

NAVSEA - Product Requirements Document

About the Team

Team Name: Sea++

Company: Navy Pt. Mugu

Team Members:

- Lyuda Panina (Lead)
- Vivian Ross (Scribe)
- Emily O'Mahony
- Kyle Kam
- Thao Phan

Introduction

What is the problem?

Navy ships contain multiple systems, ranging from weapons to communication, that need to be maintained. These systems are up to forty years old, and they need maintenance in the form of repairs, upgrades, or retrofits. The existing model for undertaking maintenance involves sending a trained expert to a ship in order to work with the system. This is resource intensive and inefficient.

How is the problem being addressed today?

Existing AR applications exist to walk users through systems maintenance. These applications work by overlaying a model of the system that the user needs to interact with and using this model to highlight the components that the user needs to interact with. This application has its benefits in that it is self contained. This eliminates the need for the trained expert, since the application contains all of the information necessary for a user to maintain the machine. However, this presents problems in that it is necessary to build the application for every system that needs it. Given that some systems are forty years old, it may be difficult to find people that have both the specialized systems knowledge and the technical expertise to develop the HoloLens model. Some of the existing technologies that serve this functionality include:

- Four Eyes Lab - A research application that uses video conferencing and remote scene exploration in order to allow spatial annotations to be viewed in augmented reality
- Navy Research Application - A HoloLens application that can overlay schematics with step-by-step instructions on to the real world counterpart

Objective, Implementation, and Assumptions

What is our objective and who will benefit from it?

Our goal is to develop an application to allow a system expert to help an untrained sailor maintain a system. This would save on resources by eliminating the need for the expert to be transported to the ship, and it would maximize efficiency by allowing the expert to work remotely on multiple systems from the same location. This application will be an augmented reality based application that will be deployed on a HoloLens 2. The augmented reality interface will allow the expert to more clearly direct the sailor on how to maintain the system and conduct repairs.

How will we implement our solution?

We intend to stream video input from a HoloLens 2 to a remote computer. The person using the computer will be able to add arrows, highlighting, or other relevant annotations to the video stream. The person using the HoloLens will be able to see these annotations projected onto their video stream. We will achieve this with a Unity app configured for use with HoloLens and the Mixed Reality Toolkit.

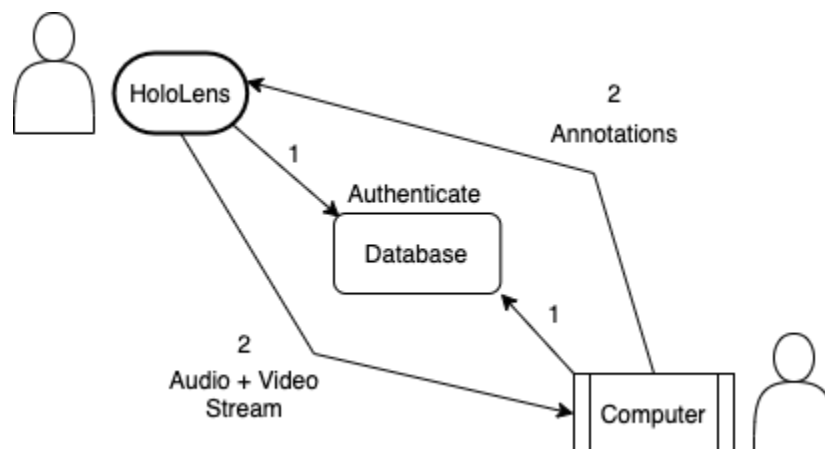
What are the necessary assumptions to be made?

Our application will assume that the person on the ship wearing the HoloLens has a clear vision of the broken machine: adequate lighting and no obstructions in the way on the ship. The HoloLens user and the experts both must be logged into the application in order to start streaming/viewing. Authentication is necessary for tracking in case something goes wrong or there is malicious behavior from any party.

System Architecture Overview

The diagram below shows the data flow of our interactive applications. There are two users, the HoloLens user, and the computer user. At the beginning, each user will log in to their own unique accounts, with their information authenticated by the database that stores user information. Once logged in, the HoloLens user will create a name for their video stream and the computer user will be able to see the names of the available streams on their end, and select the desired one.

