Efficient Drone Control in Emergency Situations

Filippa Strandlund

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2024

MASTER THESIS

<> remote.aero

Efficient Drone Control in Emergency Situations

Filippa Strandlund

LUND UNIVERSITY

Efficient Drone Control in Emergency Situations

Published by
Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design (MAMM01)
Supervisor: Kirsten Rassmus-Gröhn
Co-supervisor: Alexander Sandström and Fredrik Falkman at Remote Aero
Examiner: Joakim Eriksson

Abstract

This master's thesis has been conducted in collaboration with Remote Aero. The aim is to determine important design principles when developing an interface to be used for drone control in emergency situations. To determine the end user, an interview was conducted with the stakeholder. The end user was defined as an individual that works as a sea rescue volunteer at the Swedish Sea Rescue Society with experience of flying drones and has a drone authorization in category specific.

Moreover, data gatherings regarding control of drones, drone training, and sea rescues were conducted with potential end users. The results from the data gathering and interview with stakeholder was used to develop two personas in purpose of obtaining a deepened understanding of the target group. Thereafter, product requirements were determined which were prioritized using MoSCow analysis. Some of the essential needs was that the interface should assist the operator in conducting the flight safely and fast.

Furthermore, internal and external search was used to identify solutions to the problem statement. These solutions were visualized in different prototypes in Figma. Finally, a high-fidelity prototype was modeled which was used in usability testing with potential end users. This prototype and the results from the testing laid the foundation for the implementation of the final interface which were created with React, React Map GL, MapBox, GitHub, and Visual Studio Code.

The final product is an interface that can be connected with either a simulated or real drone. An additional usability testing was performed and in which the interface was connected to a simulated drone which the participants could control. In addition, the interface has been used in real flights to control the Swedish Sea Rescue Society's drones, however not under a real emergency situation.

Lastly, one can conclude that it is fundamental to develop products that match the user's mental model and contains error prevention. Moreover, it can be anti productively to include flexible processes since these instead can contribute in making the user slower. Finally, it can be efficient to limit information in an interface, both to its amount and the distribution within the interface.

Keywords: drone control, interaction design, interface, Figma, React, MapBox

Sammanfattning

Detta examensarbete har utförts i samarbete med Remote Aero. Målet var att fastställa viktiga designprinciper vid utformning av ett gränssnitt som ska användas för styrning av drönare i nödsituationer. För att fastställa slutanvändaren genomfördes en intervju med uppdragsgivaren. Slutanvändaren definierades till en individ som volontärarbetar som sjöräddare på Svenska Sjöräddningssällskapet med erfarenhet av att flyga drönare och har drönarkort i kategorin specifik.

Vidare utfördes datainsamling från potentiella slutanvändare gällande styrning av drönare, drönarutbildning och sjöräddning. Resultatet från denna datainsamling samt intervjun med uppdragsgivaren användes för att ta fram två personas i syfte att få en fördjupad förståelse av målgruppen. Därefter fastställdes produktkrav som prioriterades med hjälp av MoSCoW analys. Några av de viktigaste kraven var att gränssnittet skulle assistera operatören att genomföra flygningen säkert och snabbt.

Fortsättningsvis användes brainstorming och extern sökning för att identifiera lösningar till problemformuleringen. Dessa lösningar visualiserades i olika prototyper skapade i Figma. Slutligen modellerades en high-fidelity prototyp som användes i användbarhetstestning med potentiella slutanvändare. Denna prototyp och resultatet från testningen utgjorde grunden för implementationen av det slutgiltiga gränssnittet som skapades med hjälp av React, React Map GL, MapBox, GitHub och Visual Studio Code.

Slutgiltiga produkten är ett gränssnitt som kan kopplas till antingen en simulerad eller riktig drönare. En ytterligare användbarhetstestning utfördes och i denna var gränssnittet kopplat till en simulerad drönare som potentiella slutanvändare fick styra. Utöver detta har gränssnittet använts i flera flygningar för att styra Sjöräddningssällskapets drönare, dock inte under en riktig nödsituation.

Avslutningsvis gick det att konstatera att det är fundamentalt att utveckla produkter som motsvarar användarens mentala modell och som förebygger fel. Vidare kan det vara kontraproduktivt att införa flexibla processer eftersom dessa istället kan göra användaren långsammare. Slutligen är det effektivt att begränsa informationen både till innehåll men även distributionen i gränssnittet.

Nyckelord: drönarstyrning, interaktionsdesign, användargränssnitt, Figma, React, MapBox

Acknowledgments

I would like to thank my supervisor Kirsten Rassmus Gröhn, at LTH, for her constant commitment and contribution to this master's thesis. Your guidance and feedback, from start to end, have been crucial when conducting this degree project.

Moreover, I would like to thank Fredrik Falkman and Alexander Sandström, at Remote Aero, for giving me this opportunity. Thank you for your invaluable support and feedback. It has been a true pleasure working with you.

Lastly, I would like to thank everyone that participated in interviews, questionnaires, and usability testings. You have provided me with many important insights and helpful feedback.

I sincerely appreciate all of your time and dedication.

Thanks!

Stockholm, May 2024

Filippa Strandlund

Table of Contents

1. Introduction

In this section, background, purpose, problem description, research questions, definitions of concepts, and development tools of this degree project are presented.

1.1 Background

This degree project was conducted at Remote Aero AB, a company that develops drone control software. Remote Aero is one of the partners in the Swedish Sea Rescue Society's research project "Eyes on Scene 2 - Säkrare och effektivare Sjöräddning med drönar-stöd", with funding from Swedish Transport Administration. (Remote Aero, n.d). The purpose of the "Eyes on Scene 2"-project is to make sea rescues safer and more efficient with the support of drones (Sanchez-Heres et al., 2021).

Swedish Sea Rescue Society is a volunteer organization conducting sea rescue services. They participate in 90% of all maritime rescue efforts in Sweden. The organization has 2500 volunteer crew members and 74 rescue stations. Their rescue fleet consists of approximately 270 units that are specially adapted for any challenge that a sea rescue can entail (Swedish Sea Rescue Society, n.d).

A sea rescue case starts with an incoming alarm to a volunteer on-call via an emergency phone (S.Johansson, personal communication, January 28). According to Sanchez-Heres et al. (2021), the information about the alarm is usually insufficient. Occasionally, the information is even inaccurate or contradictory. In order for the rescue crew to perform suitable preparations and provide relevant help, it is essential to understand the situation properly.

The "Eyes on Scene 2"-project suggests involving drones with cameras in sea rescues. Normally a crew leaves the shore within 12 minutes after an incoming alarm. By involving drones, it is possible to have "eyes on scene" in purpose of gathering information about a sea rescue case before the rescue boat has left the shore. This in turn will make sea rescues safer and more efficient (Sanchez-Heres et al., 2021).

Drones will be operated from a web based interface by one of the Swedish Sea Rescue Society's on-call volunteers. Moreover, the operations will be performed beyond visual line of sight which implies that the person operating the drone does not consistently have visual contact with the drone (Sanchez-Heres et al., 2021; SOARIZON by Thales, 2020). The drones will be fixed-wing drones (Sanchez-Heres et al., 2021).

Remote Aero's role in the "Eyes on Scene 2"-project is to develop the software of the drone system (A. Sandström, personal communication, December 12). This degree project is focused on developing the web based interface from which the drones will be controlled from. Primary focus is on deciding where the drone should go and how to point the drone's camera in a safe and fast way.

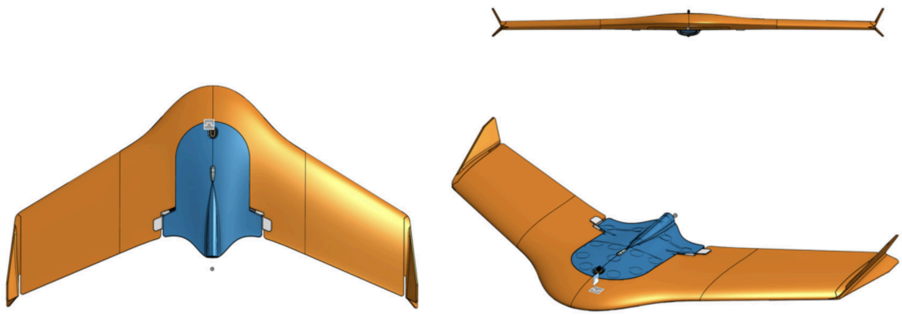

Figure 1. 3D-model of fixed-wing drone "EOS-vinge" from Eyes on Scene

From *Eyes on Scene - EOS.* [Image], by Sanchez-Heres, L., Falkman, F., Forsman, F., Hagner, O., Nilsson, V., Lundqvist, R., Bergman, K., Bjurling, O., Granlund, R., Hjalmarsson, M., & Brunnström, M., 2021. RISE Research Institutes of Sweden.

Last accessed 2024-04-11 from https://fudinfo.trafikverket.se/fudinfoexternwebb/Publikationer/Publikationer_007401_007500/Publikation_007477/eos-slut-rapport.pdf

1.2 Purpose

In order for the drone to arrive at the location before the rescue crew has left the shore, the interface from which the drone control is performed needs to allow the operator to operate fast. Existing drone control interfaces are adapted to non-dynamic use cases like scanning and inspection (A. Sandström, personal communication, January 11, 2024). Thereby, an interface adapted for this specific use case will expectedly allow faster and more efficient control of the drone.

A drone that is controlled out of sight of the operator poses a security risk to external actors. (The Swedish Transport Agency, 2023). Thus, a significant part of the design is to ensure that the interface prevents the operator from accidentally giving unplanned commands to the drone. The objective of this degree project is to research how the interface can be developed to achieve the stated goals from an interaction design perspective.

This degree project aims to contribute to the knowledge of significant interaction design principles when designing efficient control software, specifically for drones, to be used in time sensitive or pressured situations, like an emergency situation.

1.3 Problem Description

A drone control interface adapted to be used in emergency situations is to be developed. The interface should allow the operator, one of the Swedish Sea Rescue Society's on-call volunteers, to control a drone fast and safely (Sanchez-Heres et al., 2021). The drone control interface will be integrated in the following workflow.

1. Something has occurred at sea
2. Incoming alarm
3. Alarm is received by on-call volunteer at the Swedish Sea Rescue Society
4. Volunteer uses the interface to "Take control" of a drone, i.e chooses which drone to use
5. Volunteer uses the interface to launch drone from launch box
6. Volunteer uses the interface to create drone's route
7. Volunteer uses the interface to edit drone's route
8. Volunteer uses the interface to control the drone's camera
9. Drone arrives at the location of the alarm

Focus of this degree project will mainly be step 4, 6, 7 and to some degree step 5 and 8.

According to the stakeholder, previous work has been conducted with the aim of developing an automated drone control interface. However, the work with the automation proved to be immensely time-consuming. Thus, in order to as simply as possible be able to control the drone, this degree project is focused on manual control of the drone (F. Falkman, personal communication, April 8, 2024). This includes setting and editing routes by creating GoTo-points and choosing and editing focus of the drone's camera, i.e the Point of Interest

Finally, it is intended for the future that one single operator should be able to choose among several drones which drone to control and potentially control several drones at the same time (Sanchez-Heres et al., 2021). Thus, it is preferable that the interface to be developed should contain a solution where these future use cases are taken into consideration.

1.4 Research Questions

The following research questions will be examined.

1. How is a GoTo-point set?
2. How is a GoTo-point changed?
3. How is a Point of Interest set?
4. How is a Point of Interest changed?
5. How can the system minimize undesired commands?
6. How can the system allow the operator to operate fast?
7. How can the system allow the operator to operate safely?

1.5 Definitions of Concepts

In this division, concepts that have been included in this report are explained in greater detail. These concepts are related to the drone's camera, location of the drone, and general terms related to drones.

1.5.1 GoTo-point and Point of Interest

In this degree project, GoTo-point refers to a location which the drone will or is currently heading to. Point of Interest refers to a point which the drone's camera is directed towards.

1.5.2 VLOS, EVLOS, and BVLOS

SOARIZON by Thales (2020) explains that there are different terms that describe unmanned aircraft pilot's of line of sight, VLOS, EVLOS, and BVLOS.

- **VLOS** "Visual Line of Sight" implies that the pilot consistently has visual contact, solely using natural eye sight, with aircraft.
- **EVLOS** "Extended Visual of Sight" means that one or more observers, besides the pilot, are involved to have visual contact with aircraft.
- **BVLOS** "Beyond Visual Line of Sight" entails that the pilot does not consistently have "visual line of sight on the aircraft".

An illustration of the definitions can be seen in Figure 2.

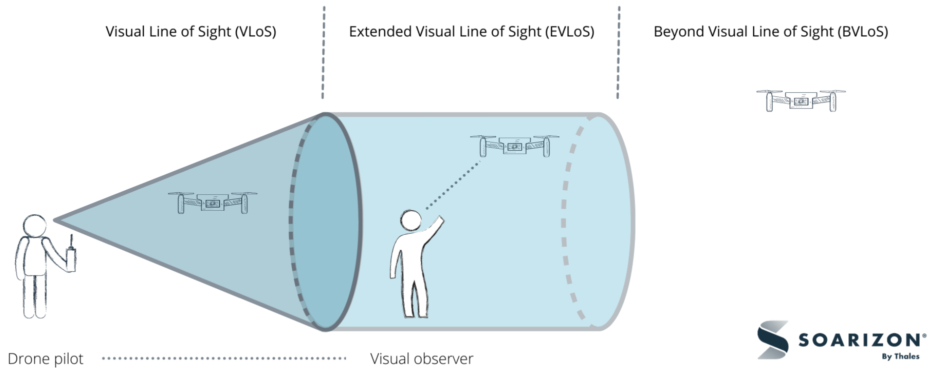

Figure 2. VLOS, EVLOS, and BVLOS illustrated

From *What are VLOS, EVLOS and BVLOS? Why do they affect drone operations?* [Image], by SOARIZON By Thales. Last accessed 2024-04-09 from https://www.scaleflyt.com/news/what-are-vlos-evlos-and-bvlos-why-do-they-affect-drone-operations

1.5.3 Drone Authorizations

There are three different types of drone authorizations, open, specific and certified. These are categorized according to the risk level associated with the flight (The Swedish Transport Agency 2023).

An open drone authorization allows the operator to fly a drone within sight and not higher than 120 meters. The open category includes three different categories, A1, A2 and A3, where each category implies how close to people and buildings the operators are allowed to fly, and with which drone. A1 entails that the operator has permission, to a limited extent, to fly certain drones near people. An A2 authorization implies that the operator has to keep a distance of at least 30 meters to people. If the flight is to be conducted at a lower speed, then it may be allowed to be performed as close as 5 meters. Lastly, A3 means that the operator only has permission to fly drones on a sideways distance of 150 meters to recreation, industry, residential or business areas (The Swedish Transport Agency 2023).

If a flight is to be conducted outside of the operator's sight or higher than 120 meters, the operator needs to have an authorization in the specific or certified category, primarily the specific category. However, if the risk level associated with the flight is high, the operator needs an authorization in the certified category (The Swedish Transport Agency, 2023; The Swedish Transport Agency, 2024).

1.5.4 Types of Drones

This chapter explains the two types of drones mentioned in this report, the fixed-wing and multi-rotor drone.

1.5.4.1 Fixed-Wing Drone

A fixed-wing drone has a single wing and is constructed to both look and function as a traditional airplane. Lift is generated by the wing which makes the drone energy efficient since it does not require any energy to maintain itself in the air. The fixed-wing drone has long endurance and fast flight speed. Nevertheless, it is relatively difficult to fly and additional training is required (Rennie, 2016).

Figure 3. Fixed-wing drone

From *Benefits of Fixed Wing Drones over Multirotors* [Photography], by Liska, D. Women And Drones. Last accessed 2024-04-05 from https://womenanddrones.com/benefits-of-fixed-wing-drones-over-multirotors/

1.5.4.2 Multi-Rotor Drone

The multi-rotor drone consists of multiple motors, hence the name. It is simple to use, provides substantial control, and has great maneuverability. Nonetheless, the multi-rotor drone requires a lot of energy to maintain itself in the air. In addition, it has limited speed and endurance (Rennie, 2016).

Figure 4. Multi-rotor drone

From *Mavic 2 Enterprise Advanced* [Photography], by Terrestrial Imaging. Last accessed 2024-04-05 from https://www.terrestrialimaging.com/drone-equipment/multi-rotor-drones

1.6 Development Tools

This chapter presents the development tools that have been used throughout this degree project.

1.6.1 Figma

Figma is a collaborative design platform that supports developing designs for digital products and visualizing ideas. By providing tools for creating, sharing and testing designs, Figma aims to facilitate decision-making, contribution, and feedback for any person that is part of a design activity. Figma provides the possibility of creating multiple pages design files where their users can insert layers, texts, images, shapes and use various tools. (Figma, n.d).

1.6.2 Visual Studio Code

Visual Studio Code is a code editor that provides tools for development activities such as coding, building, and debugging (Visual Studio Code, 2024). The code editor includes internal support for TypeScript, JavaScript, and Node.js.

Additionally, it compromises add-ons for a range of various languages such as Python, Java, C++, and C# (Microsoft, n.d).

1.6.3 GitHub

Github is a platform that supports developers in the following activities (Cliford Opoku, 2024).

- Share and host code
- Collaboration on projects
- Version control
- Creating websites
- Assign and track issues
- Controll reviews of pull requests and incorporation of modifications

1.6.4 Mapbox

Mabox provides location and map services for a range of software applications. This involves providing APIs for navigation, maps, search, accounts and vision services. In addition, Mapbox offers their users to personalize maps by giving them the option to change various attributes (Mapbox, n.d).

1.6.5 React

React is a library that can be used to build native and web interfaces. The library allows developers to combine single elements, or components as they are called in React, into complete user interfaces. Developers can write their own components or use components from other sources such as independent individuals or organizations (Meta Open Source, 2024).

1.6.6 React Map GL

React Map GL is a react wrapper for Mapbox GL JS. The wrapper contains a set of React components that can be used in conjunction with libraries that involve Mapbox GL JS (Vis.gl, n.d).

1.6.7 Material UI

Material UI is a React component library that implements Google’s Material Design. Components can be adapted with states, colors, icons, variations, and typography. (Material UI, n.d)

2. Theory

Theories regarding usability, guidelines for design to be used in emergency situations, color choices, color contrast, buttons, and menus are discussed.

2.1 Usability

According to Nielsen (2012), usability measures how simple and pleasant a user interface is to use. Preece et. al (2011) breaks down usability into the following objectives.

- "Effective to use (effectiveness)"
- "Efficient to use (efficiency)"
- "Safe to use (safety)"
- "Having good utility (utility)"
- "Easy to learn (learnability)"
- "Easy to remember how to use (memorability)"

In addition, usability is essential for the user's productivity and has an impact on how many tasks a user is able to achieve in a specified amount of time (Nielsen, 2012).

2.1.1 Usability Heuristics

Nielsen (2024) describes ten usability heuristics for interaction design.

1. **"Visibility of System Status".** The system shall constantly keep the user informed about the system's state. This is done by providing feedback to the user, preferably immediately. Nonetheless, within an appropriate time period is also acceptable. When the system status is visible to the user, they can more easily predict outcomes of their actions which in turn builds trust for the application.

2. **"Match between System and the Real World".** Placing content in a logical order, using real-world standards and user terminology, makes it more convenient for the user to learn how to use the interface. To achieve

this, user research can be performed to determine the intended end user's vocabulary and mental model.

3. **"User Control and Freedom".** Occasionally the user will conduct unintended operations within an interface. In order to put the user in control and provide user freedom, it is essential to support the user to leave these undesired actions in a convenient way. This can be done by including for example cancel, undo, and redo buttons.

4. **"Consistency and Standards".** Through maintaining external and internal consistency within the interface, the user's learnability can be improved. External consistency is achieved by implementing existing industry standards. Internal consistency is accomplished by keeping the system or product family coherent and logical throughout.

5. **"Error Prevention"**. The system shall, besides including supportive error messages, essentially prevent errors from occuring. This can be done by making the user confirm actions and reduce the number of situations where errors usually occur.

6. **"Recognition Rather Than Recall".** Reduce burden on the user's memory by making options, actions, and elements easy retrievable. Information that is needed to use the system, such as menu elements or field labels, should be obvious or effortlessly accessible. Additionally, display contextual guidance rather than a lengthy tutorial.

7. **"Flexibility and Efficiency of Use".** Allow the users to navigate the system differently depending on their expertise levels. Include a possibility to personalize functionality and content, make customized settings, and use accelerators such as keyboard shortcuts.

8. **"Aesthetic and Minimalist Design".** All information included in the interface should be carefully selected since each element reduces its proportionate clarity. Additionally, the content should be prioritized to facilitate the user's goals.

9. **“Recognize, Diagnose, and Recover from Errors”.** Error messages within the interface should be designed in a standard way, for example in red, bold text, and written with terminology that the user understands. The error message should also include a solution that could fix the error instantly.

10. **“Help and Documentation”.** The interface should strive to display documentation contextually when users need help to complete a certain operation. The documentation and help information should be to the point and include concrete actions that are required to achieve a specific task. Furthermore, it should target the user’s action and be convenient to search.

2.1.2 Error Prevention

Nielsen (2024) explains that there are two types of errors that a user can make, slips and mistakes. According to Laubheimer (2015), slips are those types of errors when the user aims to achieve a specific task, nonetheless unintentionally performs another action. One example of a slip could be when a user that intends to write an “o”, however accidentally writes an “i”. In order to prevent slips from occurring, the system shall implement the following principles.

- Supportive constraints
- Contextual suggestions
- Reasonable defaults
- Lenient formatting

Laubheimer (2015) states that mistakes are those types of errors that occur when the user interprets a situation incorrectly and tries to achieve a goal which does not correspond to the actual situation. To prevent mistakes, the system has to match the user’s mental model. This can be done by conducting the following activities.

- Gather user data to understand users’ expectations and why they conduct certain mistakes
- Implement standard design conventions
- Make the system communicate how it is supposed to be used. This may involve the following design principles.
    - Convey that a button is clickable by making it stand out from its surrounding through including a shadow
    - Communicate that a field can be filled by including a shadow on the inside of the field
- Include a preview of the outcomes of the user’s action before it is confirmed

Furthermore, Laubheimer (2015) suggests that the following principles help prevent all types of errors, including slips and mistakes.

- Minimize users' memory load by removing situations where the user has to remember information in several steps. This can be achieved by including contextual information when needed to perform a certain action.
- Make the user confirm destructive actions before they occur. However, carefully decide when to display a confirmation dialog since having too many can cause the user to stop reading them and then in turn increase the number of errors.
- Display contextual error warnings before an action is conducted

2.2 Display Design Emergency Situations

Hancock and Szalma (2003), discusses seven guidelines of display design for emergency situations. These guidelines are established on insights how operators function under task and time burden and can improve decision making in emergency situations.

- Reduce distribution of information over various sources
- New information shall be associated to the data that is presently being handled
- First hand, show an overview of information on a single display by using integrated information formats. When relevant, present changes over time of visual elements.
- Use detachable configural windows when integrated windows are not pragmatic
- Employ logical semantic mapping when using configural displays
- Apply principles of the ecological interface design to simplify fulfillment of the above guidelines
- Use simple graphics to deliver direct instructions on a low-level when the user is under most stress
- Avoid "direct structural interference" when the user is experiencing a lot of stress

2.3 Design Theory

The following section contains theory regarding high contrast colors, buttons, and menus.

2.3.1 High Contrast Colors

W3C Web Accessibility Initiative (2023) suggests that to ensure individuals with fairly low vision are able to read all text, the contrast ratio between background and text color should be at least 4.5:1. Sherwin (2015) explains that use of low contrast colors makes the readability of text decrease, makes it difficult for users to find relevant functions when scanning a page, reduces accessibility, lowers user's ability to keep focus, and extends the amount of time it takes for users to establish a correct interpretation.

2.3.2 Buttons

Suwala (2019) expresses that elements that are interactive, i.e react when being hovered over or clicked on, should be distinctly outstanding from the rest of the contents. A button's location on the screen, size, shape, and color, plays a significant role in a user's experience of an interface.

Many users' have from utilizing applications over several years, built a general understanding of the outcome when hovering or clicking on a button. Hence, buttons should be designed with a shape that users' already have a relation to. The following list contains four typical kinds of buttons that are illustrated in Figure 5 (Suwala, 2019).

