Stage Five Report

Team N

Drone Fleet Management System

Members

Name:	Student ID:
Stéphane Dorotich	10154487
Kathryn Lepine	30044629
Andy Ma	30031216
Macks Tam	30000262
Nicholas Wasilewski	30042836

Dec. 2nd, 2020

—

CPSC481 F2020

—

Tutorial 02

Professor: Lorans Alabood

TA: Philmo Gu

Links:

Link to Portfolio: https://teamlepinee.wixsite.com/cpsc481teamn
Link to GitHub: https://github.com/stephanedorotich/TeamN.git

TABLE OF CONTENTS

1 Executive Summary

Team N, a group of University of Calgary students enrolled in CPSC 481 – Human-Computer Interaction I, applied the User-Centered Design process to develop a high-fidelity prototype for their Drone Fleet Management System. Their desktop application is designed for a user to oversee a fleet of local delivery drones. Through five stages, they researched, ideated, developed iterative prototypes, and reported their progress.

After choosing a project idea, they researched the user experience. Their findings informed their design choices and system requirements. With requirements in hand, they developed a low-fidelity prototype. This early prototype helped assess their initial ideas and develop a clearer understanding of the system. Then, they developed a high-fidelity prototype. To critically examine it, they performed a heuristic evaluation which identified 50 problems, the majority of which were addressed. With an ever-improving understanding of their system, they developed a final high-fidelity prototype, the makings of which are detailed herein.

2 Introduction

This report summarizes our journey as Team N following the User-Centered Design process. Team N consists of Stephane Dorotich, Kathryn Lepine, Andy Ma, Macks Tam, and Nicholas Wasilewski. Over the past three months, we were able to investigate, ideate, prototype, and evaluate, all using the UCD process. We will describe our experiences and efforts put into creating the Drone Fleet Management System, a system created to help with the management and coordination of a fleet of drones. Through our learning, dedication, and teamwork and by creating multiple iterations, we developed a detailed and comprehensive prototype.

Figure 1: An example of a delivery drone

3 Design Problem

Since the start of the COVID-19 pandemic, there is no question that delivery services have been in demand. With lockdowns and individuals' choices not to leave their homes, there has been a remarkable increase in the need for delivery services. This includes delivery services for not only food, but also general packages. Recently, we have seen an increase in drone delivery services, such as Amazon Prime Air. This increase may be because drones are cheaper to buy and maintain in comparison to driven automobiles. Additionally, drones can be autonomous and not subject to road routes and traffic. However, there are few systems available to manage a fleet of drones. That is, a system that monitors and takes care of all drones in the delivery systems to ensure that all deliveries are completed smoothly, and drones remain in good working condition. To clarify, the problem is a lack of systems for the coordination of drones; there are plenty of systems for ordering packages and food. By creating this system, we are satisfying the needs of a delivery service by overseeing its fleet of drones.

4 Design Solution

Our solution is the Drone Fleet Management System (DFMS). This is an interactive desktop application that allows the user to manage a fleet of drones for a delivery service. A trained operator will use the system and oversee a particular region of drone deliveries. This operator will understand various protocols necessary for drones. They will be able to monitor each drone and view the drone's respective status, map position, history, destination, and much more. The DFMS will notify the operator of any dangers or issues by changing the status of a drone from active to critical. With a critical status drone, the operator will have to resolve the issue with various tasks such as recalling the drone back to headquarters or taking manual control of the drone. We believe that the DFMS could be used by companies such as Amazon Prime Air who have a drone fleet that needs to be coordinated to deliver packages to customers.

5 End-User and Stakeholders

The end-user of the DFMS is the operator. The operator uses the DFMS to monitor the drones and resolve any issues that may occur. Operators have extensive knowledge of not only the DFMS but also the rules and regulations for flying drones. There are multiple different stakeholders involved with the DFMS. The first stakeholder is the delivery service who owns the fleet of drones and is responsible for coordinating the delivery of packages to customers. The second stakeholder is the customer who is receiving a package from the drone. The third stakeholder is the retailer who is using the service to deliver their goods to customers. The fourth stakeholder is the maintenance personnel who are responsible for the drone's physical operation, like maintenance and repairs. The fifth stakeholder is Transport Canada who is the national organization responsible for transportation policies which drones must abide by. The last stakeholder is the developer who is responsible for the design, development, and maintenance of the DFMS.