Once the HoloLens and computer begin interfacing, the computer user will be able to see the video feed from the HoloLens. The computer user will be able to speak into a microphone and the HoloLens user will be able to hear them and verbally respond into the microphone in the HoloLens. The computer user will be able to hear the response on their own speakers. In addition, the computer user will be able to apply visual annotations to the video feed from the HoloLens, and the HoloLens user will be able to see the annotations appear in their own line of sight. This audio, visual, and video communication will be facilitated by the Unity program we write for the HoloLens, as well as our own application on the computer.



User Stories

As a user, I can use the HoloLens on a ship and stream video from the HoloLens to a computer remotely.

As a HoloLens user, I will put on the HoloLens and start the app. I will immediately see a login screen that has prompts for a username and password, a stream name, a “Submit” button, and a “Create Account” button.

[GitHub issue](#)

- Scenario 1: The user has a unique login and types in their information. They choose a name for this video stream to be represented as on the land side and type this into the stream name prompt. Finally, they press the “Submit” button and the login screen disappears.
- Scenario 2: The user does not have a unique login. They press the “Create Account” button and see a new screen with prompts for a username and password, and a “Submit” button.

As a HoloLens user, I can create a new unique login to use the application.

[GitHub issue](#)

- Scenario 1: The user selects the “Create Account” button on the login screen, enters a new username, password, and presses submit to create a new account. The login information gets stored in a database of user information. Then, they get taken back to the login screen.
- Scenario 2: The user selects the “Create Account” button on the login screen but no longer wants to create a new account. They press the back button to go back to the login screen.

As a HoloLens user, I can rotate my left palm towards my face to pull up a menu.

[GitHub issue](#)

- Scenario 1: The user is not currently streaming. The menu contains 2 buttons: “Start Streaming” and “Log Out”.
- Scenario 2: The user is currently streaming. The menu contains 2 buttons: “Stop Streaming” and “Log Out”.

As a HoloLens user, if the menu is currently being displayed, I can turn my left palm away from my face to remove the menu.

[GitHub issue](#)

- Scenario 1: I am currently not streaming, the menu that contains the “Start Streaming” and “Log Out” buttons will go away.
- Scenario 2: I am currently streaming, the menu that contains the “Stop Streaming” and “Log Out” buttons will go away.

As a HoloLens user, if the menu is currently active, I can choose to “Start Streaming” if the stream is not online, “ Stop Streaming” if the stream is online, or “Log Out” whenever.

[GitHub issue](#)

- Scenario 1: The menu is active, the user is not currently streaming and presses “Start Streaming”. This streams video and audio so that a user at a computer can log in, view the stream, and hear the audio, identifiable by the “stream name” specified upon login.
- Scenario 2: The menu is active, the user is currently streaming and presses “Stop Streaming”. This stops streaming video and audio to the computer user and removes the stream from viewing by the computer user.
- Scenario 3: The menu is active and the user presses “Log Out”. If the user is currently streaming, the stream will close before the user is directed back to the login page. If there is no stream active, the user will be directed back to the login page.

As a HoloLens user, I can see the annotations being made by the remote user. The annotations will remain in the same “location”, e.g. pointing to the same piece of machinery, even as the user moves their head.

[GitHub issue](#)

- Scenario 1: The user attempts to move an arrow or highlight annotation. The application will not allow the user to move any directional annotations since they are concrete and can only be modified by the computer user.
- Scenario 2: The user attempts to delete a directional annotation. The application will not allow the user to delete any directional annotations since they are concrete and can only be modified by the computer user.
- Scenario 3: The user attempts to move a note annotation to clear obstructions on the HoloLens. The note annotation will move with the user’s input.

Alternatively, I can use the accompaniment computer interface to add annotations and view the ship user’s video stream.

As a computer user, I will start the app on the computer. I will immediately see a login screen that has prompts for a username and password, a “Submit” button, and a “Create Account” button.