- Filled with squared corners
- Filled with rounded corners
- Filled with shadow
- No fill and thin border (ghost button)

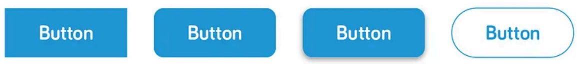

Figure 5. Illustrations of frequently used button styles

From *Build trust with buttons. 7 rules about designing buttons that will help you build satisfying User Experiences (UX) and User Interfaces (UI).* [Image], by Katarzyna Suwala. Last accessed 2024-04-08 from https://medium.com/itmagination/build-trust-with-buttons-629657f966ee

According to Suwala (2019), evidence proposes that rounded borders can simplify users' processing of information and direct them to the center of an element. In addition, a shadow functions as an optic symbol that the component stands out from the rest of the content and is interactive.

Furthermore, colors can be used to indicate whether a button contains information or an action. If a button prompts the user for an action, for example "Register" or "Buy now", it should be modeled as a Call-to-Action button. This can be done by modeling it in a highly contrasting color compared to the surrounding environment. It is ideal to choose the same color for all Call-to-Action buttons in a system. Designing Call-to-Action buttons in this way will facilitate users' identification of key functions in an interface (Suwala, 2019).

According to Suwala (2019), it is good practice to model buttons that are of "secondary importance" in less contrasting colors compared to Call-to-Action buttons. Actions that have lower significance should be modeled as ghost buttons. Usually, gray is exclusively used for elements that contextually are to a smaller extent desirable or inactive. Additionally, to ensure that a button is clearly visible, the amount of space surrounding it is equally as important.

Moreover, Suwala (2019) states that it is profitable to convey a button's state with color. For instance, to indicate that a button is getting interacted with, it can change its color from a darker or lighter shade when a user hovers over it.

2.3.3 Menus

Nielsen (2008) suggests applying the following design guidelines to allow users to rapidly scan vertical menus.

- Start list items with different words
- Label each menu element so that it starts with the "one or two most information-carrying words"
- Justify the menu in the same direction as the language is meant to be read. Left alignment if language is read from left-to-right and right alignment if language is read from right-to-left.

In addition, menus, regardless if vertical or horizontal, should avoid text in all caps since it decreases legibility of 10%.

3. Methodology

According to Friis Dam (2024), design thinking is an iterative, non-linear process that "provides a solution-based approach to solving problems". Design thinking could be broken down into the five following phases.

- *"**Empathize**: research your users' needs"*
- *"**Define**: state your users' needs and problems"*
- *"**Ideate**: challenge assumptions and create ideas"*
- *"**Prototype**: start to create solutions"*
- ***"Test**: try your solutions out"*

3.1 Empathize

The empathize stage concentrates on user analysis. Primary objective of this phase is to understand the users, their requirements, and the need for the product to be developed. In order to do that, one has to gather information about the target group and area in concern (Friis Dam, 2024).

Preece et. al (2011) states that there are three general methods to collect qualitative or quantitative data, interviews, questionnaires and observation. Interviews include an interviewer querying an interviewee a number of questions, generally eye-to-eye and synchronously. The interviews and set of questions can be highly structured or unstructured. Questionnaires are a set of questions intended to be answered asynchronously and can be conducted online or on paper. Observations is about observing users' activity and can be performed direct or indirect, depending on when the users' activity is observed.

According to Preece et. al (2011), it is fundamental to gather applicable and satisfactory data in order to establish stable requirements. Additionally, data gathering is essential in evaluation purposes when recording users performance and reception of an application. Finally, it is vital to combine and use the data gathering techniques flexibly to avoid biases.

3.1.1 Interviews

Preece et Al. (2011) discusses four primary kinds of interviews, structured, unstructured, semi-structured, and group interviews. The first three interview types vary from one another depending on how much the interviewer obtrudes the conversation by sticking to the prearranged questions. Group interviews, includes a limited number of individuals guided by a facilitator. The most suitable interview style depends on which phase the project is in, purpose of the interview, and questions included.

3.1.1.2 Unstructured Interviews

Unstructured interviews can be described as informative and more like conversations about a preplanned subject. The questions are open, which means there is no specified format or expectations of any specific content from the interviewee's answers. Each interview will have its own structure and a great amount of unstructured, nonetheless rich, data will be generated. A prominent advantage of unstructured interviews is that they usually provide substantial depth about the topic. However, the data can be tedious to analyze. This aspect needs to be taken into consideration when the interviewer chooses what interview technique to use (Preece et Al., 2011).

3.1.2 Questionnaires

Preece et Al. (2011) explains that questionnaires involve a number of predetermined questions that are to be answered asynchronously. Questions should be clearly worded, specific, and can be either open ended or closed. Questionnaires could be used independently or to confirm conclusions made from other data gathering activities. It is suitable to use this technique if the respondent is motivated to respond, otherwise an interview format is more appropriate.

3.1.3 Observation

Observation entails observing users' activity of a system, direct or indirect. The distinction between direct observation and indirect observation is when the users' activity is studied. Direct observation implicates observing the user in real time and indirect observation includes recording the activity to be studied later. The method can be used during any product development phase. Observation in an early development stage can allow insights regarding users' goals, tasks, and backgrounds. Later on in a design development process, for example during evaluation, observation can examine how well the system assists the user in the specified goals and tasks (Preece et Al., 2011).

3.2 Define

Friis Dam (2024) explains that within the define phase, the information collected in the empathize phase are analyzed and organized to define a user centric problem statement. According to Preece et Al. (2011), there are three elementary types of simple qualitative analysis, thematic analysis, categorizing data, and critical incident analysis. All of these methods could be used independently or in conjunction with one another.

Gibbons (2021) presents MoSCOw analysis and explains how the method could be used to prioritize the identified functions and features of the product to be developed. Moreover, Pressman and Maxim (2020) discusses how personas can be defined with the aim of obtaining a broader picture of the target group.

3.2.1 Thematic Analysis

When initially studying the results from a data gathering activity, potential patterns or themes can arise. These observations have to be validated with more in depth analysis, where both verifying and contradicting information from data are taken into consideration. The purpose of the data gathering presents a focal point for identifying patterns and themes. If a validated theme is highly relevant for the study goal it is identified as a main theme. If a theme is less relevant, it is identified as a minor theme. The thematic analysis can also be used on quantitative data (Preece et Al., 2011).

3.2.2 Categorizing Data

Preece et Al. (2011) explains how data can be analyzed through categorizing it according to a categorization scheme. The scheme can emerge from the data itself, be based on another well renowned organization scheme or it can be created as a conjunction of those two. The scheme should consist of relevant mutually excluded categories. The categorizing data method can be used to identify behavioral patterns and to quantify how many users that experience specified problems within a system.

3.2.3 Critical Incident Analysis

According to Preece et Al. (2011), the prominent approach within critical incident analysis is to diagnose important situations and then concentrate the analysis on these. Remaining data should be used for context when examining. The critical incident technique is notably useful when great amounts of data has been gathered.

3.2.4 MoSCow Analysis

MoSCow analysis is a simple prioritization method that could be used both on a project increment or complete project. All identified product requirements are assigned one of the four following priorities.

- **M - Must have.** Needs that are fundamental. The product will not exist if these are not delivered.
- **S - Should have.** Needs that support core functionality, however, the product will still function without these.
- **C - Could have.** Needs that are desirable and pleasant to have, nonetheless, it does not affect the product much if they are excluded.
- **W - Will not have.** Needs that are not currently required since they do not bring enough value to the product.

"Must have" is the highest prioritization and "Will not have" is the lowest. The outcome from using MoSCow analysis is four distinct groups of product requirements which will guide the designer team in focusing on the essential features of the product. There is not a specified number of how many needs there can be in one group. However, if there is an indefinite amount of time there is a risk that all product requirements will be identified as must haves. Thus, the MoSCow analysis is most suitable when developing a product within a specified time period (Gibbons, 2021). Lastly, assigning each feature an unique identifier facilitates referencing and keeping track of features (Bittner & Spence, 2002).

3.2.3 Personas

Preece at Al. (2011) describes user characteristics as the prominent attributes of the target group. The users' competencies and skills are both notable user characteristics. However, the user characteristics can also involve users' educational backgrounds, preferences, knowledge of the system, and physical or mental disabilities. This set of characteristics is called a user profile and a persona is formed when one is creating a fictional person based on the user profiles of a product. A persona is normally a representation of several users who have contributed to the data gathering. Pressman and Maxim (2020) explains that personas can be developed to obtain a deeper understanding of the intended end users, such as their nature and goals.

3.3 Ideate

In the ideate phase, product requirements shall be considered from various viewpoints. Solutions should be ideated based on the deep understanding of the user obtained from the empathize and define stages (Friis Dam, 2024). Ulrich and Eppinger (2015) presents two ideation methods, Internal Search and External Search.

3.3.1 Internal Search

According to Ulrich and Eppinger (2015), internal search or brainstorming implies to use creativity and knowledge within the individual or the team to generate concepts. To achieve a successful internal search, it is important to develop many ideas, avoid judgment of concepts, accept concepts that potentially seem unattainable, generate a good deal of sketches, and make sketch models.

3.3.2 External Search

Ulrich and Eppinger (2015) explains that the objective of external search is to investigate existing solutions to the problem statement. The external search methodology can be used to identify solutions that can solve complete subproblems or be combined with solutions from internal search.

3.4 Prototype

According to Friis Dam (2024), the prototype phase is where designers develop a set of scaled down variants of the product or selected features of it. The aim is to examine the best possible solutions from the ideate stage. Prototypes can be evaluated within the designer team itself or with external actors such as a limited number of individuals excluded from the design team.

3.5 Test

The final chapter of a design process is the test phase. In this stage, the prototypes that have been developed are being tested by designers or testers (Friis Dam, 2024).

3.5.1 Usability Testing

Moran (2019), describes how usability testing can be used to get a better understanding of intended users' preferences and nature, discover features to enhance, and diagnose issues with the system. There are three fundamentals in usability testing, facilitator, tasks and participants.

The facilitator determines a set of realistic tasks that the user should perform and then guides the participant through these. Tasks activities can be either open ended or specific. Task instructions should be delivered to the participant by the facilitator either written or read aloud. Moreover, the facilitator role includes observing the participant while performing the tasks, responding to users' queries, and asking follow-up queries. The participant should be a realistic user of the system (Moran, 2019).

In general, the think aloud protocol is used in usability testing (Moran, 2019). This implies that the user is asked to "verbalize their thoughts" when performing tasks in the system. The think aloud technique can help identify and give insights why participants do certain misconceptions (Nielsen, 2012).

Pernice (2014) specifies guidelines regarding talking to participants during usability testing. When a participant asks a question, the boomerang technique can be used. Instead of answering test participants' questions, a "generic non threatening question" can be formulated by the facilitator. This will bounce back the question to the test participant.

Lastly, the echo technique can be used to get the user to elaborate on certain topics. Repeating a phrase or part of a phrase stated by the participant and verbalizing it as a question will make the user respond by explaining what they meant with that phrase.

3.5.2 Task Completion Rates

Budiu & Nielsen (2021) states that users' success in completing tasks are fundamental in usability. One way to measure this is with task completion rates, i.e how many users that accomplish a task. To measure participants' full, partial, and non success of a task, the following success levels can be used.

- "Complete success"
- "Success with a minor issue"

- “Success with a major issue”
- “Failure”

Furthermore, Sauro (2011) defines acceptable completion rates for different kinds of tasks. Tasks that have a high failure cost, for example loss of money or life, should have a success rate of 100%. However, a completion rate around 70% is accepted for tasks that have a lower failure cost.

4. Empathize and Define

In this section, methodology described in 3.1 Empathize and 3.2 Define has been used with the aim of obtaining a deeper understanding of the intended user and defining a user centric problem statement.

4.1 User Research

With the purpose of obtaining a clear picture of the target group, data have been gathered from both stakeholders and potential end users. Methods such as interviews and questionnaires for collecting data described in 3.1.1 Interviews and 3.1.2 Questionnaires have been used. Furthermore, when applicable the data have been analyzed according to the categorizing data method described in 3.2.2 Categorizing Data. Specified drone types and authorizations are explained in 1.4.4 Types of Drones and 1.4.3 Drone Authorizations respectively. The line of sights mentioned are described in 1.4.2 VLOS, EVLOS, and BVLOS.

4.1.1 Data Gathering Stakeholder

Data was gathered from one of the stakeholders and then analyzed. The result was a list of characteristics that the user will have and prior relevant experience they most probably will have. Additionally, professional backgrounds that would be ideal, however not required, is stated.

4.1.1.1 Method

Initially, there was an unstructured interview with the stakeholder with the purpose of identifying the target group. The interview was performed online, synchronously, and the predetermined topic was the intended end user. Questions were open ended and formulated upon the interviewee's responses. Data gathered from the interview were then later categorized according to the categorizing data analysis method. The categorization scheme emerged from the data itself and had the following categories, characteristics that the intended end user will have, probably have, and ideally would have.

4.1.1.2 Results

The following results are obtained from personal communication with Fredrik Falkman (2023, January 18).

The intended end user will have the following characteristics.

- Experience of flying drones
- A drone authorization in category specific
- Volunteer as a sea rescuer at the Swedish Sea Rescue Society

The intended end user will most likely have the following experience.

- First-hand on-site
- Flying one drone at a time
- Flying drones within sight
- Controlling drones from a mobile or bigger screen

The intended end user would ideally, nonetheless not necessary, have experience in any of the following professions.

- Air traffic controller
- Pilot

4.1.1.3 Discussion

From the interview results, one can make the conclusion that the intended end user will be a person that has experience of flying drones, a drone authorization in category specific, and works as a sea rescue volunteer at the Swedish Sea Rescue Society. The users will have varying experience of flying drones and air traffic. Thus, the control interface has to be designed in a way that ensures that users with less air traffic or drone experience can make informative decisions within the interface. The user's prior drone experience is mainly from flying drones within sight and controlled from a mobile or bigger screen. In this application, the operator is required to fly the drone out of sight and will most likely control the drone from a computer screen with a mouse.

4.1.2 Data Gathering Potential End User

Data gathering has been conducted through interviews and questionnaires with individuals from the intended target group. The data gathering activities were either focused on drones, including topics as drone training and safe control of drones, or sea rescues, including questions regarding necessary or desirable information during an emergency call.

4.1.2.1 Method

Based on the results of data gathering from the stakeholder, the intended end user was identified as a person that had experience of flying drones, has a drone authorization in category specific, and works as a sea rescue volunteer at the Swedish Sea Rescue Society. Thus, persons that had any of these characteristics were identified and contacted via email.

Individuals with experience of flying drones and having a drone authorization were queried via email if they would like to answer questions regarding their drone experience via email or a conference call. If the participant chose to answer questions via email, data gathering was performed as a questionnaire with open ended questions. If the participant chose to answer via conference call, data gathering was conducted as an unstructured interview with open ended questions. The questions were divided into three categories, experience of drones, drone training, and control of drones, and were as follows.

- Experience of drones
  - What experience do you have of drones?
  - How long have you been flying drones?
  - How often do you fly drones?
- Drone training
  - Have you completed a drone training course?
  - What drone training did you complete?
  - How was the drone training conducted?
  - Was safety around the use of drones included in the training?
- Control of drones
  - What kind of information is absolutely necessary when controlling drones?
  - What kind of information is desirable when controlling drones?
  - What kind of information is absolutely necessary to conduct a safe drone flight?
  - What kind of information is desirable to conduct a safe drone flight?

Furthermore, a questionnaire with open ended questions was conducted with an individual that has a background as an air traffic controller, rescue leader, and sea rescuer. The questions were focused on sea rescues and as follows.

- How does a sea rescue work? Please explain step by step if possible.
- How long does it take from the alarm to emergency call out?
- How long does it take from the alarm until the maritime rescue is on site?
- How is the situation experienced during an emergency call? What feelings arise?
- What opportunities are there to create an overview of the situation before the maritime rescue is at site?
- What kind of information is necessary to know in the event of an emergency call?
- What kind of information is desirable to know in the event of an emergency call?
- Is there any other information I should know about an emergency call?

The results from the questionnaires and interview were then categorized according to a categorization scheme that emerged from the data itself.

4.1.2.2 Results

The results from data gathering with potential end users consist of information regarding drone training, safe control of drones, and sea rescues.

4.1.2.2.1 Drone Training

Both of the participants had conducted The Swedish Transport Agency's A1/A3 open category drone training. One of the participants had additionally conducted the A2 category and a commercial drone operator training. The commercial drone operator training was within the open category, nonetheless involved some parts relevant for the specific category. Answers from the participants demonstrated that their drone training within the open category contained the following information.

- Drone flight safety
- Responsibilities of a operator before, under, and after a flight
- Allowed heights to fly drone within
- Areas that one are allowed or not allowed to fly drone within
- Weather conditions

4.1.2.2.2 Safe Control of Drones

Verbalizing from participants shows that the following information is absolutely necessary to safely control a fixed-wing drone BVLOS.

- Speed
- Air speed
- Height
- GPS position
- Heading
- Time remaining of battery
- Live video feed

Additional information that is desirable to know, nonetheless not absolutely necessary to safely control a fixed-wing drone BVLOS, is as follows.

- Distance from operator
- Horizon gyro
- Rate of descent
- Rate of climb

4.1.2.2.3 Sea Rescues

Answers from the respondent demonstrates that the following information is necessary to know before an emergency call out.

- Type of event
- Type of boat
- If it leaks water into the boat
- Number of people that needs help
- Number of people in the water
- Medical overview
- Communication plan

The respondent verbalizes that usually further information is received during an emergency call out. In addition, the respondent expresses that occasionally the initial location of the alarm is incorrect and that the situation may change quickly.

4.1.3 Personas

In this chapter, methodology regarding personas presented in 3.2.3 Personas are used to develop two personas. The personas are based on data gathered in different data gathering sessions during the project.

4.1.3.1 Method

With the purpose of acquiring an improved understanding of the intended end user, two personas were created, Based on the responses from the interviews and questionnaires with stakeholder, individuals with a drone authorization, and sea rescuer, two initial personas were created. The personas were later improved after the usability testing of the high-fidelity prototype, see 6. Test High-Fidelity Prototype, was performed. This was done to provide a broader description of a typical user.

4.1.3.2 Results

Sarah is 37 years old and conducted The Swedish Transport Agency's training A1/A3 and A2 within the open category 3 years ago. Since Sarah acquired her drone authorization, she has been flying drones 1-2 times per month. The purpose of the flights, which has been conducted with her fixed-wing drone, has mainly been to take photos or record material for personal projects. Sarah has been a member of the Swedish Sea Rescue Society since 6 years back and when she heard about their "Eyes on Scene"-project, she decided to obtain a drone license within category specific.

Per is 52 years old and has been working as a commercial pilot for around 20 years. He has always had a strong interest in aviation, however Per had no experience of flying drones until around 2 years ago when he decided to obtain a drone license. Per conducted The Swedish Transport Agency's training A1/A3 within the open category and owns a multirotor drone which he now flies with every weekend. Moreover, Per has been a rescue leader at the Swedish Sea Rescue Society for about 15 years and participated in numerous marine rescues.

4.2 Product Requirements

In this section, product requirements regarding drone control and design principles are determined based on the data collected. The drone control needs laid the foundation for what design principles to investigate, and from which additional requirements were established. Finally, all of the needs were prioritized using MoSCow Analysis, see 3.2.4 MoSCow Analysis.

4.2.1 Drone Control

This chapter presents how prior data and newly collected data was used in conjunction to determine product requirements for drone control.

4.2.1.1 Method

Initially, product needs regarding drone control were established based on results from prior data collection from potential end users with drone authorizations, see 4.1.2 Data Gathering Potential End User. Furthermore, an unstructured interview was conducted with the stakeholders. A prototype of a drone control interface created by one of the stakeholders was studied before the interview.

The questions asked during the interview were open ended and primarily focused on getting a better understanding of the prototype and drone control in general. Stakeholders' responses and prototype were then used to identify requirements regarding drone control and the interface to be developed. In addition, needs were established from the reports "Obemannade luftfartyg i kommunal räddningstjänst" published by Swedish Civil Contingencies Agency and "Eyes on Scene - EOS" published by RISE Research Institute of Sweden. All of the requirements regarding drone control were compiled in a list and categorized into the following groups.

- Before Flight
    - Before Control
    - After Control
    - Before & After Control
- During Flight
- Before/During Flight

4.2.1.2 Results

The data gathering activities resulted in a deepened understanding of drone control in general and how the interface is going to be used. Following list contains some of the key points from the data gathering activities.

- Interface shall assist the operator to plan the flight safely
- Interface shall provide the operator with the possibility to continuously being able to determine if the flight is conducted safely
- Interface shall allow the operator to operate fast
- Interface shall be adapted to be used during emergencies

All of the requirements that were identified regarding drone control are listed under the section drone control in Appendix A, see Appendix A. Requirements.

4.2.1.3 Discussion

Based on the results from the data gathering, one can state that it is essential that the interface is adapted to be used in emergency situations and allows the operator to operate both safely and quickly. Safely meaning the operator is able to make sure that the drone does not harm any other actors. Quickly meaning that the operator is able to control the drone so that it can be at the specified location of the alarm as soon as possible. Thus, design guidelines that support these requirements were to be investigated.

4.2.2 Design Principles

This section describes how needs were identified from design principles regarding usability and display design for emergency situations, see 2.1 Usability and 2.2 Display Design Emergency Situations.

4.2.2.1 Method

In accordance with theory discussed in 2.1 Usability, usability involves how safely and efficiently an operator can use an interface. Additionally, usability has an impact on how many tasks an operator is able to do within a specified time frame. Thus, a high usability would align with the stated needs regarding that interface has to be safe and quick to use. Therefore, theory about usability was used to identify requirements for the interface.

Moreover, the interface is to be used in an emergency situation. Hence, design principles about display design for better decision making in emergency situations were studied. This theory was then used to determine additional requirements regarding this matter to make the interface adapted to be used during emergency situations.

4.2.2.2 Results

Following needs regarding usability were established from theory presented in 2.1.1 Usability Heuristics and 2.1.2 Error Prevention.

- Interface shall keep the user informed about the system's state
- Interface shall follow real-world conventions
- Interface shall provide user freedom and control
- Interface shall use standards and be consistent
- Interface shall contain error prevention
    - Interface shall constrain types of input
    - Interface shall provide contextual suggestions
    - Interface shall have reasonable defaults
    - Interface shall include lenient formatting
    - Interface shall match the user's mental model

    - Interface shall implement accepted design conventions
    - Interface shall communicate how it is intended to be used
    - Interface shall preview results before user take action
- Interface shall minimize the actor's memory load
- Interface shall provide flexible and efficient processes, like shortcuts
- Interface shall have a minimalistic design and only contain relevant information
- Interface shall contain error messages that help users "recognize, diagnose and recover"
- Interface shall contain documentation that helps users complete their intended tasks

Following requirements regarding display design for better decision making in emergency situations were established based on the design guidelines presented in 2.2 Display Design Emergency Situations.

- Interface shall reduce distribution of information over various sources
- Interface shall associate new information to the data that is presently being handled
- Interface shall present an overview on a single display by using integrated information formats
- Interface shall present changes over time of visual elements
- Interface shall use detachable configural windows when integrated windows are not pragmatic
- Interface shall have a logical semantic mapping when using configural displays
- Interface shall use principles of the ecological interface design to simplify fulfillment of the above guidelines
- Interface shall avoid displays requiring data transformation when the user under most stress
- Interface shall display simple graphics to deliver direct instructions on a low level when the user is under most stress
- Interface shall minimize "direct structural interference" when the user is experiencing a lot of stress

4.2.3 Requirements Analysis

The complete list of requirements can be seen in Appendix A. Requirements. These needs were analyzed using theory presented in 3.2.4 MoSCow Analysis. In this chapter the highest prioritized needs are presented. The complete requirements analysis can be seen in Appendix B. Requirements Analysis.

4.2.3.1 Method

After compiling all requirements for the interface, MoSCow Analysis was used to prioritize the stated needs. As multiplied data sources were used to establish product needs, features that were identified several times generally were assigned a higher prioritization. In addition, all requirements got an identifier to facilitate tracking and referencing.

4.2.3.2 Results

Following table includes all needs that were assigned the highest priority, "Must have", in MoSCow Analysis.