6 User Research Methods and Process

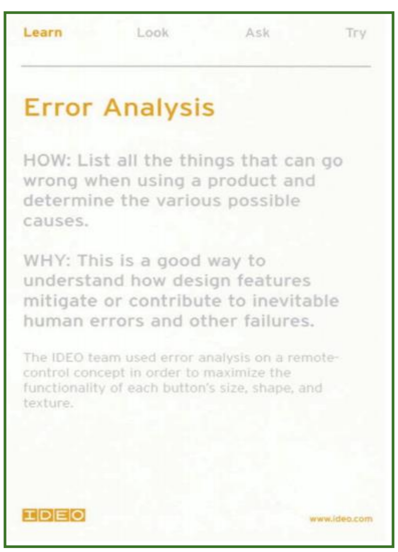


Figure 2: IDEO method card

We started by each member choosing three IDEO methods and presenting how these methods could be applied to learn about our system. We then voted and selected the three we thought best suited our project. This gave us synergistic methods that offered us a lot of insight into the project. Our first choice was Activity Analysis, which consists of listing in detail all actions, tasks, objects, performers, and interactions involved in a process. For Activity Analysis, we listed multiple processes and then reviewed each one, listing all the actions, objects, performers, and interactions for the process. This allowed us to identify which tasks we found extremely important for later stages. After completing the Activity Analysis, we found it logical and easy to transition into Error Analysis. In Error Analysis, we list all things that can go wrong when using a product and determine the various possible causes. For this research method, we proceeded by taking the current processes and listing all things that could possibly go wrong with our system. We listed every idea, and then talked about possible solutions for these issues. The last user research method chosen was Scenarios in which we created character-rich stories describing how our product could be used. Taking the information gained from our previous two research methods, we conceived specific situations that demonstrated how our product could be used and how problems that might arise could be solved.

7 User Research Findings

Our user research methods worked well together and allowed us to understand our project in greater detail. With Activity Analysis, we learned about our stakeholders, and tasks we envisioned for our project. This also allowed us to come up with our important tasks, such as View Map, View Drone Status, View Drone History, Drone List, etc. Our Error Analysis helped us identify possible issues with our current plan of the system and come up with solutions such as being able to operate the drone and making the drone's status apparent on the map. Finally, Scenarios allowed us to visualize our tasks and how our solutions could work if any problems arose. These scenarios allowed us to gain a deeper understanding of our system and how we planned to create it.

8 Design Choices and Justification

We made several key decisions when designing the DFMS. We decided that the DFMS would be a desktop application, because the large screen of a desktop is useful for displaying the status of a lot of drones on the same screen. We chose to use tabs for navigation, which enables the operator to multitask, and allows the most important areas of the system to be accessed, from any page, with a single click. We designed the main page to be a large, prominent map, so the operator can visually and simultaneously see the locations, trajectories, destinations, and statuses of all the drones on the screen. We designed an overlay on the map for the detailed status of a drone. This is efficient and allows the operator to see the drone on the map while viewing its status. We chose to add a dropdown list of drones to the map, so the operator can view the status of a drone which is off the screen and quickly find critical drones. We decided there would be a separate list of drones which can be searched, sorted, or filtered to find dormant drones and other drones without needing to find them on the map. We chose to make the user interface of the DFMS predominantly blue and white, so the green active drones and red critical drones stand out from the rest of the user interface.

