Story

Modern Lifeguards use intelligent drones to search for people that are in danger at sea and need help. The lifeguards need a solution that support their daily work: They start the search flights of the drones. If a drone detects a possible emergency, the lifeguard is alerted. He/she receives the geo coordinates, as well as the video and audio signals of the emergency location. He/she then decides about the follow-up activities (further scan, trigger rescue, continue search etc.). If the lifeguard triggers a rescue operation, the app supports decision-making by providing the suited rescue vehicle (helicopter, sea rescue ship etc.).

Persona – IoT Prototype

Sean Sea-Eagle
The Lifeguard (Drone Operator)

“Protect Lives”

About
- 35, single, 10 years of lifeguard experience.
- Stationed on a sea rescue ship in the Mediterranean Sea
- I work with the other members of the international sea rescue teams (lifeguards, helicopter pilots, captains of sea rescue ships etc.)

Responsibilities
- Coordinate drone search flights to detect emergency situations at sea
- Analysis of drone scan data to detect emergency situations
- Decide on follow-up actions if an emergency situation is detected
- Trigger rescue activities

Main Goals
- Save lives
- Execute the appropriate actions in case of an emergency situation
- Don’t panic!

Needs
- I need information about the drones (status, location, current activity
- I need precise data about the emergency situation
- I have to make the right decisions fast and quick
- I need an effective and efficient way to initiate rescue activities

Pain Points
- Missing possibility to communicate with the intelligent drones
- Wrong decisions might cause the death of human-beings
Point of View

As a coordinator of drone search flights I need a way to monitor and analyze their data so that I can trigger the appropriate follow-up activities for a successful rescue.
### User Experience Journey Template

<table>
<thead>
<tr>
<th>ACTIONS</th>
<th>ACTIONS</th>
<th>ACTIONS</th>
<th>ACTIONS</th>
<th>ACTIONS</th>
<th>ACTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check drone status and start search flight</td>
<td>Monitor the drone status</td>
<td>Wait for scan results of possible emergency situation</td>
<td>Evaluate the scan data</td>
<td>Decide which follow up-activity is appropriate</td>
<td>Execute Follow-Up activities</td>
</tr>
<tr>
<td>“Weather is good, let’s get started”</td>
<td>“Where are the places of strong currents?”</td>
<td>“Hope we do not detect emergency situations”</td>
<td>“False alert?” “Oh, no, this is pretty bad!”</td>
<td>“They need our help as soon as possible”</td>
<td>“They are rescued”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEELING</th>
<th>FEELING</th>
<th>FEELING</th>
<th>FEELING</th>
<th>FEELING</th>
<th>FEELING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drones</td>
<td>Monitor app</td>
<td>Communication with drone via monitor app</td>
<td>Monitor app</td>
<td>Monitor app Other sea rescue team members</td>
<td>Monitor app Other sea rescue team members</td>
</tr>
<tr>
<td>TOUCH POINTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MINDSET**
Prototype

BUILD prototype:
https://standard.build.me/api/projects/73756187452b02d50cd27e9a/prototype/snapshot/latest/index.html#/14774219740681544_S0

Remarks BUILD prototype:
- Use Google Chrome as web browser
- Keep Tablet as UI option
- Navigation back button does not work (tool error)
- Icon for Search Status does not change if the user presses the Start, Pause, or Stop buttons

Entry screen:
Based on the activity that the intelligent drone executes, the appropriate pushbuttons are activated:

<table>
<thead>
<tr>
<th>Drone</th>
<th>Start</th>
<th>Pause</th>
<th>Stop</th>
<th>Search Status</th>
<th>Action Required</th>
<th>Flight Time</th>
<th>Search Status Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00:05:30</td>
<td>on the way to the search area</td>
</tr>
<tr>
<td>Drone 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beluga</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>02:16:17</td>
<td>return to base</td>
</tr>
<tr>
<td>Drone 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stingray</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>01:14:27</td>
<td>found potential emergency case</td>
</tr>
<tr>
<td>Drone 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dolphin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00:00:00</td>
<td>ready to start search</td>
</tr>
<tr>
<td>Drone 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Start/Pause/Stop search flight of a drone (button action = display message)

Once drone has detected a potential emergency, the user is able to trigger handling of it (by pressing the activated Act! Button): The drone transfers the location of the emergency and a picture/video of it. The intelligent drone uses its sensors to determine the issue and estimates the number of affected human beings.

To get more details about the emergency, the user can trigger a more detailed scan of the emergency location (button action = message display) if required. He/she then can start the rescue. Because the drone detected strong indicators that an emergency is happening, the button to stop/skip the rescue/scan activities is inactive.
By pressing the Start Rescue button, the user can decide about the required elements of the rescue chain: Based on the response of the intelligent drone, some checkboxes are automatically ticked (can be unticked by the user).

The user can rate the emergency by choosing a value from the combo box:
It is possible to add a voice message that is transferred to the headquarter when the rescue chain is activated. The user can also directly contact the headquarter (pushbutton Contact Headquarter) if he/she has to clarify how to handle open issues.

Finally, the user triggers the execution of the rescue by pressing the Execute Rescue button. Pressing the Reject button allows the user to change his/hers decision (button action = navigate to previous screen).