Table 1. All requirements prioritized as highest priority "Must have" in MoSCow Analysis

Identifier	Description	Priority
FEAT 1	Interface shall provide the possibility of taking control of drone	Must have
FEAT 3	Interface shall include a representation of a fixed wing drone	Must have
FEAT 18	Interface shall be able to indicate which route is active	Must have
FEAT 21	Interface shall display the drone's state ● in launch box (before flight) ○ under control ○ not under control ● on ground (before flight) ○ under control ○ not under control ● flying (during flight)	Must have
FEAT 22	Interface shall display the launch box's state ● ready to launch (before flight) ● not ready to launch (before flight) ● launched (during flight)	Must have
FEAT 23	Interface shall display airspace map (before & during flight)	Must have
FEAT 24	Interface shall allow the operator to move around the map (before & during flight)	Must have

FEAT 28	Interface should provide a possibility for the operator to set a new GoTo-point manually ● when the drone is not launched yet (before flight/after control) ● when the drone is flying towards a GoTo-point (during flight)	Must have
FEAT 29	Interface shall provide a possibility for the operator to change the GoTo-point ● when the drone is not launched yet (before flight/after control) ● when the drone is flying towards a GoTo-point (during flight)	Must have
FEAT 39	Interface shall contain drone's GPS position (before & during flight)	Must have
FEAT 49	Interface shall allow the operator to operate fast	Must have
FEAT 50	Interface shall be adapted to be used during emergencies	Must have
FEAT 65	Interface shall match the user's mental model	Must have
FEAT 68	Interface shall preview results before user take action	Must have
FEAT 70	Interface shall make the user confirm a destructive action	Must have
FEAT 71	Interface shall warn the user about errors before they are made	Must have

5. Ideate and Prototype

In this chapter, methodology described in 3.3 Ideate and 3.4 Prototype has been used in order to generate ideas and prototypes. All of the prototyping was performed in Figma since it supports creating designs for digital products, including adding interactivity, see 1.5.1 Figma. Procedures and methods described in 3.3.1 Internal Search and 3.3.2 External Search have been used to identify solutions.

5.1 Method

Initially, the highest prioritized requirements, i.e needs that were assigned the priority "Must have", in 4.2.3 Requirements Analysis was given one or several brainstorming sessions. Some of the needs have been brainstormed individually and some as a group with other dependent needs.

Furthermore, external search was conducted to identify existing solutions to stated subproblems. The external search has been mainly focused on functionality within already existing map services such as Google Maps or Apple Maps. Solutions that were identified in the external search were combined with solutions from the brainstorming sessions.

Solutions have been generated and visualized in prototypes in three different sessions that have been focused on different concepts. Interactivity has been added to the prototypes when it has been applicable, allowing them to become more high-fidelity. When a final solution and concept was identified, a final high-fidelity prototype was created to be used in user testing.

During the whole ideate and prototype phase, continuous meetings with the supervisor and stakeholders of this degree project have been held. These meetings have primarily had the objective of evaluating different solutions, concepts, and prototypes.

Throughout this stage, Figma has been used to create illustrations of concepts and prototypes. This has been done by creating multiple page design files where shapes, images, texts, and layers have been used to visualize elements within the interface. If a component consists of several elements, for example a polygon and an ellipse, the elements have been grouped. Additionally, all elements and groups were named. Both of these actions facilitated keeping a clear structure among the design files.

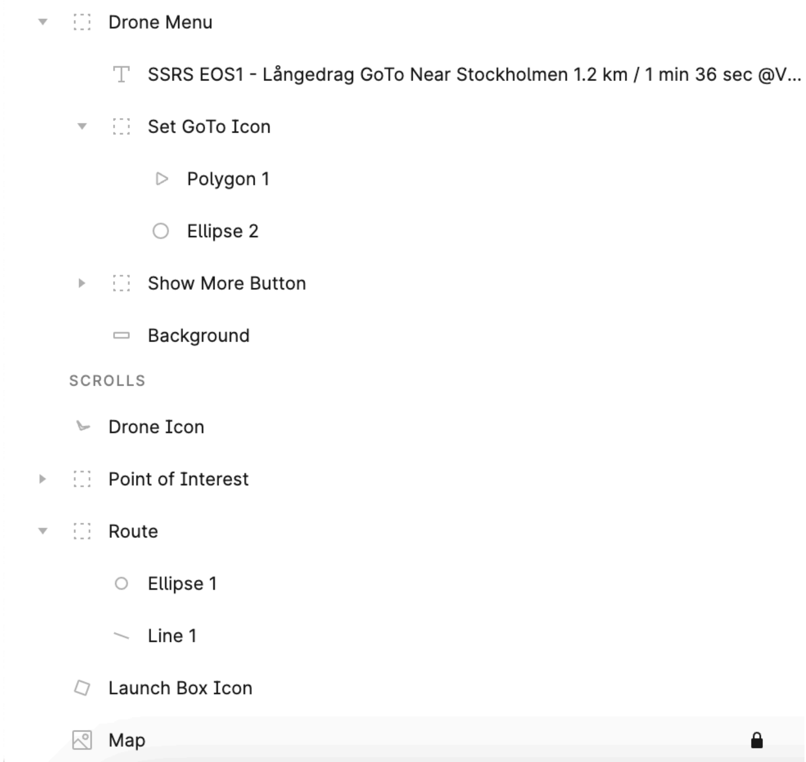

Figure 6. Individual and grouped elements that have been named when prototyping in Figma

When modeling the more high-fidelity prototypes the frames have been connected to one another through transitions. These transitions make it possible to change the page in view depending on the user's interaction with the prototype. Additionally, Figma's functionality of sharing designs has been used. This facilitated receiving feedback from stakeholders since they received personal access to the designs.

5.2 Results

This section contains images and descriptions of the results from the concept generation sessions and final high-fidelity prototype. Design decisions have primarily been based on theory presented in 2.3 Design Theory and stated product requirements in 4.2 Product Requirements. More in depth descriptions on design decisions are presented later in the report in 5.2.2 High-Fidelity Prototype and 7. Final Product Development.

5.2.1 Concept Generation

In this chapter, a selection of prototypes and visualizations from the three concept generation sessions are presented.

5.2.1.1 Drag and Drop, Icons and Moving Menus

The first concept generation was initiated with generating icons for the launch box and fixed-wing drone. Starting point was an image of a 3D-model of "EOS-vinge", seen in Figure 7. The following images are some of the generated drone and launch box icons and are presented from left to right in the same order as they were created.

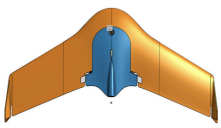

Figure 7. 3D-model of fixed-wing drone "EOS-vinge" from Eyes on Scene

From *Eyes on Scene - EOS.* [Image], by Sanchez-Heres, L., Falkman, F., Forsman, F., Hagner, O., Nilsson, V., Lundqvist, R., Bergman, K., Bjurling, O., Granlund, R., Hjalmarsson, M., & Brunnström, M., 2021. RISE Research Institutes of Sweden.

Last accessed 2024-04-11 from https://fudinfo.trafikverket.se/fudinfoexternwebb/Publikationer/Publikationer_007401_007500/Publikation_007477/eos-slut-rapport.pdf

Figure 8-14. Some of the generated fixed-wing drone and launch box icons

Moreover, the launch box and fixed-wing icon were integrated in a context where more features, such as buttons, information about the drone, and wind conditions were included. In accordance with theory presented in 2.3.2 Buttons, the buttons were modeled with a shadow-effect to indicate that they are interactive. Additionally, buttons were designed to indicate whether they contained action or information. As for example in Figure 15. “Take Control”-button is modeled in orange.

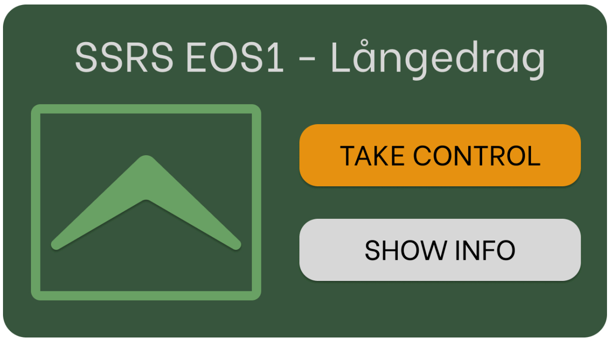

Figure 15. Drone and launch box icon with a “Take Control”-button, “Show Info”-button and drone’s name

The menus were later placed on an image of a Mapbox map, see 1.6.4 Mapbox.

Figure 16. Various drone menus placed on Mapbox map

Thereafter, interactive prototypes that included the drone's route and the possibility to set and change a GoTo-point were created. These prototypes were specifically focused on how to change the GoTo-Point. For example, one of them visualized a solution where the user could click at any location in the map and was offered an option to change the GoTo-point to that location.

Another solution involved that the user could drag and drop the GoTo-Point to a new location in order to change it. In turn, the drag and drop-solution involved two different solutions, one where the user had to enter a “Change” state by clicking on a button to be able to drag and drop the GoTo-Point. The second solution was one where the user could, without entering any specific state, drag and drop the GoTo-Point to a new location.

The following two images demonstrate the solution involving the “Change” mode. In Figure 17, the user has not entered the “Change” state, i.e has not clicked at the “Change GoTo-Point " button, and hence is not able to change the GoTo-Point. Figure 18 demonstrates that the user has clicked the “Change GoTo-Point” button, entered the “Change” mode, and therefore can drag and drop the GoTo-Point to a new location in order to change it.

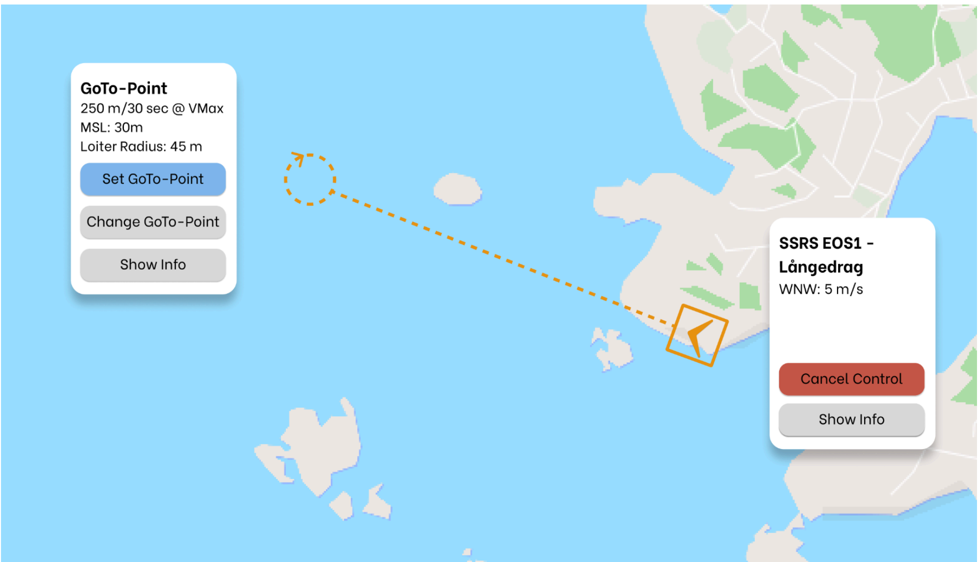

Figure 17. Drag and drop solution before entering “Change” mode visualized in Figma

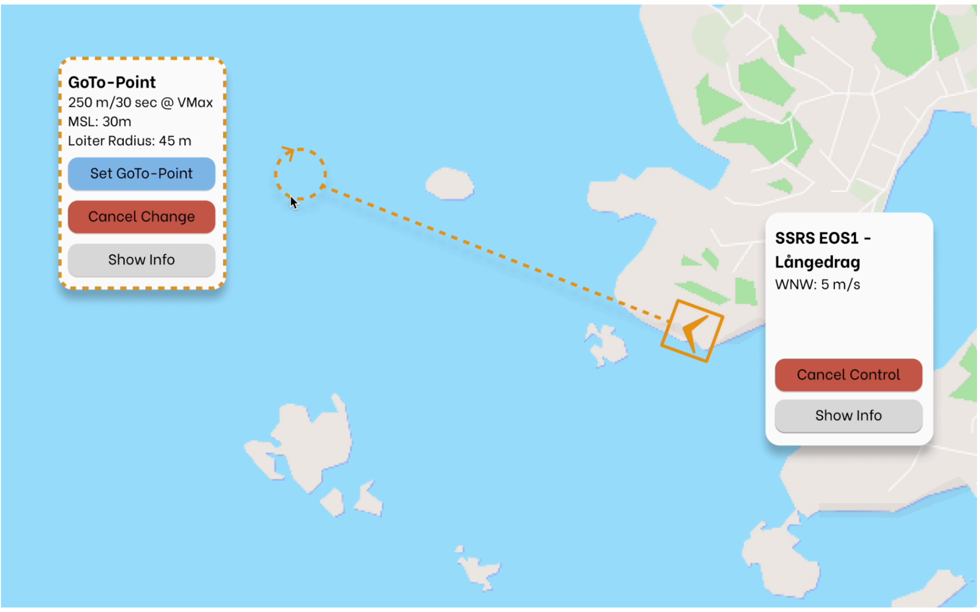

Figure 18. Drag and drop solution in "Change mode" visualized in Figma

In all of these prototypes, the drone's menu moved within the map along with the drone. All information regarding the GoTo-point was in a separate menu that had a constant position next to the set GoTo-Point.

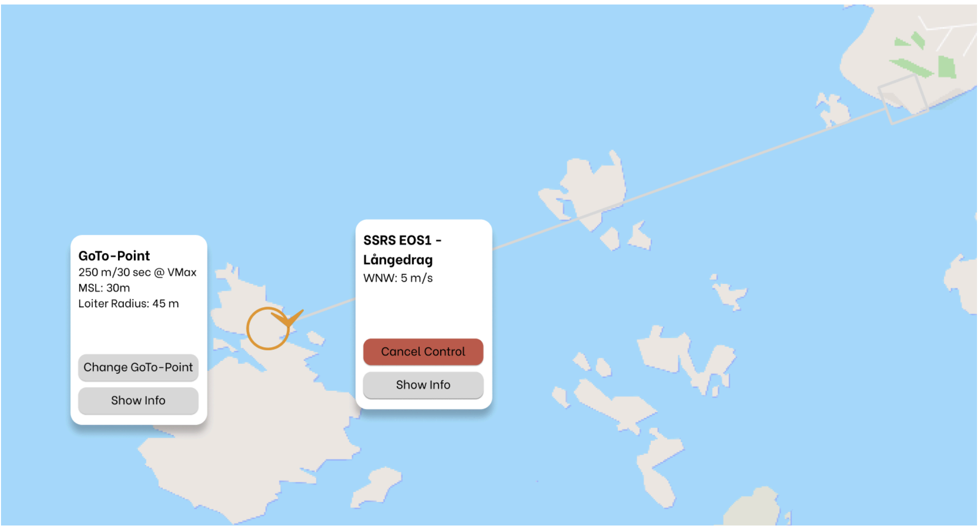

Figure 19. Drone loitering around GoTo-Point, associated drone's menu has moved within the map along the drone's route visualized in Figma

5.2.1.2 Point of Interest, Context Menu, and Buttons with Icons

In concept generation 2, Point of Interest was introduced and a new map style, provided by the stakeholder, was applied. Additionally, steppers to change input fields were added to let the user recognize that values can be changed.

The drone and GoTo-Points menus were merged after taking control of the drone. This menu, along with the newly introduced Point of Interest's menus were placed in the upper left corner. Thus, none of the menus were no longer placed next to the icon that they were associated with. This change was believed to make it more difficult to identify the location of the GoTo-Point and Point of Interest. To solve this issue, two interactive arrows were added next to the menus. These arrows contained functionality that would facilitate locating the Point of Interest and GoTo-Point. Firstly, the arrow is pointing towards the associated point. Secondly, when clicking on any of the arrows, the user's window is automatically moved to the location of the associated point.

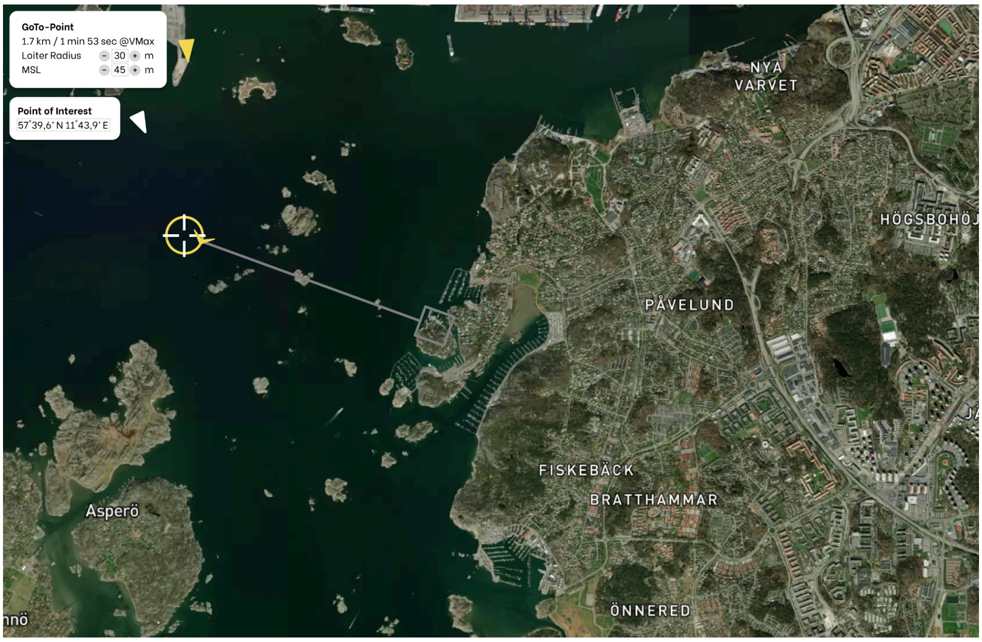

Figure 20. GoTo-Point and Point of Interest menus placed in the upper left corner

From external search and discussions with the stakeholder, a solution where the user right clicks to display a context menu were identified. When right clicking within a Google Map, a context menu is displayed. The context menu contains several options where one of them is to set directions to that point, see Figure 21.

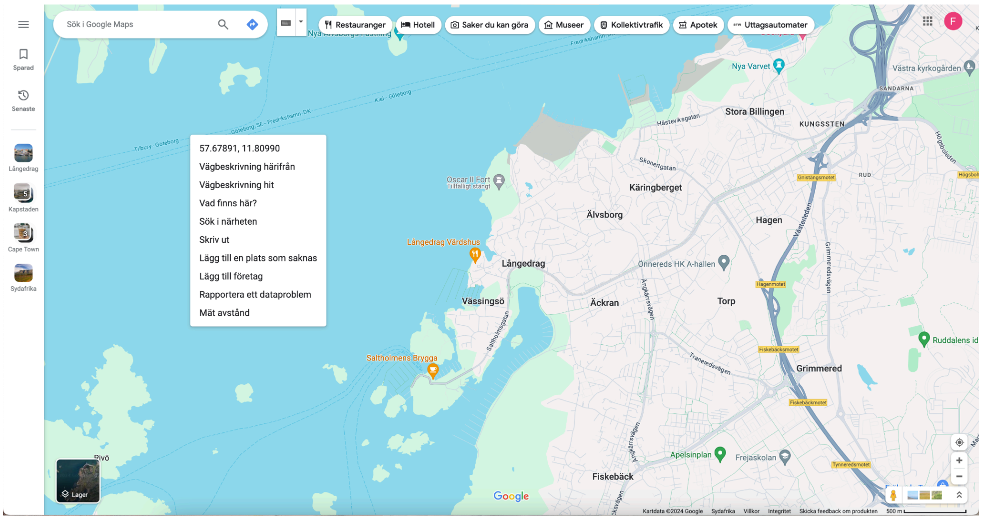

Figure 21. Context menu displayed after right click within Google Maps

From *Google Maps*. [Screenshot], by Google. 2024. Last accessed 2024-04-12 from
https://www.google.com/maps/@57.647925,11.8205988,13z?entry=ttu

Moreover, it was identified that Google Maps has vertical floating action buttons with icons at the top of the map and horizontal navbar with additional buttons with icons to the left.

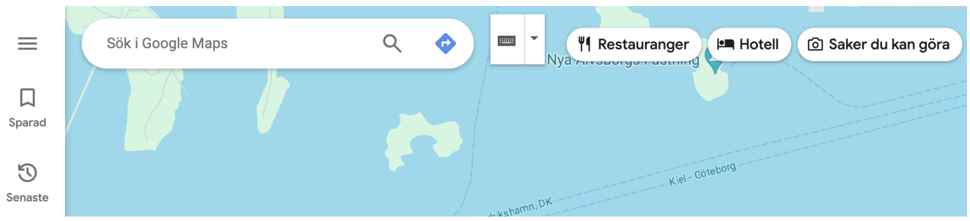

Figure 22. Google Maps's floating action buttons and navbar

From *Google Maps*. [Screenshot], by Google. 2024. Last accessed 2024-04-12 from
https://www.google.com/maps/@57.647925,11.8205988,13z?entry=ttu

This initiated a brainstorming session of similar buttons, see Figure 23-33.

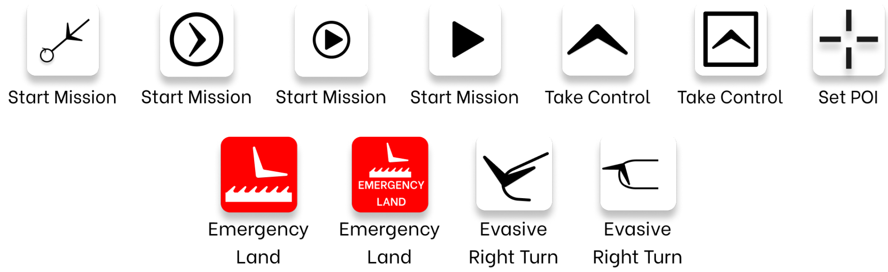

Figure 23-33. Brainstorming of icon buttons modeled in Figma

The “right click for context menu”-solution and new buttons were then integrated into the prototypes and different versions of menus were visualized, see Figure 38-40.

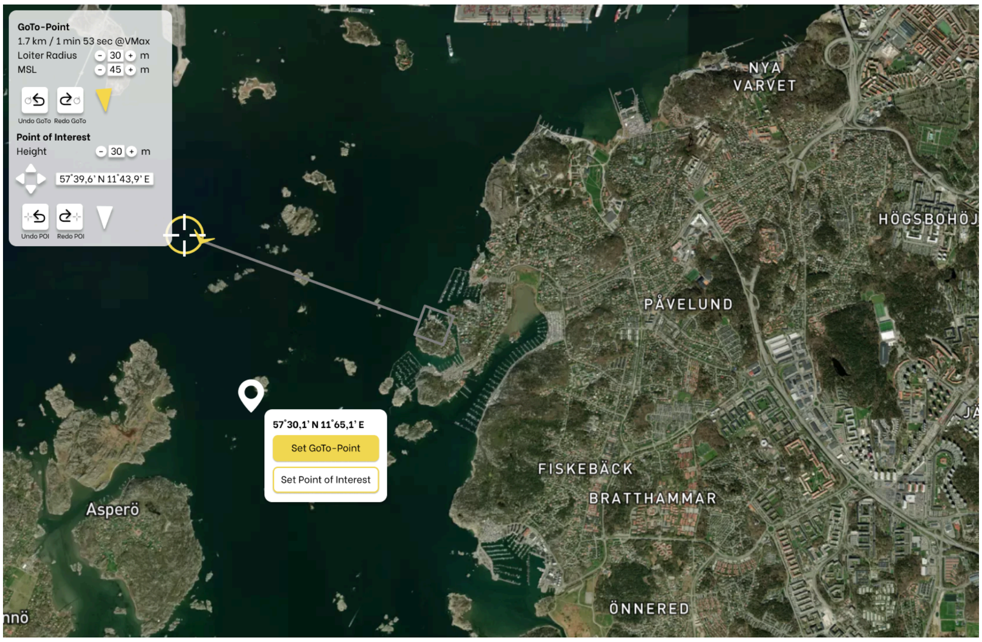

Figure 34. Buttons with icons integrated in menu and context menu after right click visualized in Figma

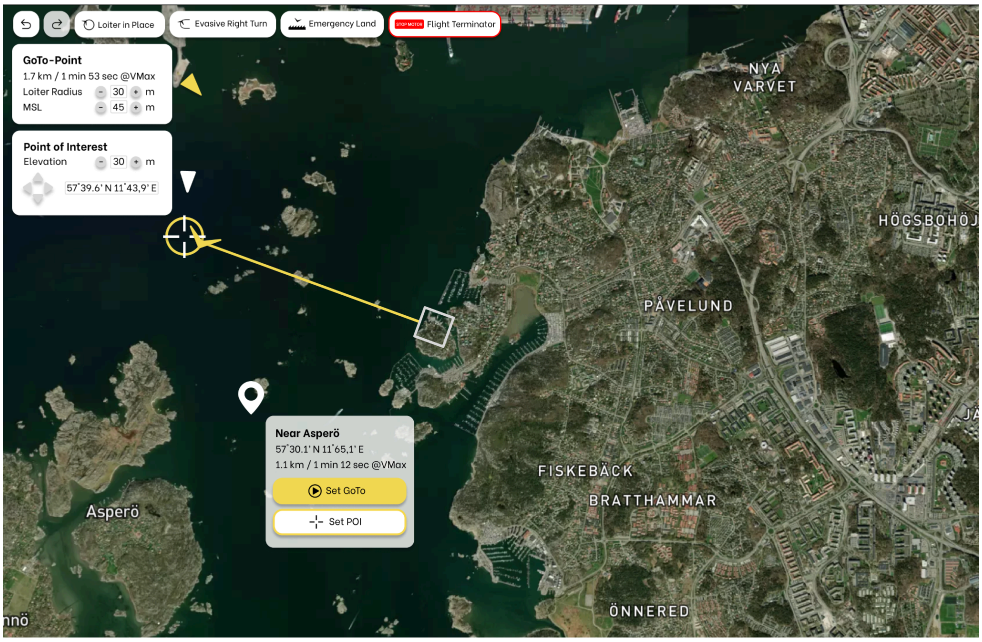

Figure 35. Buttons with icons placed horizontally at the top of the map and context menu after right click vizualised in Figma

Lastly, an additional brainstorm session of the fixed-wing drone icon was performed. The aim was to make it look more detailed and less like an arrow according to feedback from the supervisor. Starting point was again the 3D-model of "EOS-vinge".