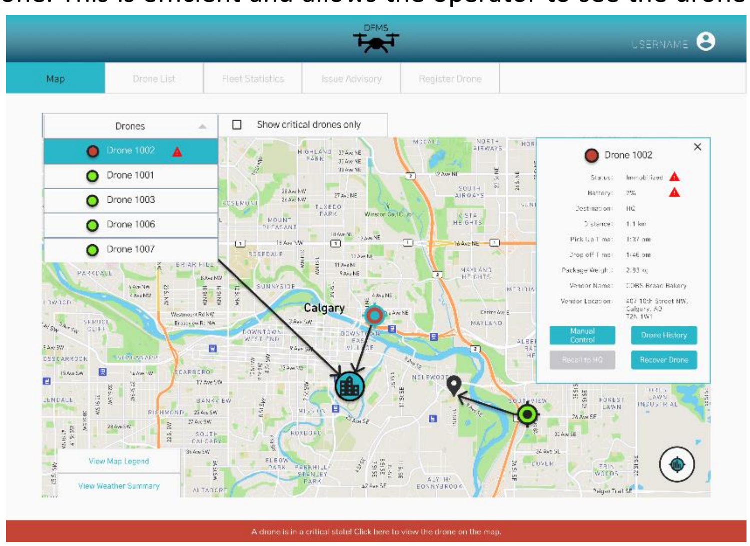


Figure 3: The results of our design choices

9 Low-Fidelity Prototype Design – Lessons Learned

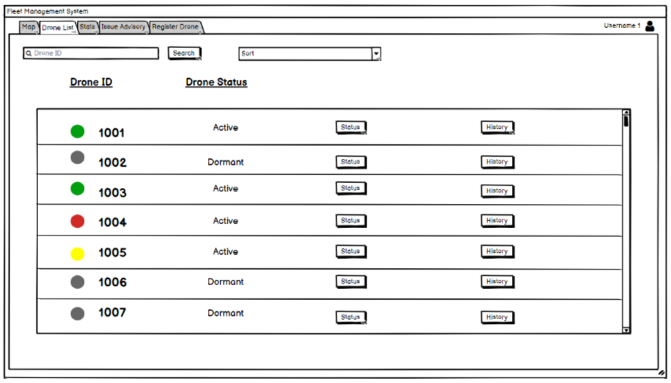


Figure 4: Drone list from low-fidelity prototype

Building the low-fidelity prototype gave us a lot of insight into our project. It formed the foundation for what our project is now. It allowed us to observe what we thought our project might look like before attempting to make a high-fidelity project. We learned which tasks were the essential to our system, namely the Map, Drone List, Drone Status, and Drone History. It also helped unify our ideas. Rather than having five different conceptions for the system (one from each member), after creating the low-fidelity prototype, our ideas coalesced into one. We discovered we had many concerns with our low-fidelity prototype, which informed what we wanted to implement, change, or leave out of our high-fidelity prototype. For example, we absorbed our drone status page into the map page, and we agreed on what information our drone list will display. This allowed us to begin the high-fidelity prototype with a better idea of how we expected the system to work.

10 High-Fidelity Prototype Design – Lessons Learned

Working on the high-fidelity prototype proved more difficult than we originally thought. We realized that we needed to add a legend to the map to increase ease of understanding in our system. We also learned we need to make sure our prototype is consistent, as we found a lot of small consistency errors with naming conventions we used. We also attempted creating overlays, especially on the map, but struggled to make it properly function with the rest of the system. For this stage, we decided to forgo the overlays on the map. We also decided to add a critical notification at the bottom of the screen for more clarity and warning for the user. As we proceeded to stage 5, we also learned to keep navigation through the system simple and clear, to not confuse the user. The two main lessons learned through building and testing the high-fidelity prototype were simplicity and consistency. We needed to make sure that a new user would not be overwhelmed and confused about the system and that our symbols and use of terms stayed consistent as well.

