CMPT 371 – TEAM 3 REQUIREMENTS DOCUMENT

Virtual Reality Medical Imaging Software with Luxsonic Technologies Inc.

February 17, 2017

Executive Summary

This document provides information to describe the features that will be included in the produce: virtual reality medical imaging software. All members of Dr. Nathaniel Osgood's CMPT 371 Team 3 will have an important role in the development and implementation of this program. Using the Unity 5 engine, the Oculus Rift VR headset, and other necessary hardware and software, we plan to deliver a functional program by the end of the project.

The planned functionality of the program falls into three categories: must-have, should-have, and could-have. Must-have features will be those promised by our team to deliver to Luxsonic by the end of the project. These features will be included in our step-through graphical user interface concept shown in the appendix of this document. Each must-have feature will include a description and set of possible user actions and scenarios. The should-have features are those that we plan to implement, but due to time constraints, cannot promise by the end of the project. A short description of these features is provided, along with the likelihood of being able to deliver them. Finally, could-have features are those that we will be unlikely to implement due to time or other constraints, but should be taken into consideration when designing the software. With the information provided in this document, Luxsonic should have an in-depth vision of the expected project for our first deliverable and final product. Since we will be employing an incremental development approach, many of the features are subject to change due to budget concerns and Luxsonic's input.

Table of Contents

1 Premise

This document is the requirements outline between the student software engineers of Dr. Nathaniel Osgood's CMPT 371 Team 3 and our client Luxsonic. Our team consists of 11 members: Brady, Brianne, Evan, Heramb, Ix, Jack, Janelle, Kevin, Kyle, Mackenzie, and Nhi. Our primary goal is to engineer a piece of software to replace a radiologist's traditional workspace with virtual reality 3D medical imaging software. We will first prioritize the rendering of 2D DICOM images and later advance to 3D. This document will include a description of actors (entities which may interact with the software system), actions which those actors may take on the system, scenarios, use case diagrams, and prototype interfaces to help foster understanding of the software.

2 Product

The product will be a virtual reality based medical imaging software. Using Unity and Oculus Rift our teams' goal is to engineer a piece of software to improve the workflow of medical imaging professionals. The software will create a configurable 3D space that contains virtual analogues of the tools used daily by radiologists. The software will support and display images that comply with the Digital Imaging and Communications in Medicine (DICOM) standard in both 2D and 3D views. Users will be able to cycle through these images as well as adjusting the depth, contrast, and zoom of the image. The product will support the display of multiple images at once, and will be controlled primarily using the Oculus Rift's Touch controllers. The images displayed within the application may be rearranged within the 3D environment of the workplace according to the user's preferences. The team will also tackle any problems relating to motion sickness or eye strain to ensure the product is usable for long hours.

3 Platform and Software

The application will be developed for 64-bit Windows systems capable of running Oculus Rift software. The engine used to create the software for the project will be Unity 5.4 and all project code will be written in C# and stored in a Github repository. Assets provided in Unity's Asset Store for virtual reality may downloaded and used if necessary to assist with development of the project. Any additional software not mentioned above may be used and will be disclosed in future versions of this document.

4 Functionality

It is important that this program to be user-friendly for any radiologists who wish to use the program. The goal is that it should be easy to handle regardless of any one user's technological background. The actions within the program should be intuitive and responsive. Careful consideration will be given to how each feature will work within the VR space, to optimize for ease of use and feasibility. Within a virtual environment, the health risks (such as nausea, muscle tension or dissociation) must be taken into account. The team will determine if any of the features or capabilities implemented could have adverse effects. These risks will be assessed on a regular basis and the software adjusted accordingly. Therefore, it is necessary that amendments will be made to this document on a regular basis. After each deliverable, the product will be evaluated on its features, impact on user health, and its usability.

5 Must-Have System Actors and Their Actions

This section will list actions that are integral to the functionality of the software. It will describe the features necessary to allow the developed software to be functional. Optional features that we would like to implement will be outlined in the following section.

Actors are possible entities (human or not) which may interact with the software system. Here we describe the purpose of each actor along with the details and process of each action. Several considerations made for each action include pre-conditions (what must happen before an action's scenario can occur), post-conditions (the result expected at the end of a scenario), and error-conditions (what may occur that prevents a scenario from reaching the expected end-point).