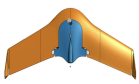
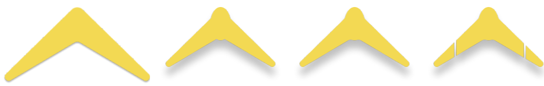

Figure 36. 3D-model of fixed-wing drone "EOS-vinge" from Eyes on Scene

From *Eyes on Scene - EOS.* [Image], by Sanchez-Heres, L., Falkman, F., Forsman, F., Hagner, O., Nilsson, V., Lundqvist, R., Bergman, K., Bjurling, O., Granlund, R., Hjalmarsson, M., & Brunnström, M., 2021. RISE Research Institutes of Sweden.

Last accessed 2024-04-11 from https://fudinfo.trafikverket.se/fudinfoexternwebb/Publikationer/Publikationer_007401_007500/Publikation_007477/eos-slut-rapport.pdf

Figure 37-40. More detailed fixed-wing drone icons

5.2.1.3 Waypoints, Hamburger Menu, and Several drones

In concept generation 3, solutions including choosing several locations that the drone should fly to (waypoints), see Figure 41, and several drones in one map were visualized. Figure 41. illustrates a solution where the user can set both GoTo-Point and waypoints.

Figure 41. Visualization in Figma of how to incorporate waypoints

Moreover, visualizing several drone menus included deciding how these should be displayed and how to indicate which one of them was being currently controlled. Since right click was used to display a context menu and options to set GoTo-Point and Point of Interest, it was necessary to find a solution to switch between which drone the user is setting a GoTo-Point for.

Figure 42 contains one solution where the currently controlled drone's menu is the middle which has all of its information displayed, a white background and a blue border. The non-active drone menus only contain information about the drone's name, the current GoTo-point of that drone and the time and distance to the GoTo-Point. In this way the operator can see if a drone has arrived at its GoTo-Point without changing the location in the map or switching menu. To switch menus the user could click at a non-active menu, and then this menu would be displayed as the active or the currently controlled one.

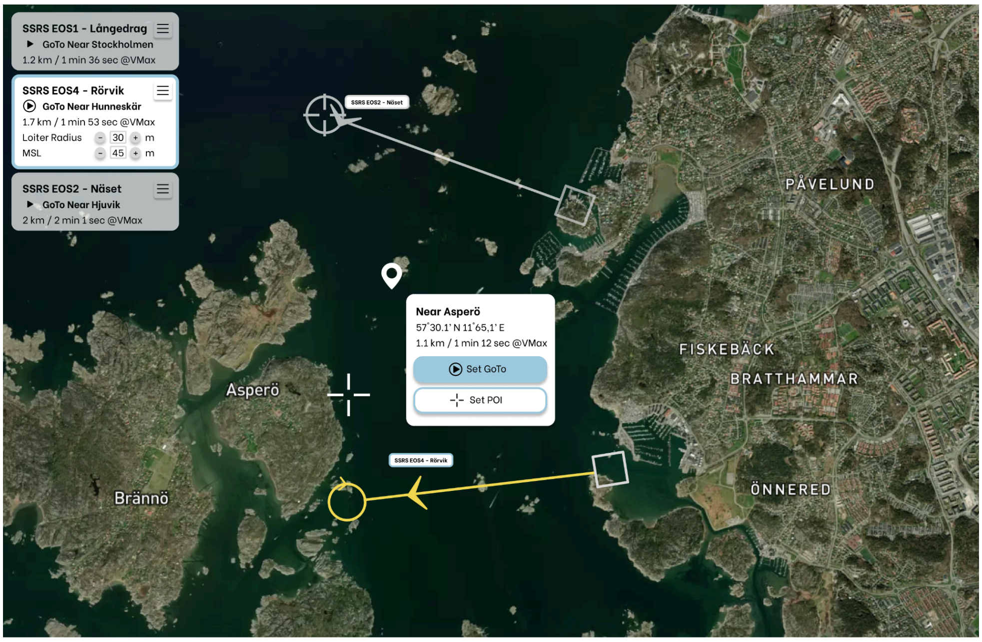

Figure 42. Visualization in Figma how several drones and their associated menus included in the map

Furthermore, according to requests from stakeholders the menus were modeled to take up as small part of the interface as possible and colors were chosen according to a "dark mode" theme. To minimize the space that the menus used on the screen, two different solutions were illustrated. One where the emergency functions were placed in a hamburger menu in the drone's menu and another one where they were placed in the context menu. See Figure 43. and Figure 44. respectively.

Figure 43. Drone menu with hamburger menu that contains additional functions visualized in Figma

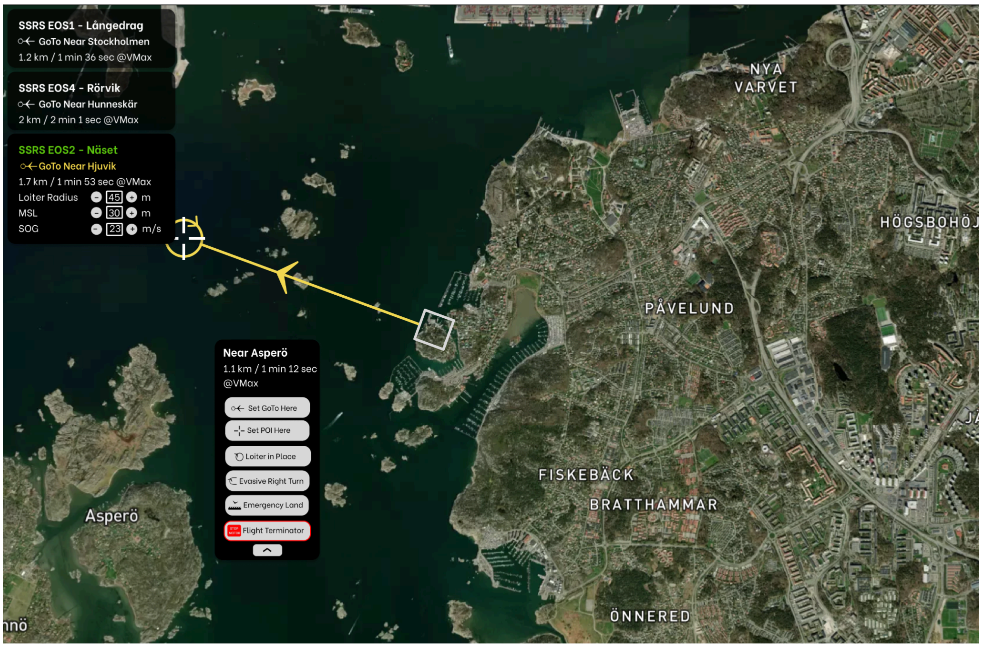

Figure 44. Colors according to "Dark mode", several drone menus, and a context menu that could be extended to view more options

The following Figures 45-47, contains screenshots from an interactive prototype with two drones that illustrates a solution of when no drones are in control, when one drone is in control and when both drones are in control. The routes, drones and launch boxes icons are provided by the stakeholder.

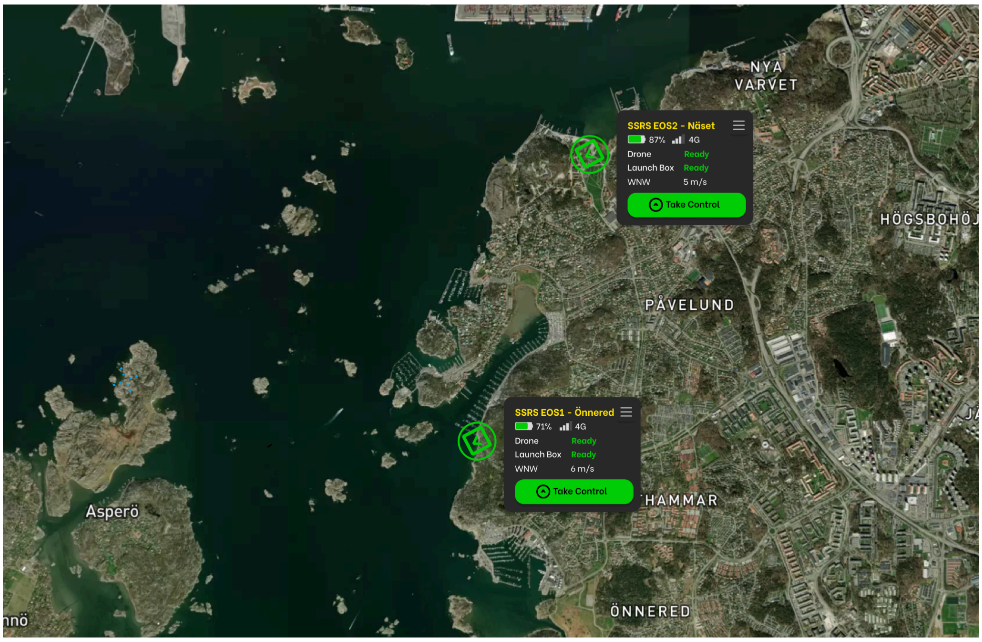

Figure 45. Prototype in Figma where several drones are included, no drones in control

Figure 46. Prototype in Figma where several drones are included, one drone in control

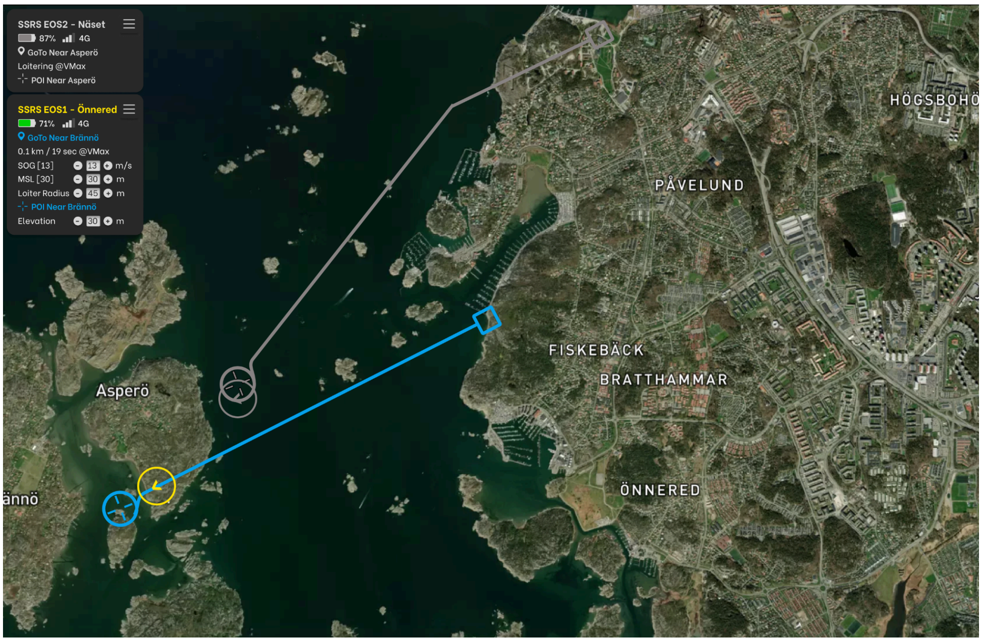

Figure 47. Prototype in Figma where several drones are included, two drones in control

5.2.2 High-Fidelity Prototype

The aim of creating this high-fidelity prototype was to develop a prototype that was high-fidelity enough to be used in a usability test where the users could perform specified tasks within the system.

In the initial view, there are three ways the user can interact with the prototype and which are as follows.

- Drag in the map to move around the map
- Click at the hamburger menu to view more options
- Click at the “Take Control”-button to take control of a drone

Depending on what actions the user takes the prototype will output different outcomes. If the user chooses to take control of a drone, they are exposed to additional ways of interacting with the system, such as deciding where the drone should go by setting a GoTo-Point and pointing the drone’s camera by setting a Point of Interest.

The high-fidelity prototype only contains one drone. However, as stated in the 1.3 Problem Description, the interface is intended in the future to contain several drones among which a single operator can choose which drone to control and potentially control several drones at the time from. Thus, the prototype will be discussed as if it contained numerous drones.

In the initial view, each drone is placed within its launch box and has a related menu next to it. The menu contains information such as the drone’s name, battery, coverage, state, launching box state and wind conditions for the location of the drone.

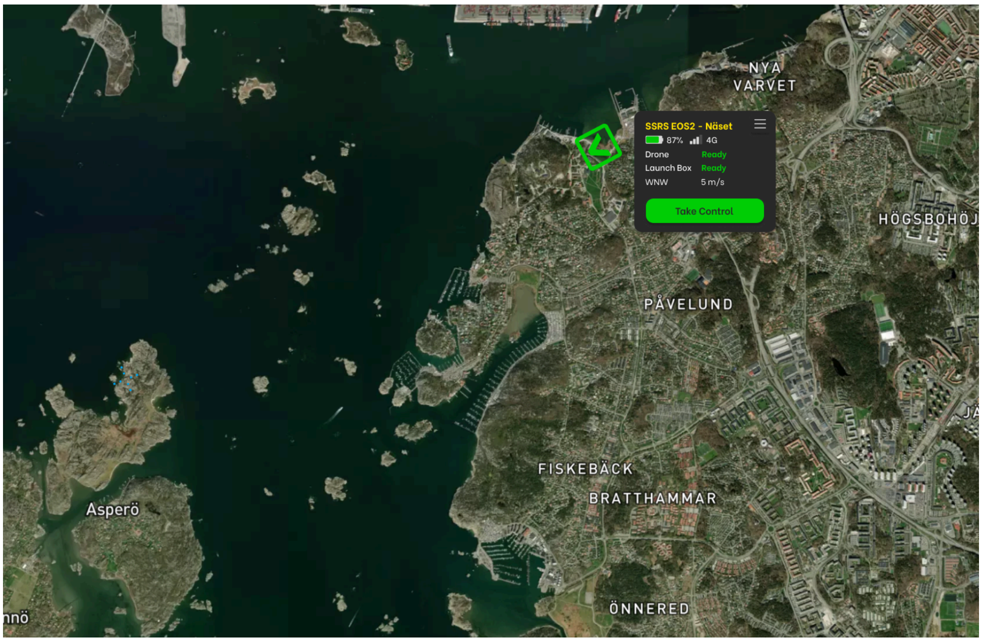

Figure 48. Initial view of high-fidelity prototype modeled in Figma

Drone menu contains a "Take Control"-button that has been designed based on applicable guidelines presented in 2.3.2 Buttons. The "Take Control"-button is designed as filled with rounded borders and a shadow effect. Rounded corners are chosen to direct the operator's focus to the center of the button where it says "Take Control".

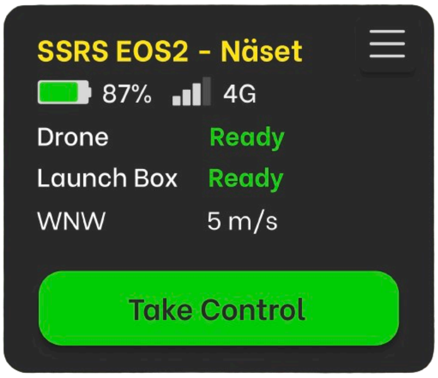

Figure 49. Drone menu including drone's name, battery, status, coverage, launch box's state, wind conditions, take-control-button

Moreover, the “Take Control”-button is designed as a Call-to-Action button to convey that this is a key feature of the interface. This was done by modeling the button with a highly contrasting color, see 2.3.1 High Contrast Colors. The chosen color of the “Take Control”-button had a contrast ratio of 6.47:1 towards the background.

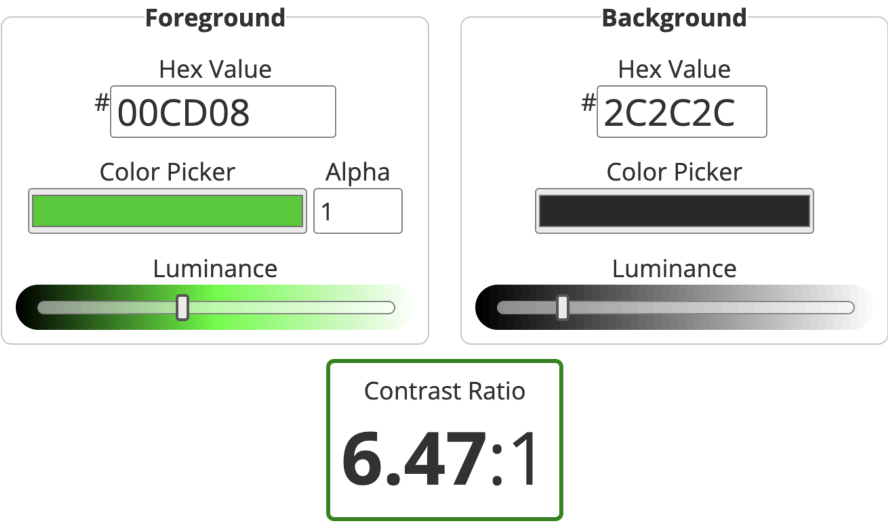

Figure 50. Contrast ratio 6.47:1 between the “Take Control”-button’s color and background color

From *Contrast Checker*. [Screenshot], by WebAIM. (https://webaim.org/resources/contrastchecker/)

Furthermore, the drone menu contains a hamburger menu which in turn contains two buttons, one to display the drone’s history and one to display more information about the location of the drone.

In the initial view, there are three ways in which the user can interact with the prototype, drag in the map to move around, click at the hamburger menu or the “Take Control”-button. When clicking at the hamburger menu, the “Show Info” and “Show History” buttons are displayed. These are modeled in gray to indicate that these buttons, within this context, probably are less desirable for the operator, see 2.3.2 Buttons.

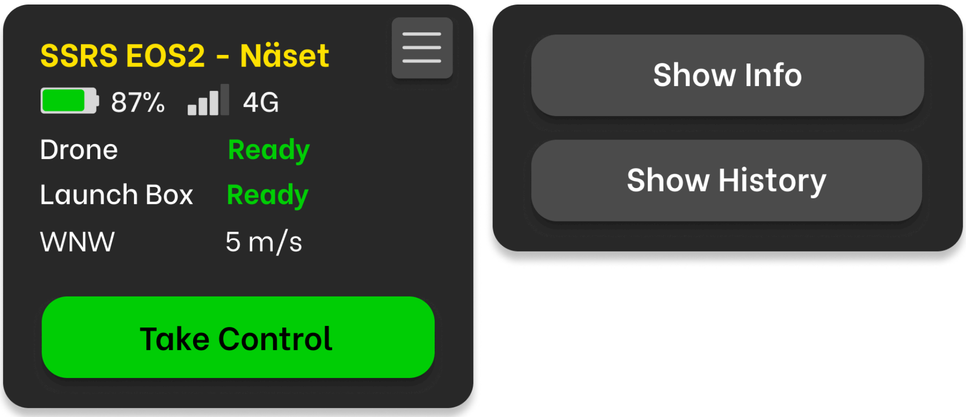

Figure 51. Drone menu and hamburger menu with buttons "Show Info" and "Show History"

When the user clicks at the "Take Control"-button, the drone turns yellow in order to demonstrate that the drone has entered a new state, according to FEAT 51 regarding that the system should keep the user informed about its state. The yellow color represents that the drone is now under control of the operator. For the same reason, the launch box also changed color. Furthermore, the drone's menu is moved from next to the drone to the upper left corner.

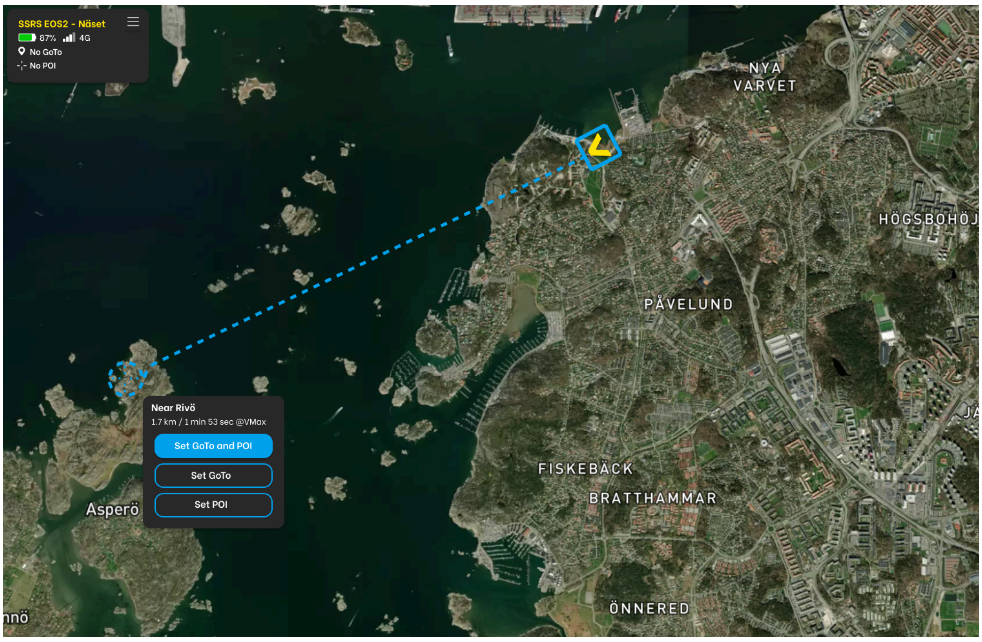

Figure 52. An automatically suggested route is suggested to the user when pressing “Take Control”

A GoTo-point and Point of Interest is automatically suggested to the operator. Both of them are designed in a dashed line to indicate that this is a preview. The suggested location has an associated menu with the location's name, three buttons and time and distance for the drone in control to fly to that destination. The menu containing information about the location has been positioned next to the location itself in order to associate it to the location, in accordance with FEAT 73.

To accept the suggested GoTo-point and Point of Interest the user can click at the “Set GoTo-point and POI”-button. The button is designed as a Call-to-Action button by modeling it in a highly contrasting color, see 2.3.2 Buttons.

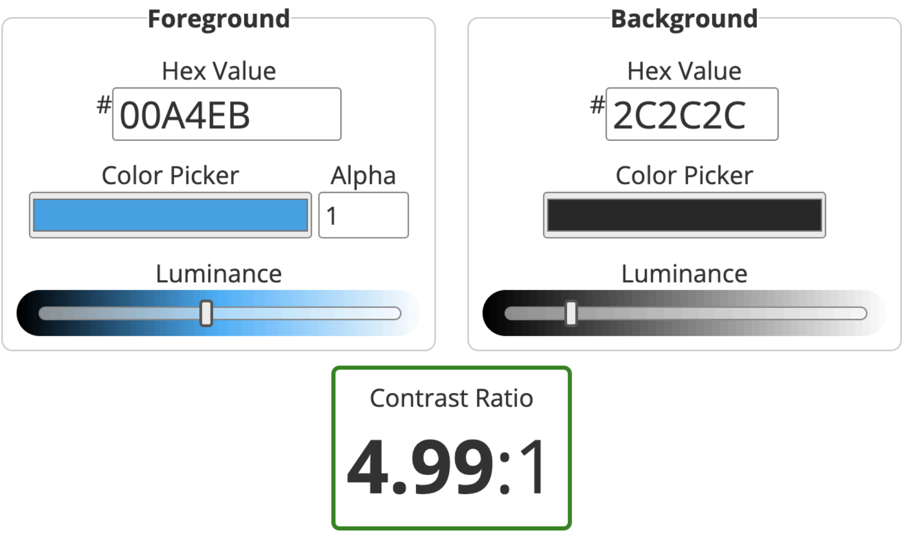

Figure 53. Contrast ratio 4.99:1 between the "Set GoTo and POI"-button's color and background color

From *Contrast Checker*. [Screenshot], by WebAIM. (https://webaim.org/resources/contrastchecker/)

Below the "Set Goto and POI" button there are two other buttons "Set GoTo" and "Set POI". These buttons can be used if the operator only wants to accept either the GoTo-Point or Point of Interest. Both of these are modeled as ghost buttons in order to convey to the operator that these buttons are of secondary importance to the "Set GoTo and POI" button, see 2.3.2 Buttons. This was done by modeling the buttons with no fill and a border. All buttons within the interface have been modeled with a shadow effect in order to convey to the operator these objects stand out and can be interacted with.