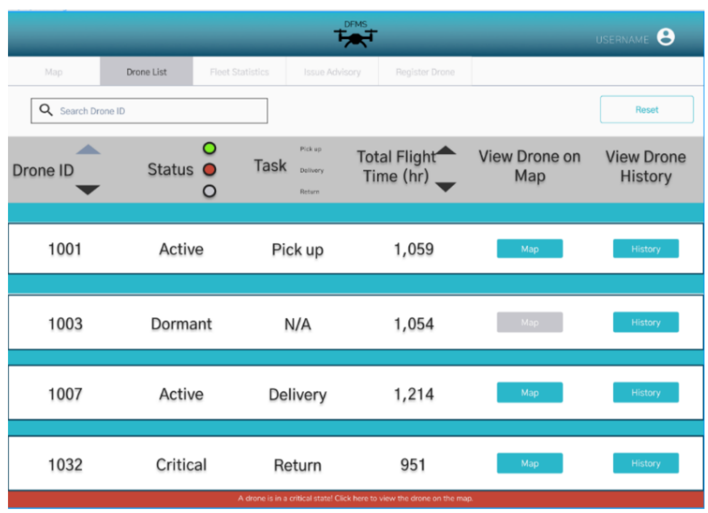


Figure 5: Drone list from first high-fidelity prototype

11 Heuristic Evaluation and Findings

We used the heuristic evaluation to assess the usability of our high-fidelity prototype. First, we split ourselves into groups of three evaluators and two reviewers. Next, each evaluator independently used Jacob Nielsen's 10 usability heuristics to generate a list of concerns with the design of our high-fidelity prototype. Then, the reviewers independently identified the problems raised by the evaluators and rated each on their severity. Finally, the reviewers met to discuss their findings, agree on the severity ratings, and decide which problems to fix.

The heuristic evaluation uncovered many usability problems with our high-fidelity prototype. The most severe issues were related to error prevention. The DFMS had no way to alert the operator when there was a drone in a critical state, and our Issue Advisory, Recall, and Drone Recovery Request designs lacked ways to prevent accidental actions. Our Manual Control page was missing the drone's ID, status, and location, as well as buttons to start and stop controlling the drone. We also identified many consistency issues, but they were generally less severe, with the biggest issue being the inconsistent default ordering of the drone list and the dropdown list of the map.

12 Stage Four Heuristic Evaluation Design Changes

To follow up on the heuristic evaluation, we needed to make a lot of design changes. We changed the bar at the bottom of the window so that when there is a drone in a critical state, it turns red to alert the operator and can be clicked to view the detailed status of the critical drone. We added confirmation popups to the Issue Advisory, Recall, and Drone Recovery Request designs, to prevent accidental actions by the operator. We added the missing drone ID, status, location to the Manual Control page, as well as buttons to Assume Manual Control and Return to Autopilot. We fixed the consistency issues by making the parts that were inconsistent the same. For example, both the drone list and the dropdown list of the map now show critical drones before active drones by default, with drones of the same status being sorted by ascending drone ID.