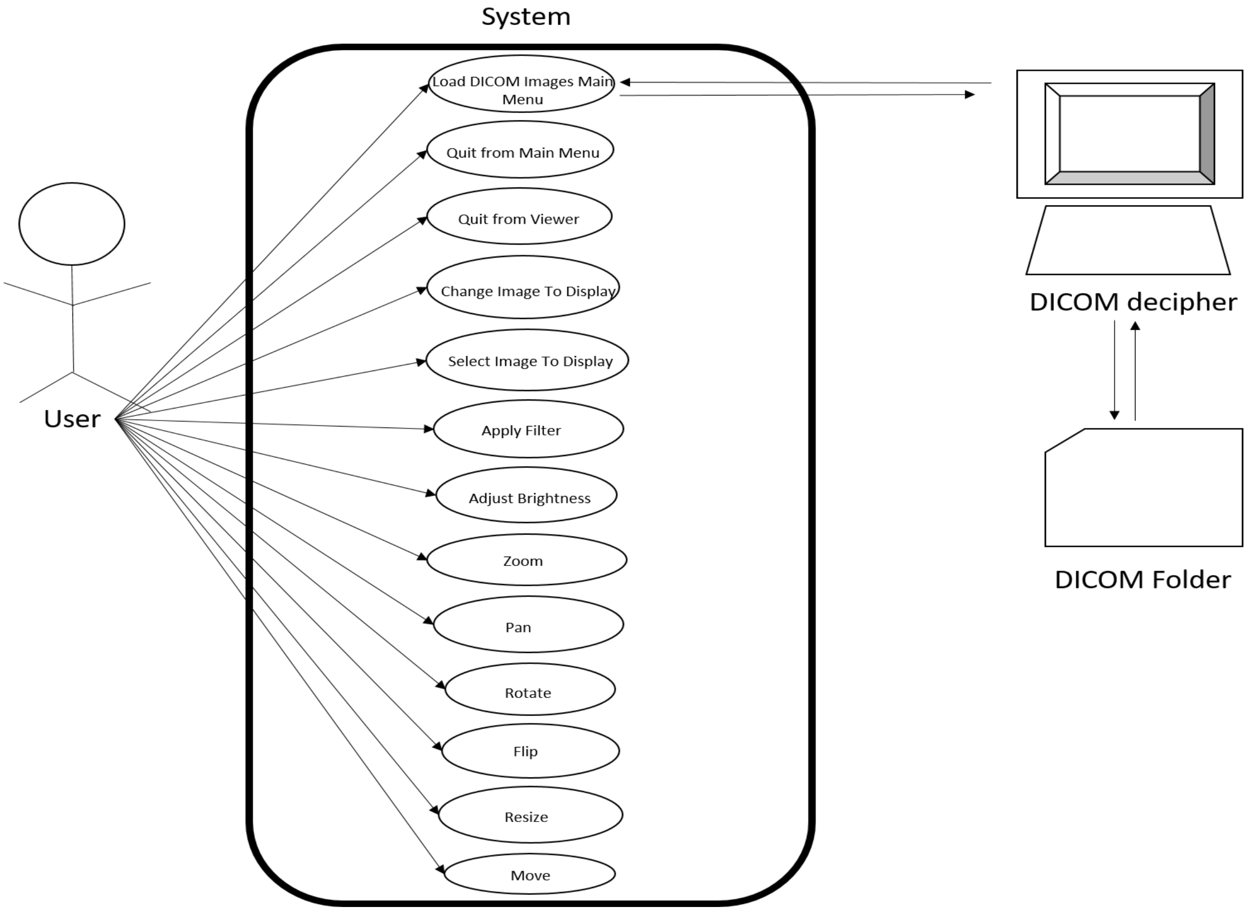
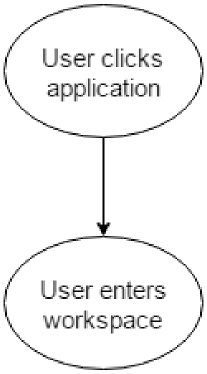


Figure 1: System Diagram for the software
This diagram represents the minimum set of features required for the software to be functional.

5.1 Actor: User

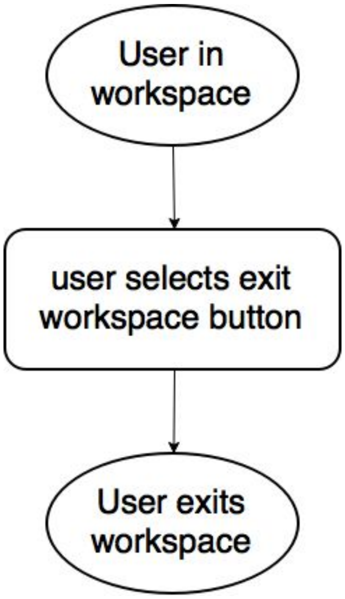
The User is the person interacting with the system. The User will initialize the program, select files for use, and use the workspace to interact with the specified files. When interacting with the files, the User may change the contrast of, brightness of, apply filters to, zoom in and out of, rotate, flip, move, and resize the image.