Figure 54. Location menu including name of location, time and distance to location, "Set GoTo and POI"-button, "Set GoTo"-button and "Set POI"-button.

If the user chooses to accept the suggested route by pressing, the dashed lines turn solid to indicate they no longer are previews. Moreover, additional information are added into the drone's menu such as speed, mean sea level, drone settings, and its chosen GoTo-point and Point of Interest.

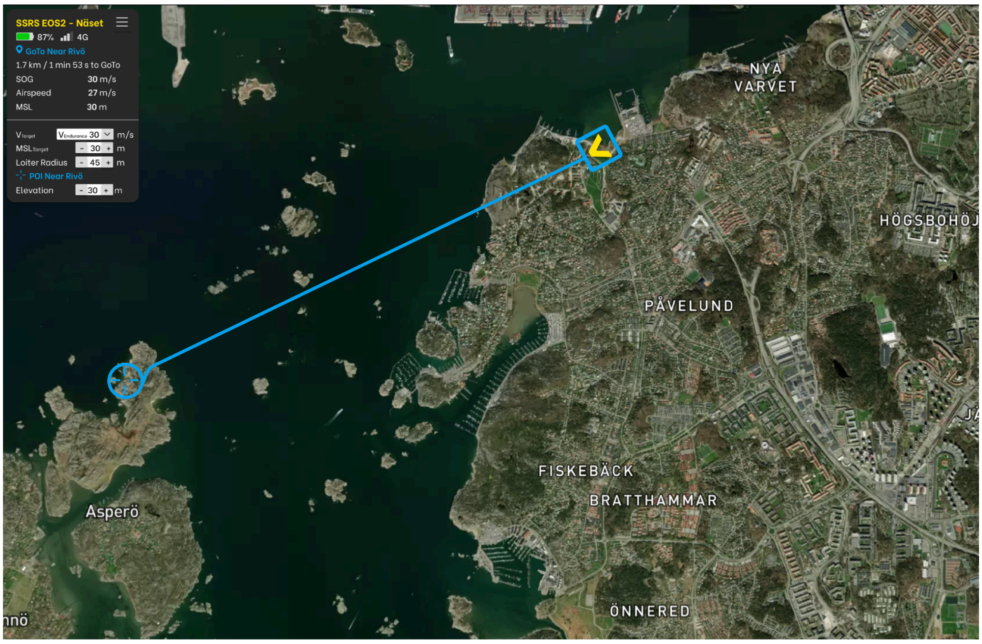

Figure 55. High-fidelity prototype modeled in Figma, drone has an active GoTo-point and Point of Interest

In order to choose another GoTo-point or Point of interest, the user can right click anywhere in the map. If the user right clicks, a previewed route is drawn from the drone's current location to where the user clicked. The location's associated menu is again displayed next to it. As stated, this solution was identified in Google Maps and was chosen to increase the users' learnability of the interface, in accordance with FEAT 54.

Due to limited functionality in Figma, such as no distinction between right click and left click, the prototype was not able to function this way. Instead the user could only change GoTo-point and Point of Interest in certain states by left clicking in the map. Additionally, the proposed route was not drawn to where the user clicked, however, to a predetermined location.

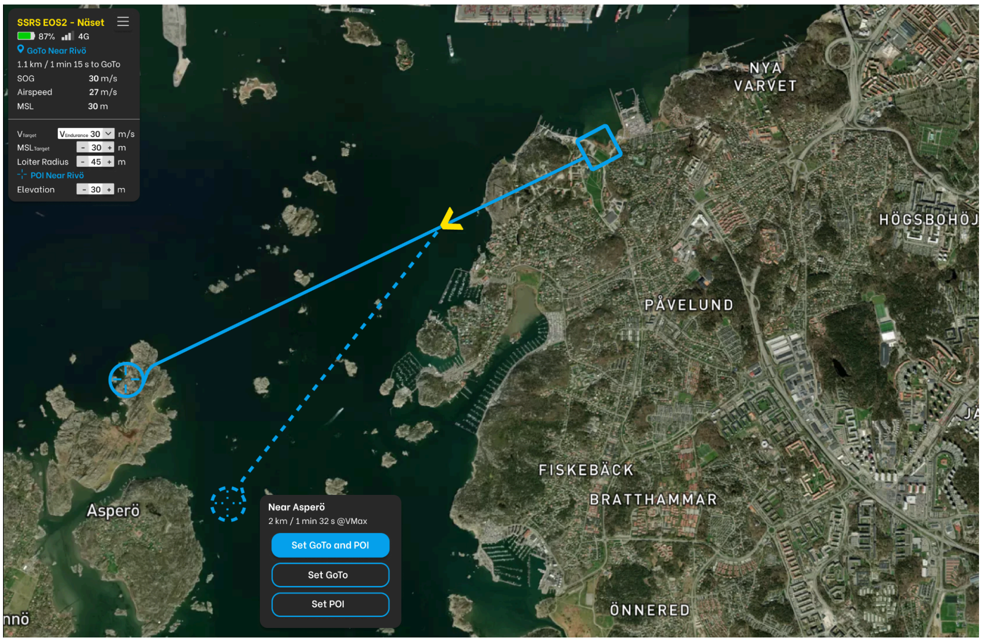

Figure 56. High-fidelity prototype modeled in Figma, changing GoTo-point and Point of Interest

If the user chooses to only change the GoTo-Point or Point of Interest by pressing either the "Set GoTo-button" or the "Set POI"-button, the Point of Interest and GoTo-point would no longer share location. In this case, a thin dashed line drawn from the drone to the Point of Interest appears. This line is functioning as a line of sight to the Point of Interest to make it possible for the operator to easily locate the active Point of Interest.

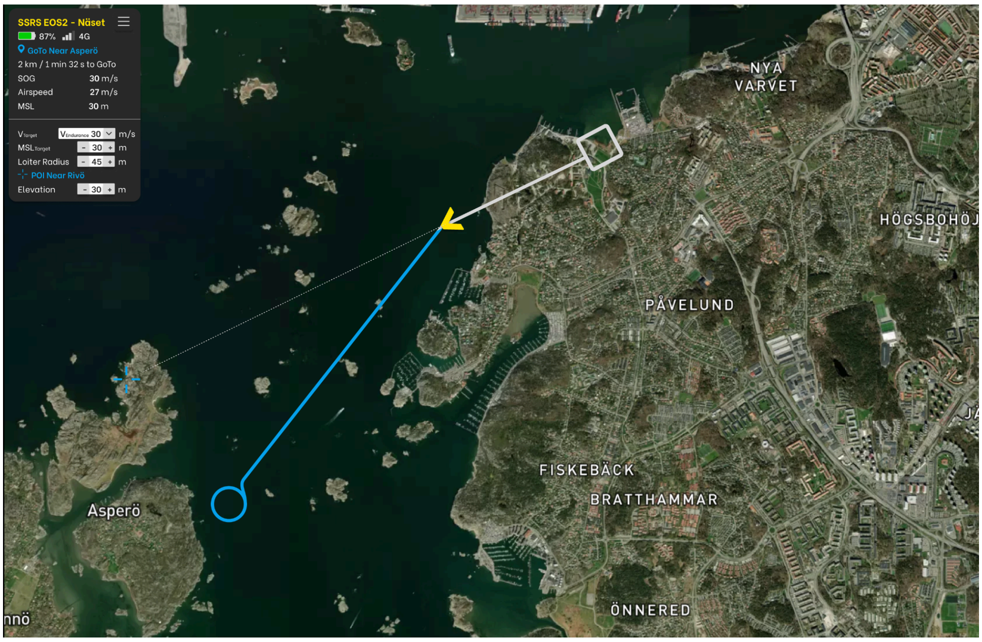

Figure 57. High-fidelity prototype modeled in Figma, line of sight between drone and Point of Interest

To indicate what route the drone already has flown and which is upcoming, the route that the drone already has flown successively turns gray. The color choice is to indicate the element's state and that this element is probably less relevant in this context, see 2.3.2 Buttons. Even though the element itself is not a button, gray is chosen to keep internal consistency, according to FEAT 54.

Moreover, this guideline has been applied throughout the interface. For example, elements that are related to the GoTo-point or Point of Interest are kept in the same blue color regardless if the elements are buttons, lines, icons or text. Another example is that additional functions throughout all states can be found by clicking at the hamburger menu. The hamburger menu's position is also consistent in all drone menus to facilitate access to it.

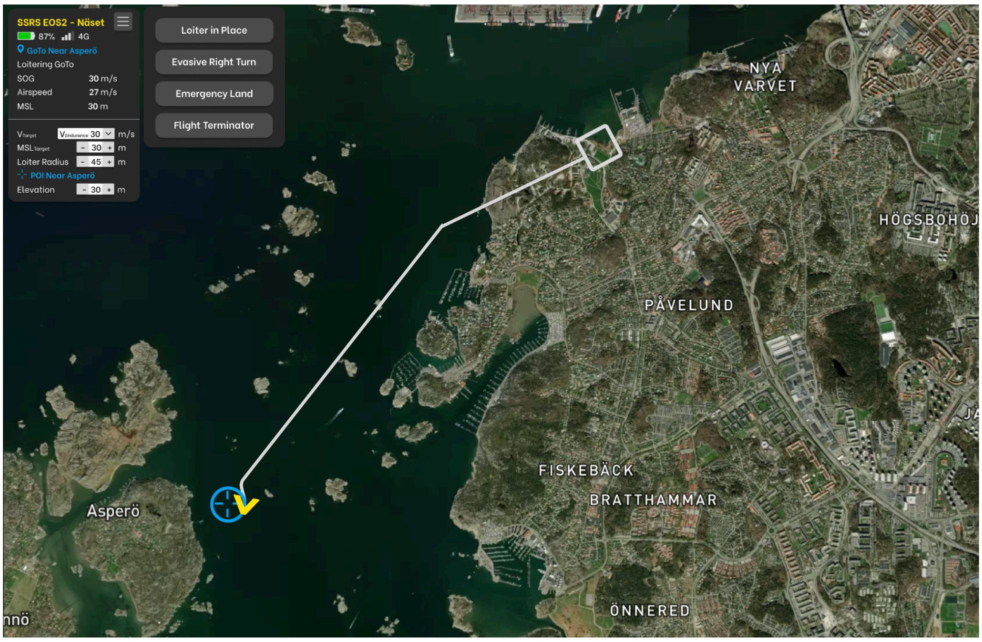

Figure 58. High-fidelity prototype modeled in Figma, hamburger menu with additional functions

All menus have been designed as a vertical left-justified menu where each menu item has the aim of starting with various words to allow rapid scanning, see 2.3.3 Menus. To keep maximum legibility, all caps have been avoided in all places except for abbreviations which have been kept in cases.

Information within the menus has been carefully selected in order to keep a minimalistic design to achieve a higher usability according to FEAT 58. When applicable, information within the prototype has been presented in a single display to create a simple overview, in accordance with FEAT 72 and 74.

The fixed-wing drone and launch box icons used in the prototype have been designed by the stakeholder. In addition, the prototype's colors have been chosen according to a "dark mode" theme. All of the mentioned features have been included due to requests from the stakeholder.

6. Test High-Fidelity Prototype

This section presents the usability testing conducted on the high-fidelity prototype modeled in Figma. Methodologies discussed in 3.5.1 Usability Testing have been used to perform the tests. To analyze the results, methods presented in 3.5.2 Task Completion Rates have been applied.

6.1 Method

To evaluate the high-fidelity prototype, usability testing was chosen as a testing method. As stated in 3.5.1 Usability Testing, the participants in usability testing should be realistic users. Thus, participants that are members in the Swedish Sea Rescue Society were contacted. The participants had varying experiences of flying drones, from no experience to professional experience. However, participants without drone experience had professional experience of aviation, one of them as a helicopter technician and one as an airplane pilot.

The usability testing was performed remotely via Zoom where the test participant was granted remote control over a computer to interact with the prototype in Figma via their own mouse. Test participants were given a brief introduction about how the test was going to be performed, instructions about how to control the prototype, i.e via mouse, and an explanation that the abbreviation POI stands for Point of Interest. However, they were not given an explanation of the meaning of the term.

According to methods described in 3.5.1 Usability Testing, test participants were given specified tasks to perform in the prototype. Moreover, the participants were asked to think aloud when conducting the tasks. Finally, the boomerang and echo technique was being used while monitoring the test.

Test participants were asked to perform the following tasks.

1. Show drone's history
2. Show more information about drone

3. Take control of drone
4. Set GoTo-Point and Point of Interest
5. Explain how you would have increased loiter radius
6. Change GoTo-point and Point of interest
7. Emergency land drone

After the test participants had performed the specified tasks they were asked the below stated following-up questions.

1. What does the gray line symbolize [i.e the route that the drone has already flown]?
2. What is the difference between the dashed and solid line?
3. What does this line symbolize [i.e the line of sight between Point of Interest and drone]?
4. Explain all items within the drone menu (before and in control).

Lastly, the features, functions, terms were explained to the participant and then they were asked these follow-up queries.

1. What would you change if you had the possibility to do so?
2. What is ambiguous?

During the test, notes were taken about participants' mouse clicks, questions during the test, how they completed specified tasks and their answers to the following-up questions.

To quantify the users' success of each task, task completion rate was used, see 3.5.2 Task Completion Rates. The users' performance on each task was categorized into one of the four following categories.

1. **"Complete success"** - Participant performed the correct task in the expected way
2. **"Success with one minor issue"** - Participant succeeded in completing the task but expressed some confusion
3. **"Success with a major issue"** - Participant eventually succeeded in completing the task but either pressed somewhere else at first or expressed great confusion or both
4. **"Failure"** - Participant did not succeeded to complete the task

6.2 Results

The following text is a summary of the most important test results. In order to view the complete test results, see Appendix C. Usability Testing of Prototype. The following table contains the success levels and task completion rates of each task.

Table 2. Levels of success and completion rates for each task in usability testing of high-fidelity prototype

Task/ Level of success	**1. Show drone's history**	**2. Show more information about drone**	**3. Take control of drone**	**4. Set GoTo-Point and Point of Interest**	**5. Explain how you would have increased loiter radius**	**6. Change GoTo-Point and Point of Interest**	**7. Emergency land drone**
Complete success	3	3	2	2	0	0	2
Success with one minor issue	0	0	1	1	0	0	0
Success with one major issue	0	0	0	0	1	3	1
Failure	0	0	0	0	2	0	0
Task completion rate	100%	100%	66%	66%	0%	0%	66%

All of the participants performed "Task 1. Show drone's history" and "Task 2. Show more information about drone" with "Complete success". Thus, the task completion rate of those tasks are 100%. Regarding "Task 3. Take Control Of Drone" and "Task 4. Set GoTo-Point and Point of Interest", there were two participants that performed the tasks with "Complete success" and one with "Success with one minor issue". This resulted in a task competition rate of 66%.

Most of the participants performed "Task 5. Set GoTo-point and Point of Interest" with "Success with a major issue" and one of the participants failed to perform the task. Moreover, all of the participants performed "Task 6. Change GoTo-Point and Point of Interest" with "Success with a major issue". Hence, the task completion rate for task 4 and 5 was 0%. Lastly, most of the participants performed "Task 5. Set GoTo-point and Point of Interest" with "Complete success" and one of the participants performed the task with "Success with a major issue". This implies a task completion rate of 66%.

An overview of the task completion rates can be seen in Figure 59.

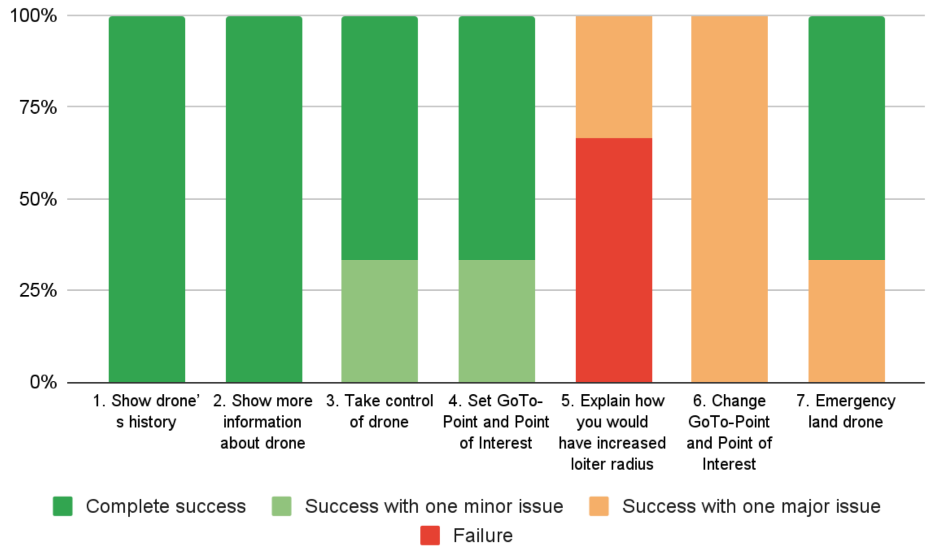

Figure 59. Task completion rates for the three participants that participated in the usability testing of the high-fidelity prototype

Participants usually succeeded with tasks that were related to the hamburger menu, i.e "Task 1. Show Drone's History", "Task 2. Show More Information About Drone", and "Task 7. Emergency Land Drone". When participants performed task 1 and 2 there was no confusion expressed and neither of the tasks were included in the respondents answers to the follow-up questions.

Regarding "Task 7. Emergency Land Drone", there was one participant that initially pressed on the drone and then started discussing whether the emergency actions could be controlled from a separate mouse.

Success rate on “Task 4. Set GoTo-Point and Point of Interest” and “Task 6. Change GoTo-Point and Point of Interest” were generally low among participants. Many of them accidentally performed these tasks when it was not their intention to do so. In addition, participants expressed that there were many things happening at the same time.

All participants were successful when performing “Task 3. Take Control of Drone”. There was one participant that expressed confusion about the outcome of the task. Additionally, one of the participants mentioned in the discussion that it could be a good idea to include even more information within the drone’s menu so that the operator could make a more well informed decision on what drone to take control of. However, one can state the placement, colors and labels of the “Take Control” button seems to be intuitive to the participants since all of them, without hesitation, clicked at this button when they were asked to take control of the drone.

Regarding “Task 5. Increase Loiter Radius”, the completion rates were low. Many of the participants expressed that they did not know what loiter radius was. One of the participants succeeded in increasing the loiter radius without knowing the meaning of the loiter radius.

Furthermore, participants understood that the dashed and solid line represented different states of routes. The general interpretation was that the solid line was the active route and that the dashed line represented the future. However, one of the participants said that the solid line is the planned route and the dashed one represents the drone’s actual route.

Lastly, there were several words and expressions that the participants verbalized that they were unsure of the meaning of it. The participants had particularly difficult to make the connection that Point of Interest was related to the drone’s camera. Some of the participants verbalized that they did not understand what MSL and SOG stands for. In addition, one of the participants pointed out that elevation should not be used as a value that can be changed but rather be read by the operator.

6.3 Discussion

When evaluating the hamburger menu, the task success rate was high, none of the participants expressed confusion or included it in their answers in the follow-up queries. Thus, one can make the conclusion that the placement of the hamburger menu and its content matches the user's mental model.

The results show that one of the participants failed to emergency land the drone. Being able to emergency land the drone is a task with a high failure cost and is associated with one of the most important requirements of the interface, to operate the drone safely. The participant who failed to emergency land the drone proposed that the emergency actions could be controlled from a separate mouse. For the next prototype, solutions on how to separate the emergency actions from the rest of the functions, either within the interface or to an external unit, could be investigated.

Completion rate on "Task 4. Set GoTo-Point and Point of Interest" and "Task 6. Change GoTo-Point and Point of Interest" were generally low among test participants. One explanation could be the limited level of interactivity within the prototype and that left and right click were not differentiated within the interface.

The option to change GoTo-point and Point of Interest is planned to only be available if the user has right clicked somewhere within the map. In this prototype, once a user had taken control over the drone, the option became available every time they clicked at any element that did not have any interactivity associated with it. Thus, one can state that this interactivity limitation caused the participants to accidentally view this option, which in turn, caused participants to set GoTo-point and Point of Interest unintentionally. Hence, the task completion rates would probably have been higher on these tasks if the interactivity did not have the stated limitations.

The users' unintended change of GoTo-point and Point of Interest may also be an explanation to why many of the participants expressed that it felt like a lot happened at the same time when setting a GoTo-point and Point of Interest. From these results, one can argue that the user might unintentionally set a GoTo-Point or Point of Interest if the option is available to them which in turn will cause great confusion. However, since the prototype did not function as planned regarding this feature, more testing of this functionality needs to be performed.

Task 5 involved increasing the loiter radius and the success rates among participants were low. The result may be explained by the fact that few of the participants knew what loiter radius was. Therefore, it could be reasonable to investigate whether this expression should be changed. In addition, one may also examine other approaches for how to increase the loiter radius. However, this function has a relatively low failure cost and hence other functions will be investigated prior to this one.

Lasty, the results demonstrated that participants had difficulties relating Point of Interest to the drone's camera. It was not explained in advance to the user's that the drone had a camera, however, several of them requested to see the drone's live video feed which shows that they assumed the drone had a camera. Thus, alternative terms for Point of Interest may be examined. As stated, some of the participants expressed that they did not understand abbreviations as SOG (Speed over ground) and MSL (Mean sea level). Therefore, one should consider using the complete word instead of the abbreviation in the next prototype. Moreover, elevation was used in the wrong way within the prototype and hence the correct way to use it will be determined and implemented.

7. Final Product Development

This section declares how the final product was developed and evaluated. Descriptions of how development tools, such as Visual Studio Code, GitHub, MapBox, React and React Map GL, have been used are included, see 1.6 Development Tools.

7.1 Method

The final product was developed in Visual Studio Code using development tools such as GitHub, React, and MapBox. Visual Studio Code was used as code editor and the interface was built as a react application using TypeScript. Moreover, GitHub was used for version control.

Coding was initiated with creating a React application in Visual Studio Code. Thereafter the map from MapBox was set up. This was done by implementing the map as a component with React Map GL. A style chosen by the stakeholder was applied to the map.

As previously described in 5.1 Method, an organization scheme was used for naming and grouping elements in Figma during the ideate and prototype phase. This organization scheme was used as a foundation for what components to implement in React. For example if the organization scheme contained a group named “Drone Menu”, a “Drone Menu” component was built in React with a similar structure and naming.

The interface consists of components that have been designed solely for this project and components that have been imported from React Map GL and Material UI. As identified in 4.2 Product Requirements, the interface should use standards, therefore components from Material UI that implements Google Standad’s library were used when possible rather than creating new components. When creating new components and it has been suitable, elements have been exported as SVGs from

Figma and incorporated into the components. Thus, some of the elements are identical to the ones used in the high-fidelity prototype.

Moreover, the interface was implemented based on the high-fidelity prototype, see 5.2.2 High-Fidelity Prototype and the conclusions from the usability testing, see 6. Test Prototype. However, additional solutions have been identified from Google Maps, such as placements of components. Design decisions for elements that have been kept from the high-fidelity prototype is discussed in greater detail in 5.2.2 High-Fidelity Prototype and hence will not be explained repeatedly in this chapter.

Lastly, the interface was connected with a simulated drone from which it was possible to both send and receive messages from. These messages could be used to send commands to the drone, such as setting a GoTo-point, and which the simulated drone acted upon.

7.2 Results

The final product is an interface that can be connected with a simulated or real drone. All the following figures are images from when the interface has been connected to a simulated drone. Thus, positions and a majority of values within the interface have been received from the simulated drone. Likewise, when doing certain actions within the interface these are sent as commands to the simulated drone which in turn takes action based on the command.