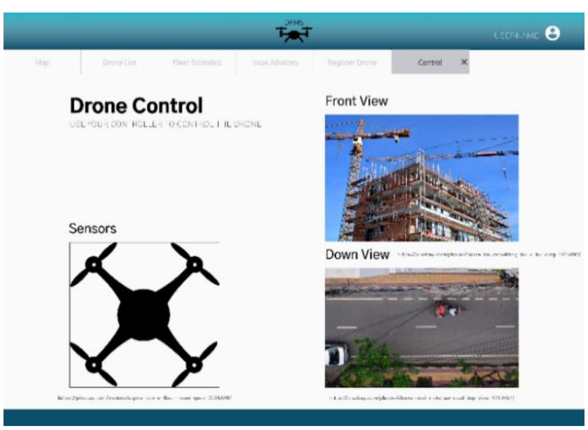


Figure 6: Drone control before heuristic evaluation

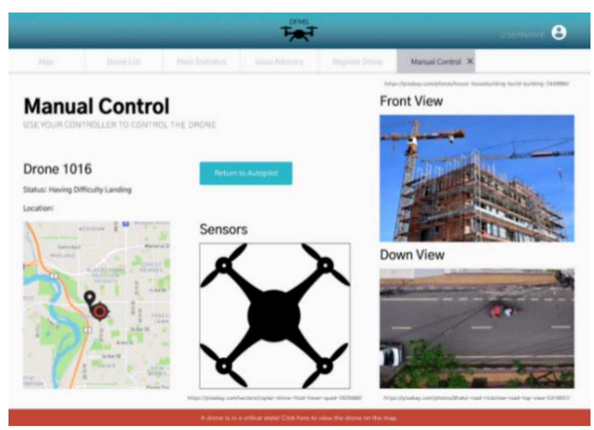


Figure 7: Drone control after heuristic evaluation

13 Stage Five Design Changes and Future Changes

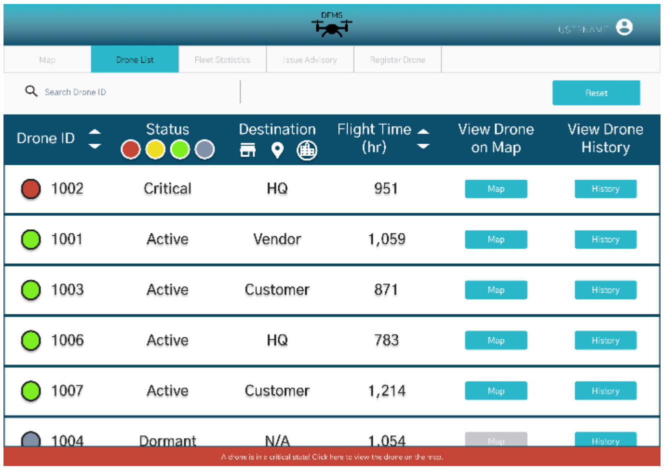


Figure 8: Drone list in final prototype

In stage five, we fully implemented two more tasks: drone recovery and register new drone. We also made many visual and functional changes to our current system. First, we added a legend to the map, removing any confusion of symbols. Secondly, we changed two label names, giving more clarity to the user: “Task” to “Destination” and “Recall” to “Recall to HQ”. Lastly, we made changes to some of the previously implemented tasks, such as scroll functionality, adding status color to drone list, clicking tab to reset, etc. These changes increase the user's ability to navigate seamlessly through the system, as well as clarify things that could be confusing. In the future, keyboard shortcuts, weather overlays for the map, zoom functionality for the map, and upgrading the remaining horizontal tasks should be implemented. Furthermore, other suggested changes in the future