5.1.1 Action: Initialize Program



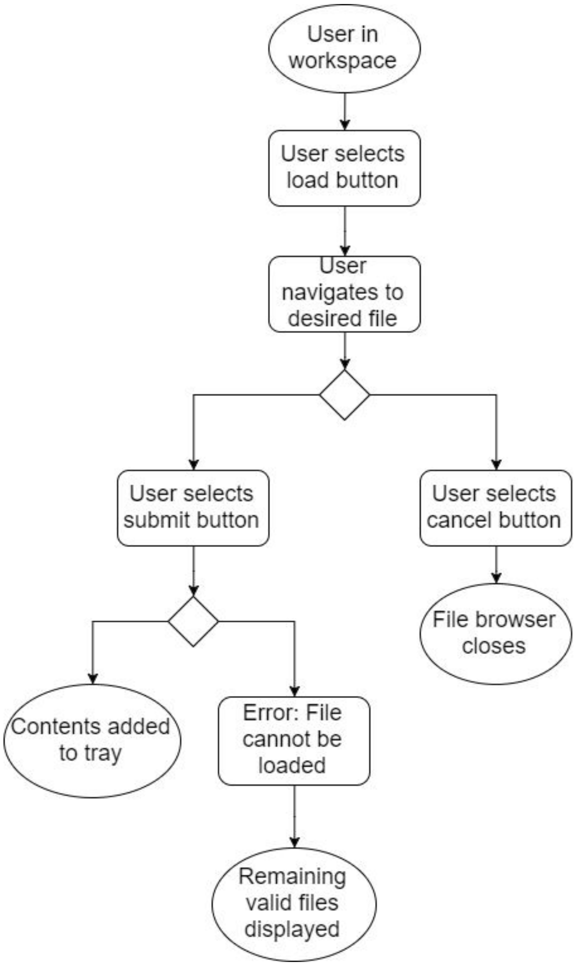
- Pre-condition:
  - None
- Post-condition:
  - User is in application
- Basic path:
  1. User clicks application
  2. User enters workspace

5.1.2 Action: Quit Application



- Pre-condition:
    - User in application
- Post-condition:
    - Application is closed
- Basic path:
    1. User presses Quit button
    2. Application is closed

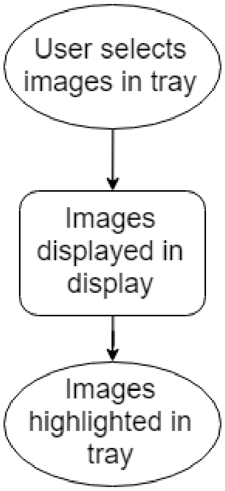
5.1.3 Action: Load DICOM File


User in workspace
User selects load button
User navigates to desired file
User selects submit button
User selects cancel button
File browser closes
Contents added to tray
Error: File cannot be loaded
Remaining valid files displayed

- Pre-condition:
    - User is in workspace
- Post-condition:
    - Files loaded into tray to display
- Basic path:
    1. User in workspace
    2. User selects Load button
    3. User navigates to required file
    4. User selects submit
    5. Contents added to Tray
- Alternative path:
    1. User presses cancel in file browser
    2. File browser closes
- Error-condition path: Files not found

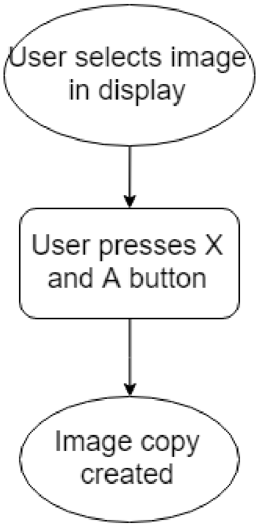
1. User selects submit
2. Error message notifies users which file cannot be loaded
3. Remaining (valid files) displayed in Tray

5.1.4 Action: Select Images to display from Tray


User selects images in tray
Images displayed in display
Images highlighted in tray

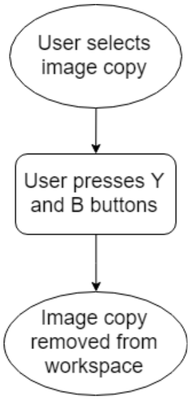
- Pre-condition:
  - User in workspace
  - Tray contains loaded images
- Post-condition:
  - Images displayed in Display
- Basic path:
  1. User selects desired images in Tray
  2. Images displayed in the Display
  3. Images displayed highlighted in Tray

