In order for the drone's navigation to be done smoothly and effectively, the delay time of the camera output must be the lowest possible (closest to real time). At the beginning of working with the camera, the first delay we reached was 27 seconds, which is impossible to navigate this way. We invested a lot in finding better methods to reduce this time until we arrived at a delay that satisfies us and can be worked with.

10 SUMMARY, EVALUATION, CONCLUSIONS AND FUTURE WORK

10.1 Summary

DroneGuard is a system that is used as an aid to a rescuer at sea by using a drone that allows for a better ability to distinguish the bathers, quick identification of a bather who is in danger of drowning, the ability to communicate with the bathers and even release rescue means when necessary. The system was developed in order to expand the rescuer's toolbox and was designed so that the drone can be operated without experience and without prior knowledge of flying drones.

10.2 Future work

During this project, we developed reliable software that can be integrated with any programmable drone. As a POC, we used the smallest programmable drone on the market, which had issues with stability and flight time. In our vision, we would like to integrate our software with a serious drone and also attach some add-ons such as a life jacket and communication devices as well as continually improve the algorithm and add an AI that can track bathers in danger.

We believe this can really assist rescue forces and make a significant impact**.**

11 REFERENCES

[1] Shah, S. (January 18th, 2018). Lifeguard drone completes world-first ocean rescue. Retrieved from https://www.engadget.com/2018/01/18/little-ripper-lifeguard-drone-rescue

[2] Kelion, L. (November 13th, 2013). Iran develops a sea rescue drone prototype in Tehran. Retrieved from https://en.wikipedia.org/wiki/Pars_robot

[3] The National. (October 10th, 2018). The Flying Rescuer: meet Dubai's new life-saving drone. Retrieved from https://www.thenational.ae/uae/the-flying-rescuer-meet-dubai-s-new-life-saving-drone-1.779153

[4] Coxworth, B. (August 16th, 2018). “Lifeguard drone” rescues swimmers in Spain”. Retrieved from https://newatlas.com/drone-rescues-swimmer/55934/

[5] A simple navigation system for an UAV

https://www.researchgate.net/publication/254032588_A_simple_visual_navigation_system_f or_an_UAV

[6] Terrain aware image clipping for real-time aerial mapping https://www.researchgate.net/publication/327897436_TERRAIN_AWARE_IMAGE_CLIPPIN G_FOR_REAL-TIME_AERIAL_MAPPING

[7] Real-time and post-processed georeferencing for hyperspectral drone remote sensing https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2/789/2018/isprs- archives-XLII-2-789-2018.pdf

[8] Tudor, (January 09, 2021). Setting up your Drone for use with Map Pilot. Retrieved from https://support.dronesmadeeasy.com/hc/en-us/articles/206034873-Setting-up-your-Drone-for-use-with-Map-Pilot

[9] Parrot, How do I use the Touch & Fly function? Retrieved from https://www.parrot.com/us/support/anafi/how-do-i-use-the-touch-and-fly-function

[10] Integrated UAV-based real-time mapping https://www.mdpi.com/2220-9964/8/5/219/pdf

[11] Direct georeferencing with on board navigation components of light weight UAV platforms
https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B7/487/2012/isprsarchives-XXXIX-B7-487-2012.pdf

[12] Understanding maps of earth
http://www.nhcmtc.org/Extensions/Files/2013/1626/EarthKAM-GeoUnderstandingMaps.pdf