could be to add drone history navigation (allowing navigation to other drones in history rather than going back to the list tab), submitting maintenance requests, and support different type of drones (lightweight, heavyweight, etc.).

14 Conclusion

Throughout our project journey, there were lots of challenges. There were many times we felt frustration and uncertainty about how certain aspects should be implemented or how to proceed with next steps. The UCD process required us to think hard about our system both individually and as a team to overcome obstacles. By jumping over these hurdles, we were able to experience the success of creating a system that is appealing, usable, and well considered and that allows operators to manage and coordinate a fleet of drones. As we suggested above, there still many changes that could be made to the DFMS to improve the user experience. However, we believe that our iterations of the Drone Fleet Management System represent diligent teamwork and a successful implementation of the UCD process.

Appendix

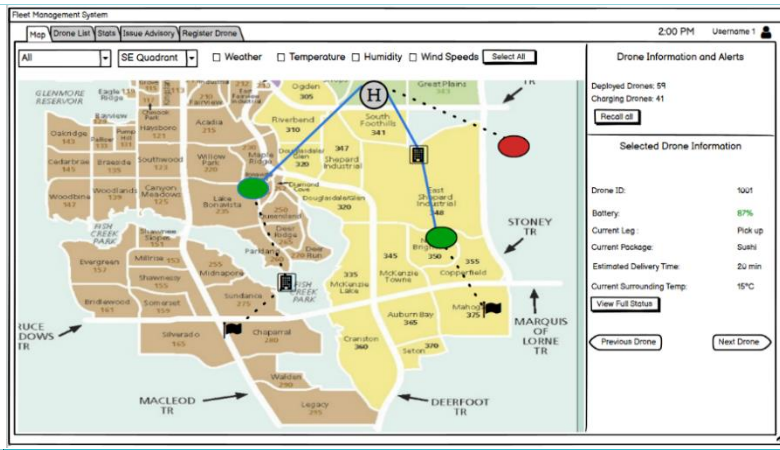
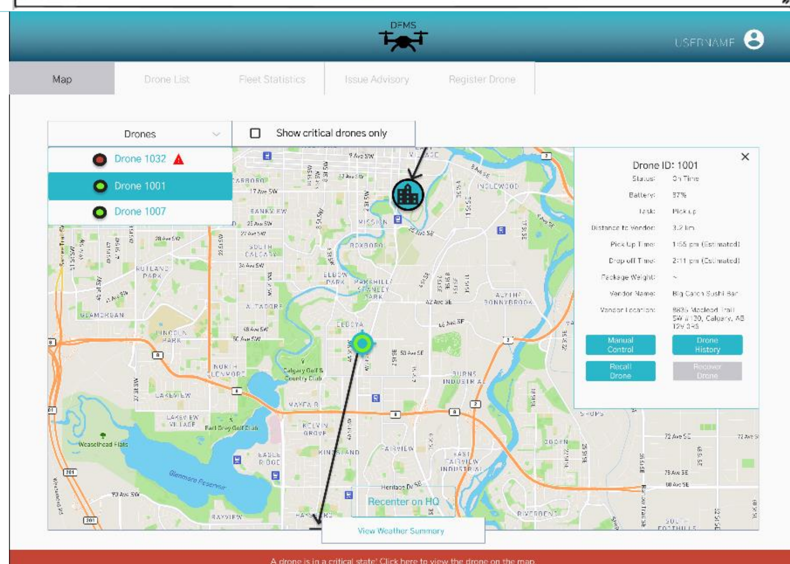
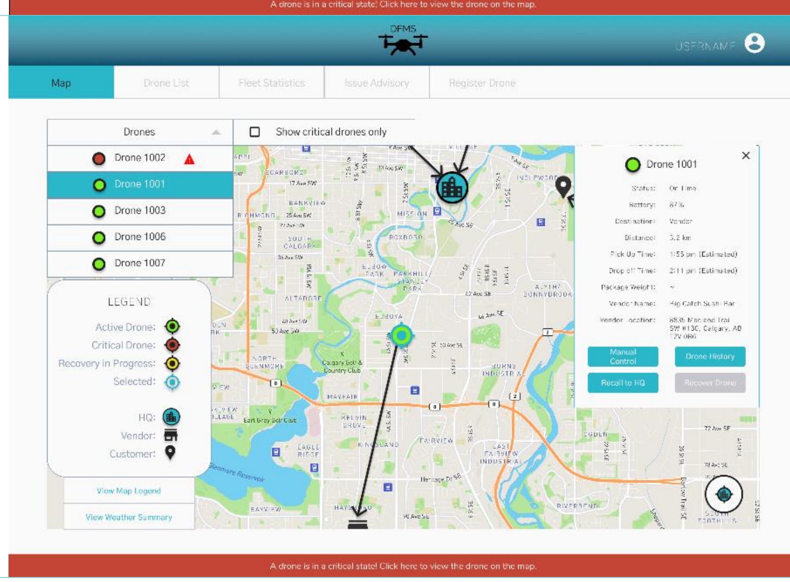
Image Sources