Initial view contains a map, drone, launch box, drone menu, navigation control, full screen control, preview of take off route, gimbal cone and fabricated alarm. The map allows the user to move within it, zoom in and out, and to enter full screen, in accordance with product requirement FEAT 23.

Figure 60. Initial view of interface in the final product

Navigation and full screen control, which both are components from React Map GL, have been included and placed in the lower right corner in alignment with placement of similar functions within Google Maps. This was done to keep external consistency in accordance with FEAT 54.

Figure 61. Placement of zoom and navigation functions within Google Maps

From *Google Maps*. [Screenshot], by Google. 2024. Last accessed 2024-05-06 from https://www.google.com/maps/@57.647925,11.8205988,13z?entry=ttu

Figure 62. Navigation and full screen control's placement within final product

Information and elements within the drone menu have been carefully selected to keep a minimalistic design with only relevant information in accordance with product requirement FEAT 58. The selected information is based on drone control requirements. This includes the drone's name (FEAT 7), hamburger menu (FEAT 36), battery (FEAT 42), drone's state (FEAT 21 & 51), launch box's state (FEAT 22 & 51), wind conditions (FEAT 8), and a “Start flight” button (FEAT 1).

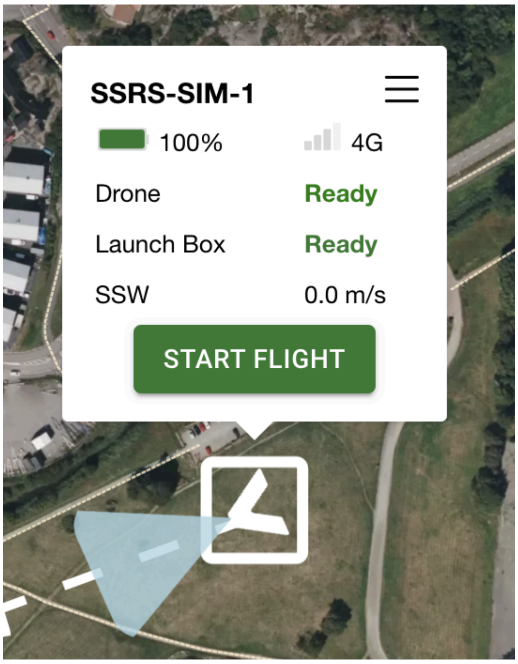

Figure 63. Drone menu in the final product

The drone menu consists of a “Popup” component from React Map GL. This component was chosen since it had built in characteristics that aligned with previously stated product requirements. Firstly, the “Popup” component is attached to the drone's latitude and longitude rather than a position on the screen. Hence, the drone menu's position does not change when the user moves within the map. This feature was desirable since it facilitates that new information within the menu is associated with the drone, according to product requirement FEAT 47. Moreover, the “Popup” component has a consistent size regardless of how much the user zooms in or out within the map which aligns with product requirement FEAT 48 which entails that elements within the interface should always be visible to the user.

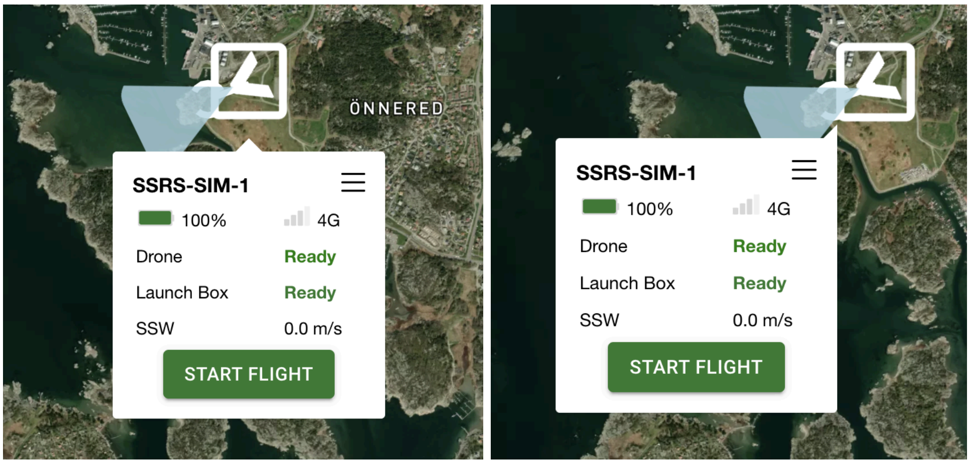

Figure 64 and 65. Drone menu is always attached to drone regardless of how the user moves around within the map in the final product

React Map GL provides another component, “Marker”, that also has the characteristic that elements wrapped within the component have a consistent size regardless of how much the user zooms in or out within the map. Thus, the drone and launch box was wrapped in a “Marker” component to fulfill product requirement FEAT 48. Additionally, the “Marker” component is attached to a longitude and latitude rather than a position on the screen. Hence, wrapping the drone within a “Marker”, whose position is updated depending on the simulated drone’s position, implements FEAT 39 which encompasses displaying the drone’s GPS position.

Figure 66 and 67. Size of drone menu, drone, launch box, gimbal cone and alarm remains consistent when zooming out in the final product

The “Start flight” button is a component from Material UI. It was modeled as a Call-To-Action button to indicate to the user that this is a key function. This was done by choosing one of Material UI’s colors that was highly contrasting towards the white background. The chosen color had a contrast ratio of 5.12:1 towards the background.

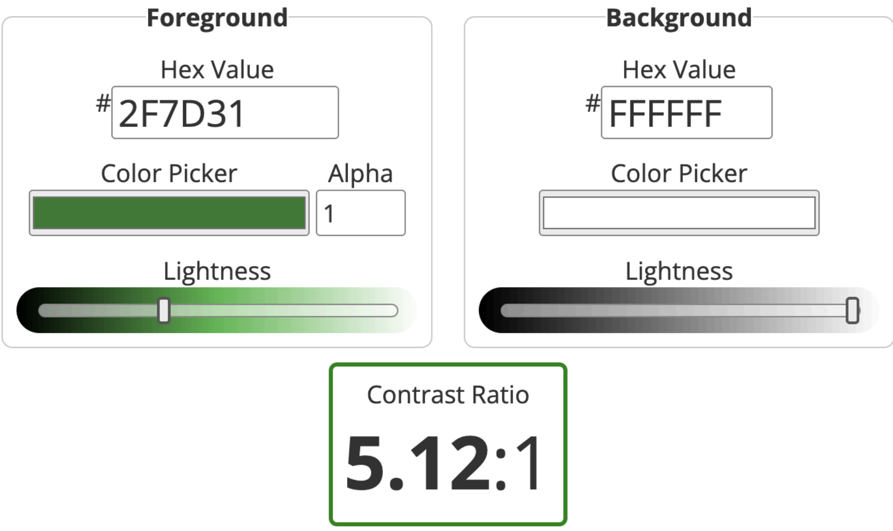

Figure 68. Contrast ratio 5.12:1 between Call-To-Action button "Start Flight" and background in the final product

As previously described in 2.3.3 Menus, all caps decrease legibility by 10%. However, as described in 2.1.1 Usability Heuristics, using standards is essential and since all caps were a standard setting for the Material UI button component, this was chosen to be kept. Furthermore, the button comes with a shadow-effect which was maintained since it aligned with established product requirement FEAT 67 about that the design should communicate how it is supposed to be used.

Moreover, the button has a built-in hover effect that conveys the button's state when being hovered over. This effect was chosen to be retained in accordance with theory regarding that the hovering effect is profitable to communicate a button's state, as presented in 2.3.2 Buttons. The "Start flight" button makes it possible for the operator to take control of a drone and start the flight with one mouse click, i.e allowing the operator to operate fast in accordance with FEAT 49.

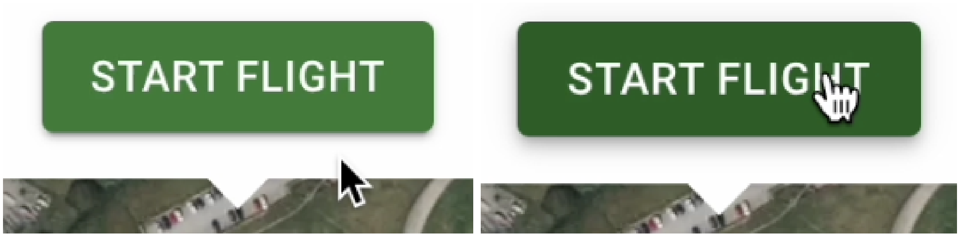

Figure 69 and 70. Hover effect for the "Start flight" button in the final product

For the same reason, the hovering effect was included in the hamburger menu which also is a component from Material UI. The hamburger menu itself contains two functions: "Show History" and "Show Info" in accordance with product requirements FEAT 36 and FEAT 2.

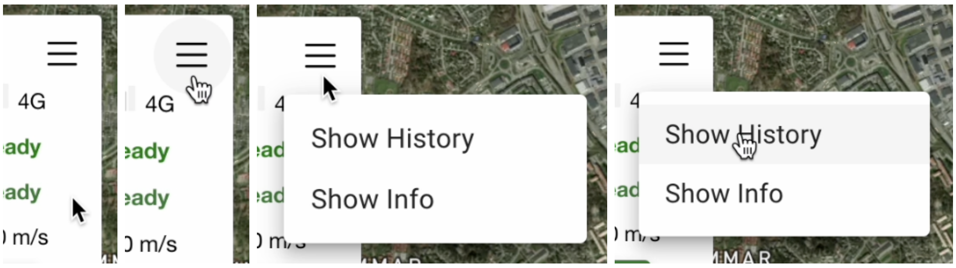

Figure 71-74. Hover effect for the hamburger menu in the final product

Furthermore, the drone's automatically decided take off route is displayed in alignment with product requirement FEAT 26. The route has a dashed style to convey that this route is not active yet, according to FEAT 18 and FEAT 51.

When pressing the "Start Flight" button the drone launches and starts flying along its predetermined take off route. To indicate that the drone is in control and flying, the drone turns yellow. The route becomes solid to indicate that it is now active, i.e the route that the drone is flying. In addition, a blue cone designed by the stakeholder is attached to the drone. This cone represents where the drone's camera is directed.

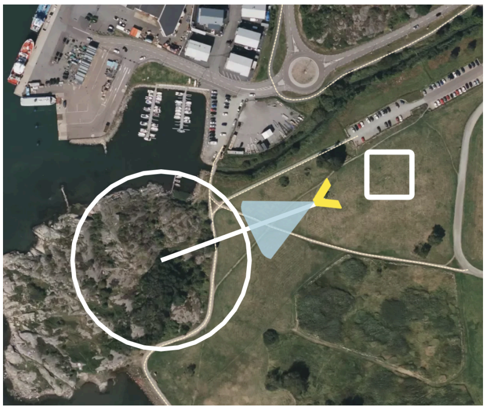

Figure 75. Take off route displayed before pressing "Start Flight" in the final product

Figure 76. Drone flies automatically decided take off route after pressing "Start Flight" in the final product

After pressing start, the prior menu attached to the drone is replaced with a new menu placed to the left. This menu contains some of the elements from the prior menu, such as the drone's name, battery, and coverage. Additionally, the hamburger menu is included as well. All the menu elements kept from the prior menu have been placed in the same locations to keep internal consistency in accordance with product requirement FEAT 54.

Figure 77. Drone loitering after launching the drone in the final product

This drone menu contains three different sections that are divided by two lines. The first section contains the elements kept from the prior menu and information about an eventual GoTo-point. When the user has not chosen a GoTo-point, the text “No active GoTo-point” is displayed. This text has been included to keep the user informed about the system’s state, in accordance with product requirement FEAT 51. When the user has chosen a GoTo-point, the coordinates, and the time and distance for the drone to fly to that point is shown (FEAT 27 & 32).

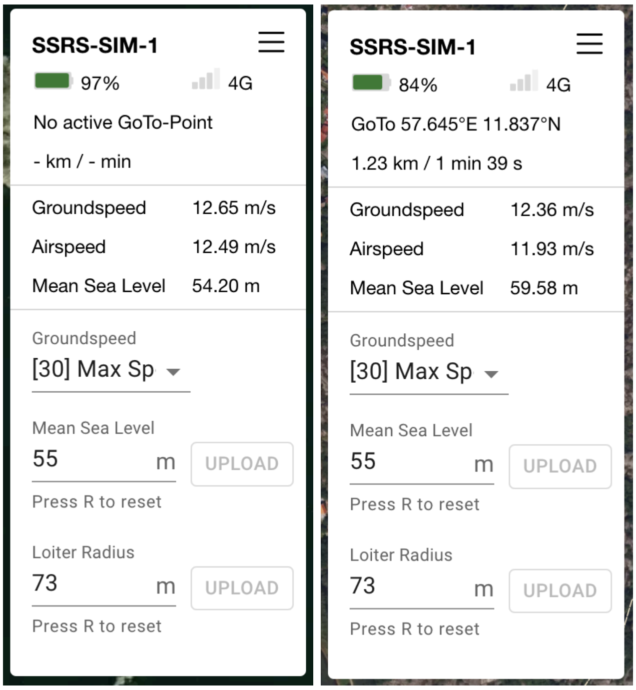

Figure 78 and 79. Drone menu with no active GoTo-point and active GoTo-point in the final product

The hamburger menu contains emergency functions in accordance with product requirement FEAT 17. Furthermore, the same hovering effect is used again to maintain consistency (FEAT 54).

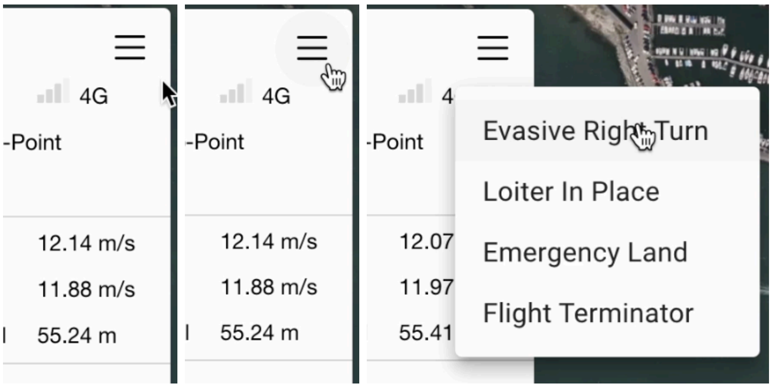

Figure 80-82. Hover effect of the hamburger menu in the final product

Second section contains the drone's groundspeed (FEAT 38), airspeed (FEAT 19), and mean sea level (FEAT 27). Third section includes input fields, a drop down menu, and buttons to update values of the drone. The drop down menu contains functionality to change the target speed (FEAT 38). Moreover, there are two input fields with an associated upload button each. One of the text fields can be used to change the drone's mean sea level (FEAT 37) and the other one can be used to change its loiter radius.

The input fields for changing mean sea level and loiter radius works in the same way to keep internal consistency (FEAT 54). In addition, the input fields constrain types of input (FEAT 61) so that it is not possible to write any letters. In accordance with FEAT 63, the text fields have reasonable defaults.

Moreover, users have been provided with the possibility of changing the values in different ways. Users can either choose to change the value with the stepper button or with the keyboard. When using the keyboard the user can choose to either highlight the value and overwrite the current value or erase the current value with backspace and write a new value. Offering the user different ways of doing certain tasks provides the user with flexible processes, as in alignment with FEAT 57.

The buttons next to the fields get enabled whenever the user has changed the value within the field. This is to keep the user informed about the system's state, i.e value has been changed and needs to be sent to the drone, and help users complete their intended tasks in accordance with FEAT 51 and 60.

To make the process of using the input fields more flexible and efficient in accordance with product requirement FEAT 57, a shortcut has been added. Instead of using the upload button, users can press the “Enter” key to send the value to the drone. If the user chooses not to use the upload button they have two remaining options, they can either reset the value by pressing “R” or execute the value by setting a new GoTo-Point. To minimize users’ memory load (FEAT 56) and help users complete their intended tasks by providing documentation (FEAT 60), a helper text has been added under the input field.

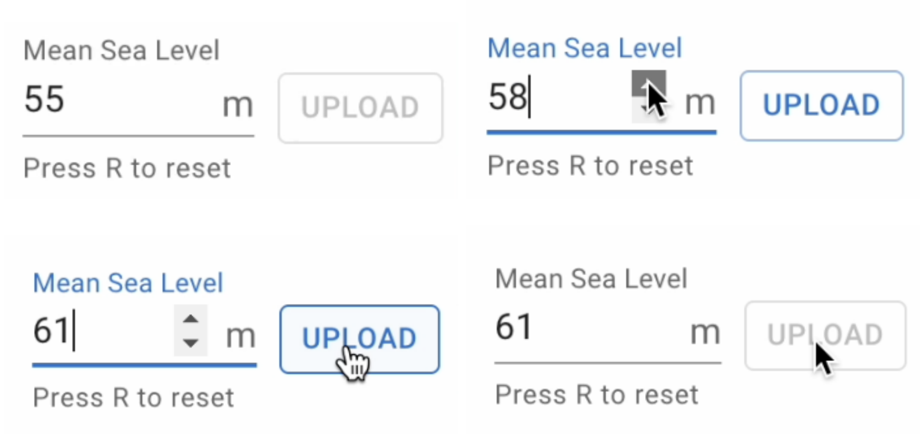

Figure 83-86. Update mean sea level by using stepper and upload button in the final product

FEAT 28, 29, 43, and 44 states that the operator should be provided with the possibility of changing GoTo-Point and Point of Interest. As described in 5.2.2 High-Fidelity Prototype, a solution where the user right clicks to display a context menu was identified. A similar context menu was implemented in the interface to keep external consistency (FEAT 54) and match the user's mental model (FEAT 65).

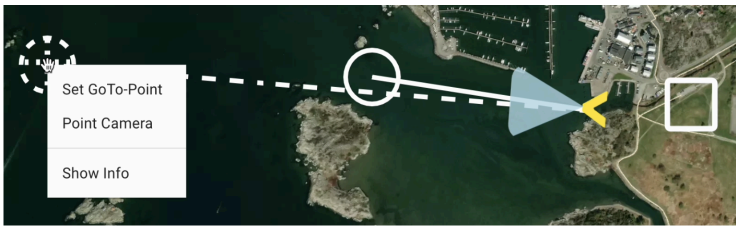

Figure 87. Context menu in Google Maps

From *Google Maps*. [Screenshot], by Google. 2024. Last accessed 2024-05-14 from https://www.google.com/maps/@57.6703113,11.8438605,3570m/data=!3m1!1e3?entry=ttu

Figure 88. Context menu displayed when the user right clicks in the final product

The context menu contains three different actions, “Set GoTo-Point”, “Point Camera”, and “Show Info”. FEAT 34 states that the interface should provide the possibility of displaying information about other locations than the drone’s location. Hence, the “Show Info” action was included within this menu as well. From the results in 6. Test High-Fidelity Prototype, it was demonstrated that few users made the connection between Point of Interest and the drone’s camera. The button that earlier was labeled “Set POI” was therefore labeled “Point Camera”.

Product requirement FEAT 68 defines that the interface shall preview results before the user takes action. Thus, a preview of the GoTo-Point and Point of Interest that would be streamlined, if the user chose any of those actions, is displayed to the user. To indicate which is preview or active, in conformity with FEAT 18, the preview has been modeled in dashed and active in solid.

7.3 Discussion

While tracking all features and elements within the final product to either theory or product requirements, it was noticed that the drone's coverage could not be tracked back to any requirement. Thus, this element should be removed in the future. One feature that was included, however not explicitly stated as a product requirement is the possibility to change the loiter radius. This need was identified in 4.2 Product Requirements, nevertheless have been unintentionally excluded when stating all product requirements.

Furthermore, some of the established drone control product requirements ranked "Must Have" in 4.2.3 Requirements Analysis have been excluded. More specifically, FEAT 28 and FEAT 29's functionality that the interface should provide the possibility of change and set a GoTo-Point before the drone has been launched has been neglected. These were not included since the simulated drone has a predetermined take off GoTo-Point which cannot be changed.

Moreover, FEAT 70 regarding making users confirm destructive actions was not applicable. This product requirement was not relevant due to the fact that the destructive actions, i.e the emergency functions, were empty buttons that did not lead to any action when clicking on them. Thus, it was not possible for the user to make any destructive actions within the interface and hence this product requirement was excluded. However, if the buttons were actually leading to an action some kind of confirmation dialog should be included.

Lastly, FEAT 71 states that the interface shall warn the users about errors before they are made. However, theory presented in 2.1.1 Usability Heuristics explains that one way to prevent errors is to reduce the number of situations where they can occur. In addition, theory discussed in 2.1.2 Error Prevention states that too many confirmation dialogs can cause users to stop reading them and then in turn increase the number of errors. Thus, the main focus has been on reducing these kinds of situations instead of inserting warnings or confirmation dialogs.

This has been done by limiting the user in what actions they can make. One example could be how to prevent the user from uploading an invalid mean sea level value, such as letters, to the drone. If FEAT 71 were followed strictly, the user had been warned when typing letters in the text field. However, this situation has instead been handled by not making it possible to write letters in the input field, and hence, this error cannot occur.

8. Test Final Product

In this chapter, the usability testing of the final product is described. Methods to perform the testing are presented in 3.5.1 Usability Testing. Analysis methods used are explained in 3.2.1 Thematic Analysis and 3.2.3 Critical Incident Analysis.

8.1 Method

Test of the final product was performed on six participants that could be real users of the product. Participants were either individuals with experience of flying drones or members of the Swedish Sea Rescue Society that had a professional background as airplane pilots, nonetheless with no experience of flying drones. None of the users had seen the interface before participating in the testing. However, the two individuals with experience of flying drones had participated in data gathering with potential end users (4.1.2 Data Gathering).

The testing method was 3.5.1 Usability Testing and chosen due to the same reasons presented in 6. Test High-Fidelity Prototype. Testing was performed remotely via Zoom where the user was granted control over the facilitator's computer from which they could control the simulated drone. In the beginning of the Zoom meeting, the participants were asked for permission to record the screen while testing.

Thereafter the following instructions were read aloud to the participants.

- You will be granted control over the facilitators computer from which you can control a simulated drone
- You will be asked to perform a number of tasks within the interface
- The goal is to achieve the tasks as fast and safe as possible
- Your amount of time to complete tasks will be measured
- To operate in the interface you are allowed to use your mouse, both left and right click, and keyboard
- Think aloud while performing the tasks, i.e share all your thoughts while operating in the interface

- Drone has a predetermined take off path
- After all tasks are completed, you will be asked some follow up questions
- Before the test starts you will have some time to familiarize with the interface and ask any questions you may have

After the instructions were read to the participants, the participants got a few minutes to familiarize themselves with the interface and ask questions. This was done due to the fact that when the interface is used in real situations, the users will have seen and used it beforehand. Thus, in order to make the testing situation as realistic as possible, the participant also had the chance to get familiarized with the interface.

Lastly, the background and tasks that the participant were supposed to perform were read aloud to the user, which were as follows.

Background

“You are a volunteer on-call at the Swedish Sea Rescue Society. Just now there is an incoming alarm from a private person saying that their propeller suddenly stopped working and that they need assistance. Your task is to fly the drone to the location of the alarm and monitor the situation with the drone's camera. You want to arrive at the location of the alarm as soon as possible but you cannot fly over land, except during the drone’s predetermined take off route.”

Tasks

- Fly the drone to the location of the alarm
- Point the drone’s camera at the location of the alarm

When the drone is at the location of the alarm, the drone should do the following.

- Loiter in a radius of 100 m
- Fly 30 m over the sea

After the tasks were completed, the participants were asked the following questions.

1. Explain how you would have emergency land the drone
2. How would you have changed the drone’s speed?
3. Did you accidentally give any undesired commands?
4. What changes could be made to make control of the drone safer?

5. What changes could be made to make control of the drone faster?
6. Is anything ambiguous?

When testing was finalized the video recordings were observed, i.e indirect observing described in 3.1.3 Observation. Thereafter the data was analyzed by using thematic and critical incident analysis in conjunction. All results that were associated with any of the research questions or important situations determined by critical incident analysis were defined as major themes. Less relevant situations or incidents were identified as minor themes.

Lastly, the average completion time for important actions was calculated. Moreover, the thesis writer's time for completing the tasks were measured. Even though the thesis writer did not have the correct characteristics to be a realistic user, their experience with the interface could arguably be considered as an expert's knowledge of the interface since they have designed it. This was done to get a baseline for how fast different participants were.

8.2 Result

Major themes were defined as how to start the flight, set and change GoTo-Point, set and change Point of Interest, minimize the risk of sending undesired commands, emergency land drone, and operate fast and safe. Minor themes were change speed of drone and change and upload mean sea level and loiter radius.