[GitHub issue](#)

- Scenario 1: The user has a unique login and types in their information. Finally, they press the “Submit” button and the login screen disappears if the login is correct.
- Scenario 2: The user has a unique login but the input username and/or password is incorrect or does not exist. An error message will appear on the screen.
- Scenario 3: The user does not have a unique login. They press the “Create Account” button and see a new screen with prompts for a username and password, and a “Submit” button.

As a computer user, I can create a new unique login to use the application.

[GitHub issue](#)

- Scenario 1: The user selects the “Create Account” button on the login screen, enters a new username, password, and presses submit to create a new account. The login information gets stored in a database of user information. Then, they get taken back to the login screen.

- Scenario 2: The user selects the “Create Account” button on the login screen but no longer wants to create a new account. They press the back button to go back to the login screen.

As a computer user, I can see a list of online streams I can join.

[GitHub issue](#)

- Scenario 1: The user is signed in and can see a list of online streams they can choose to join. There will also be a “Log Out” button at the bottom right of the screen.
- Scenario 2: The user is not signed in. They are not able to see any stream lists because they are not signed in.

As a computer user, I can choose to join a stream from the online streams list

[GitHub issue](#)

- Scenario 1: The user chooses to join a stream. A window of the stream pops up along with a back button to take the user back to the list view, a toolbar with available annotations they can drag onto the stream, and the “Log Out” button.
- Scenario 2: The user chooses to join a stream but is unable to load it because of an error (no connection, stream offline, etc.) An error message will be displayed.

As a computer user, I can choose to log out anytime

[GitHub issue](#)

- Scenario 1: The user is viewing a stream when they log out. The stream will close and will no longer transmit audio to the HoloLens user. The computer user is taken back to the login screen.
- Scenario 2: The user is not viewing a stream when they log out. The user will be taken back to the login screen.

As a computer user, I can add an arrow, circle, or highlight annotation to the stream

[GitHub issue](#)

- Scenario 1: The user is able to add the arrow, circle, or highlight annotation successfully. The arrow will track the object it is placed on and will remain there until the computer user chooses to modify it.
- Scenario 2: The user is unable to add the arrow, circle, or highlight annotation successfully. An error message appears on the screen to let them know.

As a computer user, I can send messages to the stream

[GitHub issue](#)

- Scenario 1: The user is able to send the message successfully. The message will show up on the text log for the HoloLens user to read.
- Scenario 2: The user is unable to send the message successfully. An error notification will show up on the screen alerting that there is an error.

As a computer user, I can click the “clear all” button to clear all existing annotations.

[GitHub issue](#)

- Scenario 1: The clear all function succeeds and all of the annotations disappear from the HoloLens render
- Scenario 2: The clear all function fails and the computer user gets an error message

As a computer user, I can select an existing annotation and delete it.

[GitHub issue](#)

- Scenario 1: The user successfully selects an annotation either directly on the screen or the drop down menu of existing annotations to delete a certain annotation.
 - Scenario 1a: The user selects the annotation on the screen, a box appears with an x on it. The user can delete the annotation by clicking on the x. They can deselect the annotation by clicking elsewhere on the screen.
 - Scenario 1b: The user selects the annotation on the drop down menu, they can right click and choose delete on the annotation to delete it.
- Scenario 2: The user is not able to delete the annotation successfully. An error message will appear on the screen.

As a computer user, I can resize an existing annotation.

[GitHub issue](#)

- Scenario 1: The user resizes the annotation successfully by clicking on the desired annotation, a box will appear, and the user can drag the corner of the box to resize it.
- Scenario 2: The user is not able to resize the annotation successfully. An error message will appear on the screen.

As a computer user, I can move an annotation around the screen.

[GitHub issue](#)

- Scenario 1: The user moves an annotation successfully by clicking on it, a box appears, and dragging the box around the screen.
- Scenario 2: The user is not able to successfully move an annotation. An error message will appear on the screen.

Technologies Employed

- Frontend:
 - Unity (C#)
 - Mixed Reality Feature Toolkit
 - ReactJS
- Backend
 - Firebase