Figure 1: https://flytrex.com/static/projects/pictures/xdrone-pic.png.pagespeed.ic.Hxo1Rt_muT.png

Figure 2: http://hcitang.org/uploads/Teaching/ideo-method-cards-2by1.pdf

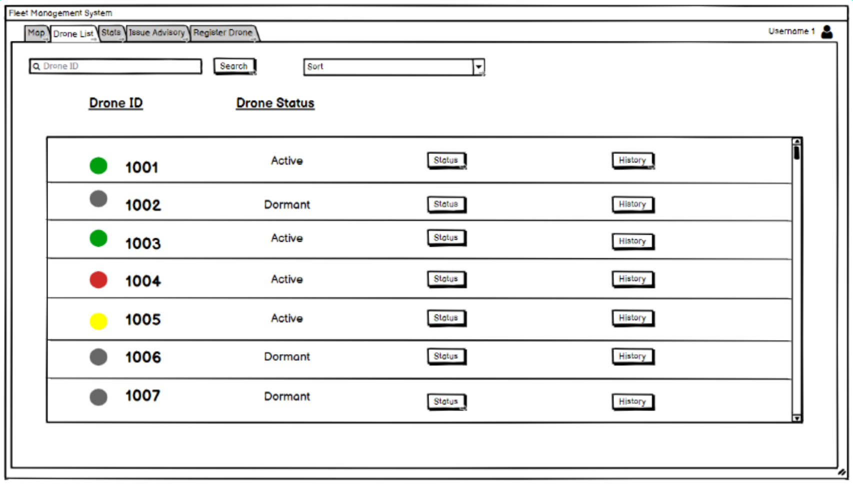
BEFORE & AFTER SCREENSHOTS

Map View

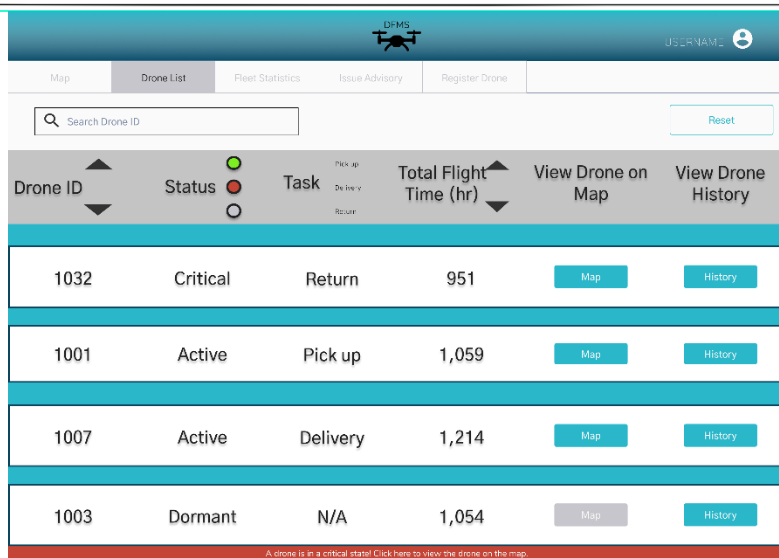
<table>
<tr><td>Stage Three</td><td>
</td></tr>
<tr><td>Stage Four</td><td>
</td></tr>
<tr><td>Stage Five</td><td>
</td></tr>
</table>

Drone List

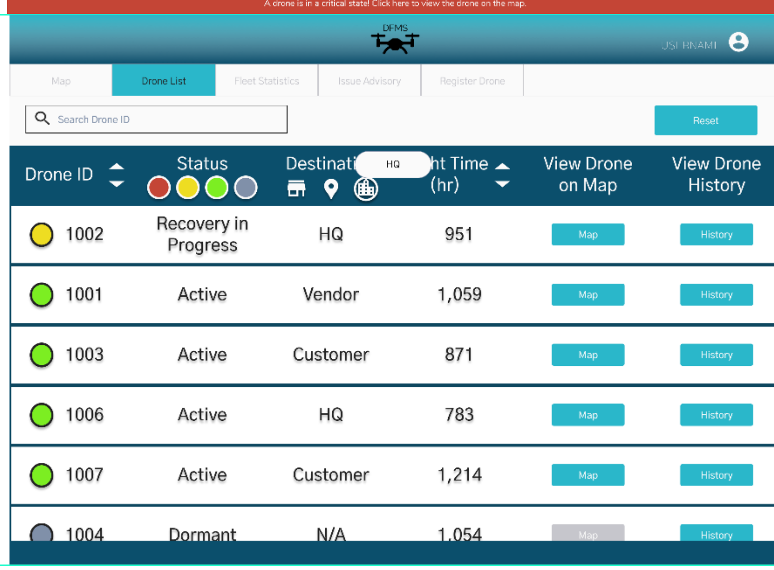
Stage Three



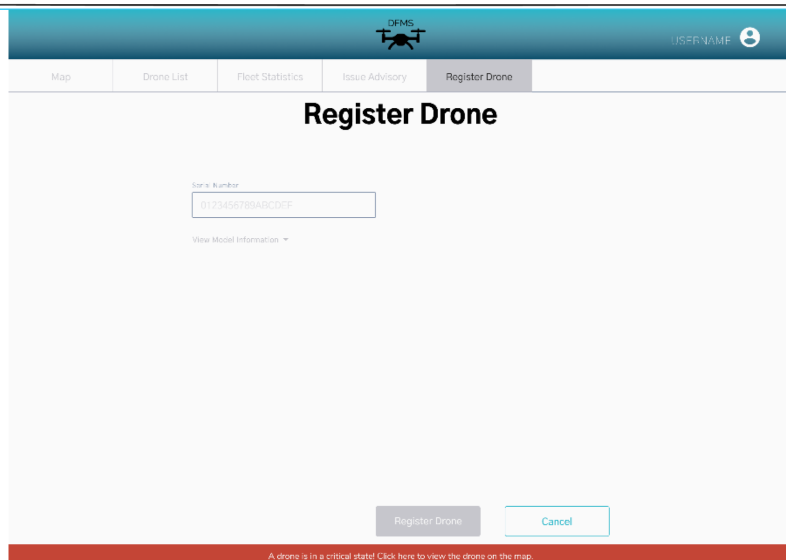
Stage Four



Stage Five



Register Drone

Stage Three	
Stage Four	
Stage Five