Table 3. contains the participants' completion times of important actions. As stated, it was possible to change and upload mean sea level and loiter radius in different ways. Hence, the partipant's chosen method for changing and uploading the value have been included.

When the participants performed the testing, it was noticed that another context menu was displayed when right clicking at the alarm. Participant 5 clicked at an option from this context menu which caused them to accidentally terminate the test by leaving the website. Thus, participant 5 had to perform the test two times. ”T1” contains the results from the first test and “T2” the second test.

Moreover, it was observed that when the participants flew close to the alarm, the drone got covered by the alarm popup so that it was no longer possible to see the drone.

During participants' 2 and 5 tests, which were the first two tests, some functionality errors were discovered. Participant 2 was the first participant and the following error was discovered and fixed afterwards.

- The mean sea level value had an offset of 5 m

Following errors were fixed after participant 5's test.

- User could not open hamburger menu when the map was in full screen
- Preview of loiter radius were occasionally displayed at the wrong location

To solve the full screen error, the full screen button was removed. This means that none of the other participants had the opportunity to enter full screen.

Table 3. Completion time and methods for tasks included in usability testing of final product

Participant	**Ref.**	**1**	**2**	**3**	**4**	**5**		**6**	Average completion time (excl. Ref.)
Drone experience	No	No	No	No	Yes	Yes		No	
Airplane pilot	No	Yes	Yes	Yes	No	No		Yes	
						T1	T2		
Push start flight button	1	2	1	3	12	11	5	17	7.3
Set/Change GoTo-point	1	1	2	2	2	2	1	1	1.6
Set/Change Point of	1	3	2	2	1	2	2	4	2.3

Interest									
Change mean sea level	< 1	1	4	10	3	2	8	6	4.9
Method for changing mean sea level	Highlight and write value with keyboard	Highlight and write value with keyboard	Stepper buttons	Stepper buttons	Highlight and write value with keyboard	Writevalue with keyboard	Stepper buttons	Write value with keyboard	
Upload mean sea level	< 1	-	1	3	1	1	1	1	1.3
Method for uploading mean sea level	Enter	Setting GoTo-point	Mouse click	Mouse click	Mouse click	Mouse click	Mouse click	Mouse click	
Updating mean sea level	< 1	-	4+1	10+3	3+1	-	8+1	6+1	6.8
Change loiter radius	< 1	3	4	12	4	-	7	16	7.7
Method for changing loiter radius	Highlight and write value with keyboard	Highlight and write value with keyboard	Stepper buttons	Stepper buttons	Highlight and write value with keyboard	-	Stepper buttons	Stepper buttons	

Upload loiter radius	1	-	1	2	2	-	1	2	1.6
Method for uploading loiter radius	Enter	Setting GoTo-Point	Mouse click	Mouse click	Mouse click	-	Mouse click	Mouse click	
Updating loiter radius	1	-	4+1	12+2	4+2	-	7+1	16+2	10.2

The first task was to fly the drone to the location of the alarm. To succeed with this task, the participants had to press the "Start flight" button to launch the drone and then right click at points in the map to set out GoTo-points until the drone reached the location of the alarm. Participants 4 and 6, tried to set out GoTo-points before they had launched the drone. Both of them eventually realized that they had to press the "Start flight" button before setting out GoTo-points. Nevertheless, this misconception resulted in these participants receiving the longest times to press the "Start Flight" button.

All of the users managed to set out GoTo-points in less than 2 seconds. The participant's time for setting out a GoTo-point in the table is the time from when they set out their first GoTo-point. It is measured from when they right clicked and until they chose the option "Set GoTo-point". However, it was observed that if one clicks on the sides of the button, from either of the options in the context menu, the button does not work.

Even though all participants succeeded in setting out GoTo-points, participant 5 placed several GoTo-points in test 1 and thought that the drone would fly to each of them in the order they had set out the points. While the drone was flying, the participant then realized that the drone only flew to the latest chosen GoTo-point. After realizing, the participant started setting out one GoTo-point at the time in a desirable way.

After the testing was completed participant 5 explained that they intuitively would have chosen another way for setting out GoTo-point. The participant stated that they would have included a toolbar within the interface from which they would have chosen “Set waypoint” from and then left-clicked at locations within the map to set a waypoint. In addition, right click would then have been used to view more information about that waypoint.

Participant 5 also stated that it happened a lot when clicking right click and that it is unclear whether an action has been executed or not. In addition, the participant suggested including a close button within the context menu.

The second task given to the participants was to point the drone’s camera to the location of the alarm. To do this, the user’s had to right click at the location of the alarm and then choose the option “Point Camera” from the context menu. All of the participants successfully used the “Point Camera” function to set the Point of Interest at the location of the alarm. Additionally, the average time for completing this action was 2.3 seconds where the slowest time was 4 seconds.

Moreover, the participants were supposed to loiter in a radius of 100 m at a height of 30 m over sea at the location of the alarm. All of the users successfully updated the drone’s loiter radius and mean sea levels. Participants choose different combinations of ways to change and upload these values, however none of them used the “Enter” key to upload their values. The following table shows the average completion time for different methods when changing mean sea level and loiter radius.

Table 4. Average completion time for different methods of changing mean sea level in usability testing of final product

Change Mean Sea Level				**Average Completion Time**
Stepper buttons	4	8	10	7.3
Highlight and write value with keyboard	1	3	-	2
Write value with keyboard	2	6	-	4

Table 5. Average completion time for different methods of changing loiter radius in usability testing of final product

Change Loiter Radius				Average Completion Time
Stepper buttons	4	12	7	16
Highlight and write value with keyboard	3	4	-	-
Write value with keyboard	-	-	-	-

The data in the tables shows that participants that used the stepper buttons to change a value had the longest average completion time and highlight and write value with keyboard had the shortest.

Participant 2 said that the stepper buttons had too high granularity. The same user said that they rather had buttons to change the speed than a drop down menu and that the velocity labels are unnecessary. In addition, the user said that it could be a good idea to have a copy button next to the coordinates of the GoTo-Points so that the user could easily copy them and send them in an email or text message.

When the participants were asked to emergency land the drone, the results differentiated according to whether the participant had a background as an airplane pilot or had drone experience. All airplane pilots started with explaining how they manually would have emergency landed the drone. Individuals with drone experience, participant 4 and 5, located the “Emergency Land” function within the hamburger menu. However, both of them verbalized that they did not understand what “Flight Terminator” meant.

Participant 2 verbalized that they did not expect the drone to start loiter when it reached the edge of the circle. They stated that they expected the drone to fly into the middle of the circle and then out to the edge of the circle before it started loitering. In addition, participant 1 expressed that they were not entirely sure how to make the drone loiter.

Two of the participants had difficulties with changing the drone's speed. Participant 6 tried to change it where one can read the ground speed value and then pressed at the hamburger menu. Moreover, Participant 3 initially pressed at the hamburger menu. After a while participant 3 found the drop down menu and said that they would have changed it with the menu. However, the user has difficulties with using the menu and says that it is unclear.

Lastly, the users were not asked in general what they thought about the interface. However, the following quotations were expressed by different users during the follow up questions.

- "It was very easy, I think"
- "It is very clear what one is expected to do […], it is not difficult to handle in any way"
- "Generally, I think this seems good and that […] it is a simple way of flying"
- "I think the interface is tidy and well disposed"

8.3 Discussion

Two of the participants tried to set out a GoTo-point before they had started pressing the "Start flight". They both eventually realized that they needed to press the "Start button" first, however these participants were the slowest ones to press "Start flight". Since the interface needs to assist the pilot to operate fast (FEAT 49), it would be reasonable to investigate a solution for this problem.

Furthermore, even if the participants tried to set out a GoTo-point before they had started the flight, the interface prevented them from doing this by including error prevention (FEAT 55). The context menu is only visible to users first after they have started the flight, in turn this prevents users from sending an undesired command, in this case "Set GoTo-Point", to the drone.

As stated, one of the participants used GoTo-Points as waypoints and said that they intuitively would have set them out in another way. However, as the rest of the participants successfully used the GoTo-Points as they were intended to be used, one can argue that this function matches most of the users mental model (FEAT 65). In addition, all participants were able to set a GoTo-point in 2 seconds or less which can be concluded as fast to use since it only differs 1 second from the reference.

Moreover, it was observed that one has to press in the middle of the button in order for it to register the click. Clicks that are not registered results in an interface that is slower and less safe to use which does not align with FEAT 4 or 49. Therefore, one should consider using another component than the current one from Material UI.

The emergency land results show that the interface does not match the majority of the participants' mental model. One suggestion could be to model the emergency function as floating action buttons instead which was visualized in Figure 40. This would make the functions accessible in one click instead of two, which would arguably make them quicker to use which would align with FEAT 49. However, all of the emergency functions could be considered as destructive actions and thus, some kind of confirmation dialog should be included before the action is sent to the drone to fulfill FEAT 69.

Moving the emergency function from the hamburger menu would also make it possible to have the same functions in the menu before and after control one has taken control of a drone which in turn would improve the internal consistency of the interface (FEAT 54). Overall, this change would arguably make the interface safer and quicker to use (FEAT 4 and 49).

The results show that different users choose various methods for changing and uploading values. Hence, flexible processes in accordance with FEAT 57. However, it was noticed that the average completion time for changing a value with stepper buttons was slower than using the keyboard to change it.

To conclude whether it was the participants that used the stepper buttons that was slow or the button itself, one can look at participant 5’s completion time. Participant 5 was the only user that used different methods for changing a value. During their first test, they used their keyboard to change mean sea level which took them 2 seconds. During the second test, it took them 8 seconds to change the same value with the stepper buttons. Thus, one can make the conclusion that it took the same participant 6 seconds longer to change the value with the stepper button than the keyboard.

Furthermore, it is a fundamental requirement that the interface should be fast to use (FEAT 49). Therefore, the interface should not offer users a method to complete tasks that is slower than other methods. Thus, one could consider two changes to make this functionality faster. The first one is to lower the granularity of the stepper button, which would require the user to press fewer times on the button. This change would arguably make the stepper button faster to use since it involves less clicks. Secondly, one could remove the stepper buttons completely and hence only allow participants to change the value with the keyboard.

Moreover, the fabricated alarm should be modeled in another way due to two reasons. Firstly, the current design makes it possible to display a context menu that does not belong to the interface when right clicking on it. This context menu caused one of the participants to leave the website. This does not align with product requirement FEAT 53 of putting the user in control and is therefore not desirable. Secondly, the drone got hidden behind the alarm so that it could no longer be seen. If the participant cannot see the drone, it is difficult to plan the flight safely (FEAT 4), determine if the flight is conducted safely (FEAT 12), and to perform it with minimal disturbance for people and animals (FEAT 5).

Furthermore, some of the users had difficulties with understanding how to change the drone's speed. This function is also important in order for the pilot to be able to plan the flight safely, perform it with minimal disturbance, and operate fast. To match the user's mental model better and at the same time reduce the amount of clicks that the user has to perform, one could have three buttons labeled with the associated speed instead of a drop down menu.

Lastly, the participants' quotations about the interface suggests that the interface is easy to use and aesthetically pleasing in accordance with FEAT 58.

9. Final Product Used in Real Flights

After the final product was developed, Remote Aero incorporated the interface in their software to use it when conducting flights. To be able to perform the flight some additional features, such as including the drone's live feed video, have been added by Remote Aero.

Thereafter, Remote Aero has been using the interface several times to fly the Swedish Sea Rescue Society's drones. The interface has successfully assisted the operator in setting and changing GoTo-Points, pointing the drone's camera, and operating safely. In addition, no undesired commands have been sent to the drone.

Figure 89. Swedish Sea Rescue Society's drone inside launch box before take off

Figure 90. Swedish Sea Rescue Society's drone launched and flying

10. Discussion

This section contains discussions about the process and methods used and the results. In addition, it entails some suggestions on further development of the interface.

10.1 Process and Methods

In general, the methods chosen during this degree project have been suitable to investigate the research questions. To begin with, it was helpful to conduct interviews and questionnaires with both stakeholders and potential end users to identify product requirements.

Moreover, the usability testing of the final product with potential end users has been valuable to determine if the interface matches the user's mental model. This requirement was important since it laid the basis for many other product requirements, such as if the interface allows the operator to plan the flight safely, fast and in a way that minimizes undesired commands.

Furthermore, it would have been insightful to include two different types of interfaces in the usability testing so that one could compare task completion times and thus determine what functions are the fastest ones. Nonetheless, the current approach makes it possible to remove or change functions that one concludes is too slow.

Live video feed could have been included into the interface to make the usability testing even more realistic. However, including it could also have created confusion to the user since it would not have been possible to adapt the live video according to the actions they take within the interface.

Finally, different ways to set out GoTo-Point and Point of Interest could have been included in the usability testing so that one could investigate which approach is the most suitable. However, different solutions and approaches were considered and prototyped during the ideate phase. These prototypes were evaluated together with stakeholders and supervisors and the outcomes from the elevations substantiated in choosing the current solution which has been proved to work efficiently when using it in real flights.

10.2 Results

During the usability testing of the final product, the participants were told that they were in an emergency situation and were asked to perform the flight as safely and fast as possible. Nonetheless, operators arguably experience the situation differently during a real emergency response and when controlling a real drone rather than a simulated one. However, the interface has been used to control the Swedish Sea Rescue's drones and proved to work efficiently in a non-emergency situation. To further increase the validity of the results, it would be relevant to observe operators when controlling the drone in one of the Swedish Sea Rescue Society's practices and then evaluate it.

As discussed previously, Figma's limited interactivity made it difficult to perform usability testing. The results were still valuable to understand if the correct terminology was used and if the participants understood menu elements. However, it was challenging to make any conclusion regarding how to set a GoTo-Point or Point of Interest since the solution could not be correctly designed in Figma.

Almost all requirements that got the prioritization "Must Have" have been included in the final product and many of the "Should Have" and "Could Have" as well. Notwithstanding, it was identified that many requirements were not relevant for the identified scope of this degree project or was redundant to be able to control a drone in this context. In hindsight, it would also have been beneficial to reduce the amount of product requirements since several of them were not directly associated with the research questions. Nevertheless, MoSCoW analysis was an efficient tool to distinguish which requirements were in alignment with the purpose of the degree project and should be higher prioritized.

10.3 Further Development

One suggestion of further development would be to include functionality that allows the operator to add waypoints. This would give the user the possibility to plan several points that the drone should fly to instead of only one. Adding this functionality would also include investigating how one could remove or edit waypoints. However, the interface is designed in a way that this kind of feature conveniently could be incorporated. As an example, one could include “Add Waypoint” in the current context menu and then include information about it in the drone menu. An illustration of how waypoints could be included can be seen in Figure 41.

Moreover, it would be relevant to investigate the access to the emergency functions since few of the users in the usability testing of the final product could find them. As described earlier, a solution to this problem could be to model them as floating action buttons within the interface, see Figure 35. In addition, The buttons should clearly communicate that they are emergency functions. Finally, some type of confirmation dialog should be included since these can be considered as destructive actions.

To make the control of the drone safer and faster, there are several things one should consider. Firstly, the method for deciding the drone's speed should be changed so that it can be done in only one click. One example could be to have three buttons labeled with three different speeds. Secondly, the drone's live video feed should be included. Lastly, the fabricated alarm and context menu should be modeled in another way than its current components since both of these disturbs its intended functionality, as described in 8.3 Discussion.

Finally, as stated the drone's coverage should be removed since it is not explicitly stated in the product requirements that this should be included.

11. Conclusion

From this degree project one can conclude that it is fundamental to match the user's mental model and apply error prevention to allow the operator to operate safely, fast and avoid undesired commands. In addition, this is important in order for them to succeed in deciding where the drone should go and control the drone's camera.

One can make the conclusion that an essential part of matching the user's mental model is by writing in their terminology. This was demonstrated both in the usability testing of the high-fidelity prototype and final product. For example, few of the participants in the usability testing of the high-fidelity prototype realized that Point of Interest had anything to do with the camera. The label of this button was later changed from "Set Point of Interest" to "Point Camera" in the final product. In the final usability testing, none of the participants had difficulties with using this function or making the connections that this button was related to the drone's camera.

It is also possible to conclude that it is fundamental to do proper user research and know what background the end user will have. This conclusion can be made based on the results that all participants that had a profession as a pilot were initially unable to locate the emergency functions whereas individuals with drone experience were able to locate them. Having a good understanding of the potential end user will also facilitate matching their mental model.

Moreover, it was efficient to use error prevention to avoid sending undesired commands to the drone. In particular, it was profitable to reduce the number of situations where errors can occur by limiting the number of options for the user. In the usability testing of the high-fidelity prototype, once the user has taken control of a drone, the context menu was available to the user each time they pressed at an element that did not have any interactivity associated with it. This caused the users to set GoTo-Point and Point of Interest unintentionally.

This was also observed in the testing of the final product when users tried to view the context menu before they had taken control over a drone. If the context menu would have been displayed, one can make the assumption that they would have sent an undesired command to the drone by choosing “Set GoTo-Point”. In addition, when one of the participants accidentally displayed a context menu that did not belong to the interface, they unintentionally left the website. Thus, one way to efficiently reduce undesired commands is to reduce the number of situations where this can happen by only giving the user options that are relevant for that particular context.

Furthermore, one can conclude that maintaining external consistency and standards seems to be an efficient approach when designing control software based on the results that almost all users managed to set GoTo-Points and use the input fields and their buttons efficiently. The solution for setting GoTo-Points was identified from external search and discussions with stakeholders. Input fields and buttons were components imported from Material UI.

Regarding the input fields, flexible processes to update values were offered to the users. However, it was observed that the stepper buttons were slower to use than using the keyboard. Thus, when including several methods to do the same action it is important to observe if these slow users down if it is a stated requirement that the interface should be fast to use.

Moreover, design principles of only containing relevant information, having a minimalistic design, reducing information over various sources, and associating new information to the data that is currently being handled was all implemented in only having one menu at the time. One can make the conclusion that these principles seem to be efficient based on the general user feedback in the testing of the final product.

Lastly, one can conclude what information is absolutely necessary for safe drone control beyond visual line of sight and these are as follows.

- Drone’s GPS-position
- Live video feed
- Functionality to decide where the drone should fly
- Functionality to point the drone’s camera
- Possibility to set target altitude
- Possibility to set target speed

The following functionality is desirable to have, however not absolutely necessary.

- Time remaining of battery
- Visual representation of where the drone is flying
- Current speed
- Current altitude

References

1. Bittner, K & Spence, I. (2002). *Use Case Modeling* (1st ed.). Addison-Wesley Professional.
2. Budiu, R & Nielsen, J. (2021). *Success Rate: The Simplest Usability Metric*. Nielsen Norman Group. Last accessed 2024-04-10 from https://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/
3. Clifford Opoku, I. (2024). *How to Use Git and GitHub – a Guide for Beginners and Experienced Developers*. freeCodeCamp. Last accessed 2024-04-11 from https://www.freecodecamp.org/news/guide-to-git-github-for-beginners-and-experienced-devs/#what-is-github
4. Figma (n.d). *What is Figma?*. Last accessed 2024-04-11 from https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma
5. Friis Dam, R. (2024). *The 5 Stages in the Design Thinking Process*. Interaction Design Foundation. Last accessed 2024-04-08 from https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process
6. Gibbons S. (2021). *5 Prioritization Methods in UX Roadmapping*. Nielsen Norman Group. Last accessed 2024-04-03 from https://www.nngroup.com/articles/prioritization-methods/
7. Google Maps. (2024). [Screenshot]. Google. Last accessed 2024-04-12 from https://www.google.com/maps/@57.647925,11.8205988,13z?entry=ttu
8. Hancock, P.A., & Szalma, J.L. (2003). Operator Stress and Display Design. *Ergonomics in Design The Quarterly of Human Factors Application, 11(2),* 13-18. https://doi.org/10.1177/106480460301100205
9. Laubheimer P. (2015). *Preventing User Errors: Avoiding Conscious Mistakes*. Nielsen Norman Group. Last accessed 2024-04-03 from https://www.nngroup.com/articles/user-mistakes/
10. Liska, D. (n.d) *Benefits of Fixed Wing Drones over Multirotors*. [Photography]. Women And Drones. Last accessed 2024-04-05 from https://womenanddrones.com/benefits-of-fixed-wing-drones-over-multirotors/

11. Mapbox. (n.d). *Getting started.* Last accessed 2024-04-11 from https://docs.mapbox.com/help/getting-started/
12. MUI. (n.d). *Material UI.* Last accessed 2024-04-29 from https://mui.com/material-ui/
13. Meta Open Source (2024). *React.* Last accessed 2024-04-12 from https://react.dev/
14. Microsoft. (n.d). *Visual Studio.* Last accessed 2024-04-11 from https://visualstudio.microsoft.com/#vscode-section
15. Moran K. (2019). *Usability Testing 101.* Norman Nielsen Group. Last accessed 2024-04-05 from https://www.nngroup.com/articles/usability-testing-101/
16. Laubheimer, P. (2015). *Preventing User Errors: Avoiding Unconscious Slips.* Nielsen Norman Group. Last accessed 2024-04-03 from https://www.nngroup.com/articles/slips/
17. Nielsen J. (2024). *10 Usability Heuristics for User Interface Design.* Nielsen Norman Group. Last accessed 2024-04-03 from https://www.nngroup.com/articles/ten-usability-heuristics/
18. Nielsen J. (2008). *Right-Justified Navigation Menus Impede Scannability.* Nielsen Norman Group. Last accessed 2024-04-08 from https://www.nngroup.com/articles/right-justified-navigation-menus/
19. Nielsen J. (2012). *Thinking Aloud: The #1 Usability Tool.* Nielsen Norman Group. Last accessed 2024-04-05 from https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/
20. Nielsen J. (2012). *Usability 101: Introduction to Usability.* Nielsen Norman Group. Last accessed 2024-04-04 from https://www.nngroup.com/articles/usability-101-introduction-to-usability/
21. Pernice, K. (2014). *Talking with Participants During a Usability Test.* Nielsen Norman Group. Last accessed 2024-04-05 from https://www.nngroup.com/articles/talking-to-users/
22. Preece, J., Rogers, Y. & Sharp, H. (2011). *Interaction Design Beyond Human-Computer Interaction* (3rd ed.). John Wiley & Sons Ltd.
23. Pressman, R & Maxim, B. (2020). *Software Engineering: A Practitioner's Approach* (9th ed). McGraw Hill.
24. Remote Aero. (n.d). *Remote Aero.* Last accessed 2024-04-12 from https://remote.aero/
25. Rennie, J. (2016). *Drone Types: Multi-Rotor vs Fixed-Wing vs Single Rotor vs Hybrid VTOL.* AUAV. Last accessed 2024-04-05 from https://www.auav.com.au/articles/drone-types/

26. Sanchez-Heres, L., Falkman, F., Forsman, F., Hagner, O., Nilsson, V., Lundqvist, R., Bergman, K., Bjurling, O., Granlund, R., Hjalmarsson, M., & Brunnström, M. (2021). *Eyes on Scene - EOS.* RISE Research Institutes of Sweden.

https://fudinfo.trafikverket.se/fudinfoexternwebb/Publikationer/Publikationer_007401_007500/Publikation_007477/eos-slut-rapport.pdf

27. Sauro, J. (2011). *What Is A Good Task-Completion Rate?*. MeasuringU. Last accessed 2024-04-10 from https://measuringu.com/task-completion/.

28. Sherwin, K. (2015). *Low-Contrast Text Is Not the Answer.* Nielsen Norman Group. Last accessed 2024-04-08 from https://www.nngroup.com/articles/low-contrast/

29. SOARIZON by Thales. (2020). *What are VLOS, EVLOS and BVLOS? Why do they affect drone operations?*. Scaleflyt. Last accessed 2024-04-09 from https://www.scaleflyt.com/news/what-are-vlos-evlos-and-bvlos-why-do-they-affect-drone-operations

30. Suwala, K. (2019). *Build trust with buttons. 7 rules about designing buttons that will help you build satisfying User Experiences (UX) and User Interfaces (UI).* Medium. Last accessed 2024-04-08 from https://medium.com/itmagination/build-trust-with-buttons-629657f966ee

31. Swedish Civil Contingencies Agency. (2020). *Obemannade luftfartyg i kommunal räddningstjänst: vägledning 2.0.* https://rib.msb.se/filer/pdf/29465.pdf

32. Swedish Sea Rescue Society. (n.d). *Alla räddningsstationer.*

Last accessed 2024-04-25 from https://www.sjoraddning.se/raddningsstationer.

33. Swedish Sea Rescue Society. (n.d) *Vår räddningsflotta.* Last accessed 2024-04-25 from https://www.sjoraddning.se/raddningsflotta.
34. Swedish Sea Rescue Society. (n.d). *We save lives at sea with no government funding.*

Last accessed 2024-04-25 from https://www.sjoraddning.se/information-english.

35. Swedish Transport Administration. (n.d). *EOS2 - Eyes on Scene 2 - Säkrare och effektivare Sjöräddning med drönar-stöd.* Swedish Transport Administration. Last accessed 2024-04-12 from https://fudinfo.trafikverket.se/fudinfoexternwebb/pages/ProjektVisaNy.aspx?ProjektId=5245

36. Vis.gl (n.d) *Introduction.* Last accessed 2024-04-29 from https://visgl.github.io/react-map-gl/docs.

37. Terrestrial Imaging. (n.d) *Mavic 2 Enterprise Advanced.* [Photography]. Last accessed 2024-04-05 from https://www.terrestrialimaging.com/drone-equipment/multi-rotor-drones

38. The Swedish Transport Agency. (2023). *Kategori certifierad – en av tre kategorier.* Last accessed 2024-03-27 from https://www.transportstyrelsen.se/sv/luftfart/luftfartyg-och-luftvardighet/dronare/tillstand-for-dronare/kategori-certifierad/

39. The Swedish Transport Agency. (2024). *Kategori specifik – en av tre kategorier.* Last accessed 2024-03-27 from https://www.transportstyrelsen.se/sv/luftfart/luftfartyg-och-luftvardighet/dronare/tillstand-for-dronare/kategori-specifik/

40. The Swedish Transport Agency. (2023). *Utbildning för fjärrpiloter.* Last accessed 2024-04-11 from https://transportstyrelsen.se/dronarkort-och-utbildning/

41. Ulrich, K.T & Eppinger, S.D. (2015). Product Design and Development (6th ed.). McGraw Hill.

42. Visual Studio Code. (2024). *Visual Studio Code FAQ.* Last accessed 2024-04-11 from https://code.visualstudio.com/docs/supporting/FAQ

43. W3C Web Accessibility Initiative. (2023). *Understanding SC 1.4.3: Contrast (Minimum) (Level AA).* Last accessed 2024-04-10 from *https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html*

Appendix A. Requirements

Drone Control

Before Flight

Before Control

- Interface shall provide the possibility of taking control of drone
- Interface shall provide the option to view information about the drone
- Interface shall include a representation of a fixed wing drone

After Control

- Interface shall assist the operator to plan the flight safely
- Interface shall help the operator plan the flight in a way so that it can conducted with minimal disturbance for people and animals
- Interface shall help the operator fulfill the 1:1 rule

Before & After Control

- Interface shall contain the drone’s name
- Interface shall provide weather conditions of drone’s location

During Flight

- Interface shall display breadcrumbs received from flight terminator
- Interface shall contain timestamp for breadcrumbs
- Interface shall display drone’s live feed video during flight
- Interface shall provide the operator with the possibility to continuously being able to determine if the flight is conducted safely
- Interface shall contain drone’s course

- Interface shall present how far the drone will fly in one minute for current speed
- Interface shall display drone's heading
- Interface shall display drone's past route
- Interface shall contain, but not the functionality of, the following options
    - resume last waypoint
    - loiter in place
    - evasive right turn (30°)
    - emergency land
    - flight terminator - stop motor
        - Interface shall contain number to flight terminator
        - Interface shall present that it is possible to text number
- Interface should be able to indicate which route is active
- Interface shall display air speed
- Interface shall contain "return to home"-function

Before/During Flight

- Interface shall display the drone's state
    - in launch box (before flight)
        - under control
        - not under control
    - on ground (before flight)
        - under control
        - not under control
    - flying (during flight)
- Interface should display the launch box's state
    - ready to launch (before flight)
    - not ready to launch (before flight)
    - launched (during flight)
- Interface shall display airspace map (before & during flight)

- Interface shall allow the operator to move around the map (before & during flight)
- Interface shall contain name of GoTo-point (before flight/after control & during flight)
- Interface shall be capable of setting a GoTo-point upwind automatically (before/after control & during flight)
- Interface shall provide information about GoTo-point (before flight/after control & during flight)
    - distance to GoTo-point
    - time to GoTo-point
    - loiter radius
    - mean sea level
- Interface should provide a possibility for the operator to set a new GoTo-point manually
    - when the drone is not launched yet (before flight/after control)
    - when the drone is flying towards a GoTo-point (during flight)
- Interface shall provide a possibility for the operator to change the GoTo-point
    - when the drone is not launched yet (before flight/after control)
    - when the drone is flying towards a GoTo-point (during flight)
- Interface shall be able to automatically set point of interest to GoTo-point (before/after control)
- Interface shall be able to indicate GoTo-point status (before/after control & during flight)
    - upload pending
    - uploaded
- Interface shall always contain GoTo-point and drone information when they have been selected (before flight/after control & during flight)
- Interface shall be able to zoom in automatically (before & during flight)
- Interface shall provide the possibility of displaying information about locations other than the drone's location (before & during flight)
- Interface shall present if information within interface is active, varying or preset
- Interface shall provide the possibility of displaying drone's history
- Interface shall contain current and target altitude (before/after control & during flight)

- Interface shall contain current and target speed (before/after control & during flight)
- Interface shall contain drone's GPS position (before & during flight)
- Interface shall contain drone's climb rate (before/after control & during flight)
- Interface shall contain drone's descent speed (before/after control & during flight)
- Interface shall display battery status and time remaining (before & during flight)
- Interface shall allow operator to select point of interest of drone's camera (before/after control & during flight)
- Interface shall allow operator to change point of interest of drone's camera (before/after control & during flight)
- Interface shall contain name of point of interest (before flight/after control & during flight)
- Interface shall be able to indicate point of interest status (before/after control & during flight)
    - upload pending
    - uploaded
- Interface shall present current scope of drone's camera (before/after control & during flight)
- Interface shall keep elements, such as GoTo-points, routes or drones, in a size that is visible regardless of how much the operator zooms out
- Interface shall allow the operator to operate fast
- Interface shall be adapted to be used during emergencies

Usability

- Interface shall keep the user informed about the system's state
- Interface shall follow real-world conventions
- Interface shall provide user freedom and control
- Interface shall use standards and be consistent
- Interface shall contain error prevention
    - Interface shall constrain types of input

  - Interface shall provide contextual suggestions
  - Interface shall have reasonable defaults
  - Interface shall include lenient formatting
  - Interface shall match the user's mental model
  - Interface shall implement accepted design conventions
  - Interface shall communicate how it is intended to be used
  - Interface shall preview results before user take action
- Interface shall minimize the actor's memory load
- Interface shall provide flexible and efficient processes, like shortcuts
- Interface shall have a minimalistic design and only contain relevant information
- Interface shall contain error messages that help users "recognize, diagnose and recover"

Display Design Emergency Situations

- Interface shall reduce distribution of information over various sources
- Interface shall associate new information to the data that is presently being handled
- Interface shall present an overview on a single display by using integrated information formats
- Interface shall present changes over time of visual elements
- Interface shall use detachable configural windows when integrated windows are not pragmatic
- Interface shall have a logical semantic mapping when using configural displays
- Interface shall use principles of the ecological interface design to simplify fulfillment of the above guidelines
- Interface shall avoid displays requiring data transformation when the user under most stress
- Interface shall display simple graphics to deliver direct instructions on a low level when the user is under most stress

- Interface shall minimize “direct structural interference” when the user is experiencing a lot of stress

Appendix B. Requirements Analysis

Drone Control

Before Flight

Before Control

Table 6. MoSCoW analysis of requirements regarding drone control before flight and before control

Identifier	Description	Priority
FEAT 1	Interface shall provide the possibility of taking control of drone	Must have
FEAT 2	Interface shall provide the option to view information about the drone	Could have
FEAT 3	Interface shall include a representation of a fixed wing drone	Must have

After Control

Table 7. MoSCoW analysis of requirements regarding drone control before flight and after control

Identifier	Description	Priority
FEAT 4	Interface shall assist the operator to plan the flight safely	Should have
FEAT 5	Interface shall help the operator plan the flight in a way so that it can conducted	Should have

	with minimal disturbance for people and animals	
FEAT 6	Interface shall help the operator fulfill the 1:1 rule	Should have

Before & After Control

Table 8. MoSCoW analysis of requirements regarding drone control before flight and before and after control

Identifier	Description	Priority
FEAT 7	Interface shall contain the drone's name	Could have
FEAT 8	Interface shall provide weather conditions of drone's location	Should have

During Flight

Table 9. MoSCoW analysis of requirements regarding drone control during flight

Identifier	Description	Priority
FEAT 9	Interface shall display breadcrumbs received from flight terminator	Could have
FEAT 10	Interface shall contain timestamp for breadcrumbs	Could have
FEAT 11	Interface shall display drone's live feed video during flight	Will not have
FEAT 12	Interface shall provide the operator with the possibility to continuously being able to determine if the flight is conducted safely	Should have
FEAT 13	Interface shall contain drone's course	Could have

FEAT 14	Interface shall present how far the drone will fly in one minute for current speed	Could have
FEAT 15	Interface shall display drone’s heading	Should have
FEAT 16	Interface shall display drone’s past route	Should have

FEAT 17	Interface shall contain, but not the functionality of, the following options
	● resume last waypoint
	● loiter in place
	● evasive right turn (30°)
	● emergency land
	● flight terminator - stop motor
	    ○ interface shall contain number to flight terminator
	    ○ interface shall present that it is possible to text number
		Could have

FEAT 18	Interface should be able to indicate which route is active	Must have
FEAT 19	Interface shall display air speed	Could have
FEAT 20	Interface shall contain “return to home”-function	Could have

Before/During flight

Table 10. MoSCoW analysis of requirements regarding drone control before/during flight

Identifier	Description	Priority
FEAT 21	Interface shall display the drone's state ● in launch box (before flight) ○ under control ○ not under control ● on ground (before flight) ○ under control ○ not under control ● flying (during flight)	Must have
FEAT 22	Interface should display the launchbox's state ● ready to launch (before flight) ● not ready to launch (before flight) ● launched (during flight)	Must have
FEAT 23	Interface shall display airspace map (before & during flight)	Must have
FEAT 24	Interface shall allow the operator to move around the map (before & during flight)	Must have
FEAT 25	Interface shall contain name of GoTo-point (before flight/after control & during flight)	Could have
FEAT 26	Interface shall be capable of setting a GoTo-point upwind automatically (before/after control & during flight)	Could have
FEAT 27	Interface shall provide information about GoTo-point	Should have

	(before flight/after control & during flight) ● distance to GoTo-point ● time to GoTo-point ● loiter radius ● mean sea level	
FEAT 28	Interface should provide a possibility for the operator to set a new GoTo-point manually ● when the drone is not launched yet (before flight/after control) ● when the drone is flying towards a GoTo-point (during flight)	Must have
FEAT 29	Interface shall provide a possibility for the operator to change the GoTo-point ● when the drone is not launched yet (before flight/after control) ● when the drone is flying towards a GoTo-point (during flight)	Must have
FEAT 30	Interface shall be able to automatically set point of interest to GoTo-point (before/after control)	Could have
FEAT 31	Interface shall be able to indicate GoTo-point status (before/after control & during flight) ● upload pending ● uploaded	Could have
FEAT 32	Interface shall always contain GoTo-point and drone information when they have been selected (before flight/after control & during flight)	Could have

FEAT 33	Interface shall be able to zoom in automatically (before & during flight)	Could have
FEAT 34	Interface shall provide the possibility of displaying information about locations other than the drone's location (before & during flight)	Could have
FEAT 35	Interface shall present if information within interface is active, varying or preset	Could have
FEAT 36	Interface shall provide the possibility of displaying drone's history	Could have
FEAT 37	Interface shall contain current and target altitude (before/after control & during flight)	Could have
FEAT 38	Interface shall contain current and target speed (before/after control & during flight)	Could have
FEAT 39	Interface shall contain drone's GPS position (before & during flight)	Must have
FEAT 40	Interface shall contain drone's climb rate (before/after control & during flight)	Will not have
FEAT 41	Interface shall contain drone's descent speed (before/after control & during flight)	Will not have
FEAT 42	Interface shall display battery status and time remaining (before & during flight)	Should have
FEAT 43	Interface shall allow operator to select point of interest of drone's camera (before/after control & during flight)	Should have
FEAT 44	Interface shall allow operator to change point of interest of drone's camera (before/after control & during flight)	Should have

FEAT 45	Interface shall contain name of point of interest (before flight/after control & during flight)	Could have
FEAT 46	Interface shall be able to indicate point of interest status (before/after control & during flight) ● upload pending ● uploaded	Could have
FEAT 47	Interface shall present current scope of drone's camera (before/after control & during flight)	Could have
FEAT 48	Interface shall keep elements, such as GoTo-points, routes or drones, in a size that is visible regardless of how much the operator zooms out	Should have
FEAT 49	Interface shall allow the operator to operate fast	Must have
FEAT 50	Interface shall be adapted to be used during emergencies	Must have

Usability

Table 11. MoSCoW analysis of requirements regarding usability

Identifier	Description	Priority
FEAT 51	Interface shall keep the user informed about the system's state	Should have
FEAT 52	Interface shall follow real-world conventions	Could have
FEAT 53	Interface shall provide user freedom and control	Should have
FEAT 54	Interface shall use standards and be consistent	Could have

FEAT 55	Interface shall contain error prevention	Must have
FEAT 56	Interface shall minimize the actor's memory load	Could have
FEAT 57	Interface shall provide flexible and efficient processes, like shortcuts	Could have
FEAT 58	Interface shall have a minimalistic design and only contain relevant information	Should have
FEAT 59	Interface shall contain error messages that help users "recognize, diagnose and recover"	Should have
FEAT 60	Interface shall contain documentation that helps users complete their intended tasks	Should have

Error Prevention

Slips

Table 12. MoSCoW analysis of requirements for error prevention of slips

Identifier	Description	Priority
FEAT 61	Interface shall constrain types of input	Could have
FEAT 62	Interface shall provide contextual suggestions	Could have
FEAT 63	Interface shall have reasonable defaults	Could have
FEAT 64	Interface shall include lenient formatting	Could have

Mistakes

Table 13. MoSCoW analysis of requirements for error prevention of mistakes

Identifier	Description	Priority
FEAT 65	Interface shall match the user's mental model	Must have
FEAT 66	Interface shall implement accepted design conventions	Should have
FEAT 67	Interface shall communicate how it is intended to be used	Should have
FEAT 68	Interface shall preview results before user take action	Must have

Slips and Mistakes

Table 14. MoSCoW analysis of requirements for error prevention of mistakes

Identifier	Description	Priority
FEAT 69	Interface shall make the user confirm a destructive action	Must have
FEAT 70	Interface shall support undo	Could have
FEAT 71	Interface shall warn the user about errors before they are made	Must have

Emergency Situations

Table 15. MoSCoW analysis of requirements for systems is to be used in an emergency situation

Identifier	Description	Priority
FEAT 72	Interface shall reduce distribution of information over various sources	Could have
FEAT 73	Interface shall associate new information to the data that is presently being handled	Should have
FEAT 74	Interface shall present an overview on a single display	Could have

	by using integrated information formats	
FEAT 75	Interface shall present changes over time of visual elements	Could have
FEAT 76	Interface shall use detachable configural windows when integrated windows are not pragmatic	Will not have
FEAT 77	Interface shall have a logical semantic mapping when using configural displays	Will not have
FEAT 78	Interface shall use principles of the ecological interface design to simplify fulfillment of the above guidelines	Could have
FEAT 79	Interface shall avoid displays requiring data transformation when the user under most stress	Should have
FEAT 80	Interface shall display simple graphics to deliver direct instructions on a low level when the user is under most stress	Should have
FEAT 82	Interface shall minimize “direct structural interference” when the user is experiencing a lot of stress	Should have

Appendix C. Usability Testing of Prototype

Table 16. Test results from usability testing of prototype, including tasks and follow-up questions

Task	**Test Participant A**	**Test Participant B**	**Test Participant C**
1. Show drone's history	Pressed the hamburger menu in the drone's menu and then pressed the "Show Drone's History" button.	Pressed the hamburger menu in the drone's menu and then pressed the "Show Drone's History" button.	Pressed the hamburger menu in the drone's menu and then pressed the "Show Drone's History" button.
2. Show more information about drone	Pressed the hamburger menu in the drone's menu and then pressed the "Show Info" button. Participant said that it was a small box.	Pressed the hamburger menu in the drone's menu and then pressed the "Show Info" button.	Pressed the hamburger menu in the drone's menu and then pressed the "Show Info" button.
3. Take control of drone	Pressed the "Take Control" button and expressed that they were confused about the route that was automatically suggested.	Pressed the "Take Control" button.	Pressed the "Take Control" button.
4. Set GoTo-Point and Point of Interest	Pressed the "Set GoTo and POI"-button.	Pressed the "Set GoTo and POI"-button.	Pressed the "Set GoTo" and said that preferably they would like to press "Set GoTo" first and then "Set POI".
5. Explain how you would have increased loiter radius	Pressed the "+"-button next to the loiter radius and expressed that they did not understand	Said that they did not know what loiter radius was. The user initially pressed at the hamburger menu and	Said that preferably the user would have pressed "+" to the right of Asperö.

	what the loiter radius was.	then pressed at the POI icon on the map.	
6. Change GoTo-point and Point of interest	Participant set a new GoTo and POI without being asked to do so and were thereafter confused about what was happening. Participant say that they were not prepared that the drone would start flying directly.	Participant accidentally moved to this part of the test and was asking about only setting GoTo or POI. When the user was explicitly asked to change GoTo-point and POI, then the user pressed the "Set GoTo and POI" button.	Participant said that they rather would have liked to make sure that they have gotten the point that they wanted before it starts flying and then have a button that streamlines. Thereafter, the participant pressed in the drone menu. User then says that the user would have liked to start with "Set GoTo " in order to start with navigation. Furthermore, the participant states that they would like to make sure that the drone goes over open water, has a partial breakpoint, and start with navigation and then set the final destination. Additionally, the user stated that it happens a lot when pressing a button and that it goes too fast. Lastly, the user suggests that the planning section and "action" section should be separated physically.
7. Emergency land drone	Pressed the hamburger menu in the drone's menu and then pressed the "Emergency Land"-button.	Pressed the hamburger menu in the drone's menu and then pressed the "Emergency Land"-button.	Pressed on the drone first. Thereafter, the participant pressed on the hamburger menu and then the "Emergency Land"-button. User suggests that emergency actions could be performed from another mouse so that it would be

			more clearly muscular. Test participant suggest that perhaps the emergency actions could be placed straight ahead below the user or in the middle of the screen.
Follow-Up Questions	**Test Participant A**	**Test Participant B**	**Test Participant C**
What does the gray line symbolize [i.e the route that the drone has already flown]?	Participant said that it probably meant that it symbolized the route that the drone had been flying.	Participant said that it is the route that the drone has already flown.	Participant said that it represents the starting point and where the drone has already flown.
What is the difference between the dashed and solid line?	Participant said that this was a bit confusing and that normally the solid route is the route we try to follow and the dashed one is where it actually goes.	User is sure that the solid line is where the drone is heading, however, is unsure and guessing the meaning of the dashed line. Participant eventually makes the conclusion that the dashed line symbolizes where it would go, and calls it POI.	Participant said that the solid line is the active one and dashed represents upcoming tasks. User said that when executed, it will switch to the dashed one.
What does this line symbolize [i.e the line of sight between Point of Interest and drone]?	-	Participant said that it means the bearing to that point.	User said that it represents that the drone is still searching the POI near Rivö, however, it has gotten another navigation. Thus, the new POI is near Asperö instead. Camera goes towards Rivö and the flight navigation goes toward Asperö.
Drone Menu Before Control	Test Participant A	Test Participant B	Test Participant C

Battery	[Discussed in section Drone Menu in control]	Participant said that this is the battery level	[Discussed in section Drone Menu in control]
Coverage	[Discussed in section Drone Menu in control]	Participant said that this is the network	[Discussed in section Drone Menu in control]
Drone Ready	Participant said that this represented that the drone is ready.	Participant said that this represented that the drone is ready.	Participant said that this represented that the drone is ready.
Launch Box Ready	Participant said that it represented that the launch box is ready.	Participant said that it represented that the launch box is ready.	Participant said that it represented that the launch box is ready.
WNW 5m/s	Participant said that this represented the wind where the drone is located.	Participant stated that this probably was the wind.	Participants said that this was the wind conditions.
Drone Menu In control	Test Participant A	Test Participant B	Test Participant C
SSRS EOS2 - Näset	-	-	Participant said that "SSRS EOS2" is the name of the airline or drone and "Näset" is the starting point of the drone.
Hamburger Menu Button	-	-	Participant states that this is the menu button.
Battery	Participant said that the battery could represent the control's battery.	[Discussed in section Drone Menu before control]	Participant said that this is the energy information in percent and that they would like to have information about how much the drone needs to return back to the starting point.
Coverage	Participant said that this represented the drone's coverage.	[Discussed in section Drone Menu before control]	Participant state that this is the coverage represented in bars for the network we are communicating via and that it is the

			reception to the drone.
GoTo Near Rivö	-	Participant said that this is where the drone is going.	Participant said that this is the data which the drone is navigating towards.
1.7km / 1 min 53 s to GoTo	Participant was initially saying that this was the speed of the drone and then said that this represents how long it will take.	Participant said that this is the distance and time to GoTo-point.	Participant said that this represents distance and time.
SOG	Participant say they do not understand what SOG is. Then the participant guesses that is the speed the drone will have.	Participant said that this means speed over ground.	Participants said that this means speed over ground.
Airspeed	Participant express that they do not understand the difference between airspeed and SOG.	-	Participant said that this means the speed in air.
MSL	Participant express that they do not understand what MSL is. Participant gives a suggestion that one should be able to hover over an info box to see what the abbreviation stands for.	Participant is unsure and is guessing that it is height.	User states that this is the drone's height over sea and is wondering in relation to what.
VTarget	Participant said that this is the target velocity.	Participant said it is the speed that one wants to have.	Participant said that this is the target speed when the drone has arrived.
MSLTarget	-	Participant said that this is the height one wants to have.	-

Loiter Radius	Participant expressed that they do not know what loiter radius is.	Participant said that this is the radius of the loitering circle, (however this was after the loiter circle was explained to the participant earlier in the testing).	Participant said that this is the loiter radius when the drone has arrived.
Point of Interest	Participant said that Point of Interest is where one wants to go. Participant say that the point of Interest is the round circle and the fully featured line is GoTo.	-	-
Elevation	User said that this means always keeping a height of 30 m over sea or above everything.	Participant was sure this was the height, however, then got unsure what MSL meant since the participant stated that MSL was height too.	User said that this is the target height and that POI has a height of 30 meters. Participant also stated that elevation is the height of an obstacle, height is height over ground and that altitude is height over sea. Furthermore, the user said that one should not be able to change the elevation and that such a value should rather be presented to the operator.
What would you have changed if you had the possibility to do so? What is ambiguous?	Participant said that it would have been easier to understand the interface if the prototype would have had the functionality of moving the GoTo-point. The user thought the meanings of the dashed and solid line made sense. User proposed that Point of Interest should be changed to for example camera view. The participant	User said that when they have an incoming alarm, they usually type in where they are going by coordinates. The user gives the suggestion that one should be able to say where one wants to go by insert coordinates. Moreover, the user says that if possible it would be useful to automatically link the drone's route to	User suggests that the drone should have a preset take off flight so that other actors within the area know the drone's route. The operator would then have information when they have to make the next decision. The participant says that it is actually good that it "happens directly" when setting

	made a comparison with Point of Interests in marine charts and stated that they have a slightly different meaning there. Lastly, the participant says that one should be able to hover over abbreviations or a small info icon and see the whole word.	coordinate from the incoming alarm. Then the user asks how one could return to the initial state, i.e in the launch box. Then the user also asks if it will be possible to view the live camera feed.	a GoTo and POI. Then the user said that it would be a good idea with waypoints and to be able to plan in advance what happens after the operator has arrived at the final destination. The user said that it would be relevant to have the luminance and night vision value (chlorophyll amount and altitude of moon). Then the participant said that it would be a good idea to include the meteorological sight if the sight is bad. Moreover, the participant states that it could be relevant to tell whether it would be possible to take a picture or not before actually flying the drone to a specific location. In addition, the user said that there should be a checklist for the drone before taking control of a drone. The user also stated that information about POI and the drone's direction of approach should be included before taking control of a drone. Furthermore, the participant said that if the drone's launch box is near ships' normal routes, then potentially a timetable for the ships should be included within the interface.

			Lastly, the participant suggests that one could use audio as well in order to notify the operators about specific events.