5.1.5 Action: Select Images to copy from display



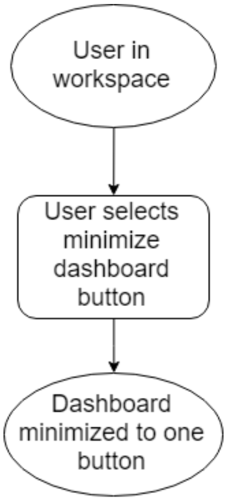
- Pre-condition:
  - Images are loaded in Display
- Post-condition:
  - Image copied from dashboard
  - Copied image transformed into Unity Game Object
- Basic path:
  1. User selects images in Display with A and X button on controller
  2. Image is copied and transformed into Unity Game Object

5.1.6 Action: Remove Copy from workspace


User selects image copy
User presses Y and B buttons
Image copy removed from workspace

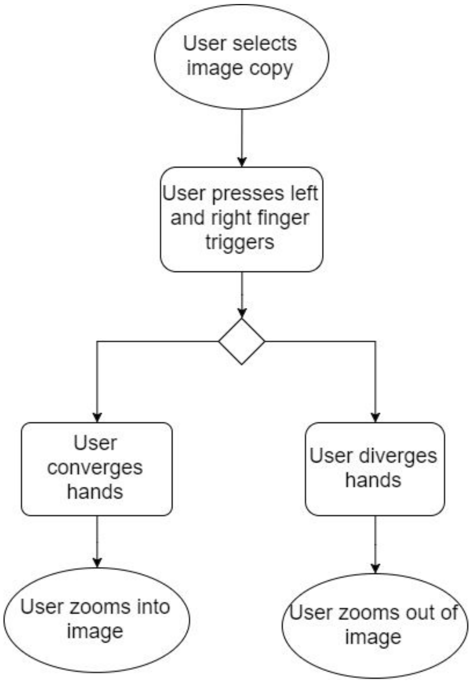
- Pre-condition:
  - Image Copy exists
- Post-condition:
  - Image Copy deleted
- Basic path:
  1. User selects image Copy with Y and B buttons on controller
  2. Image Copy is removed from workspace

5.1.7 Action: Minimize Dashboard


User in workspace
User selects minimize dashboard button
Dashboard minimized to one button

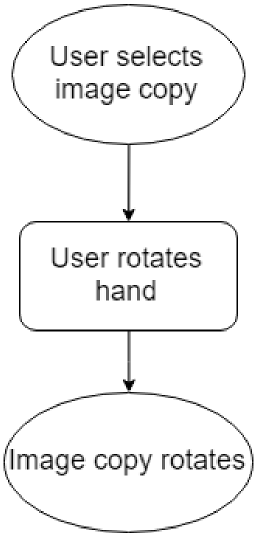
- Pre-condition:
  - Dashboard is displayed
- Post-condition:
  - Dashboard is minimized
- Basic path:
  1. User selects Minimize Dashboard button
  2. Dashboard minimizes into one button

5.1.8 Action: Zoom Copy



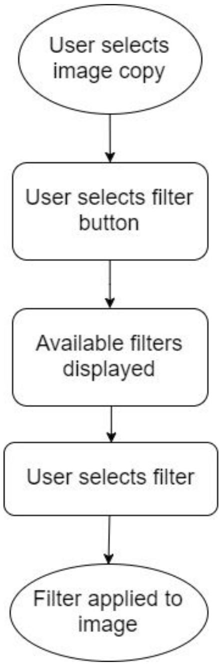
- Pre-condition:
  - Image Copy exists
- Post-condition:
  - User zooms in image Copy
- Basic path:
  1. User selects image Copy with left and right finger triggers
  2. User spreads hands to zoom in or converges hands to zoom out
  3. User zooms in or out of image

5.1.9 Action: Rotate Copy



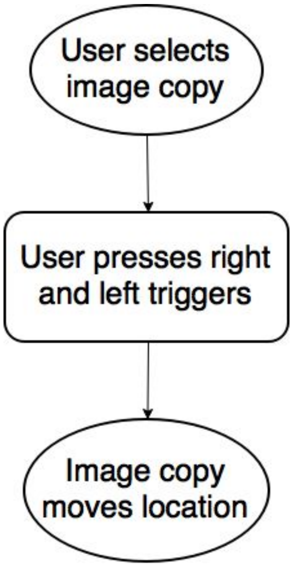
- Pre-condition:
  - Image Copy exists
- Post-condition:
  - Desired Copy is rotated
- Basic path:
  1. User selects image Copy to rotate
  2. User rotates hand
  3. Image Copy rotates

5.1.10 Action: Apply Filter to Copy



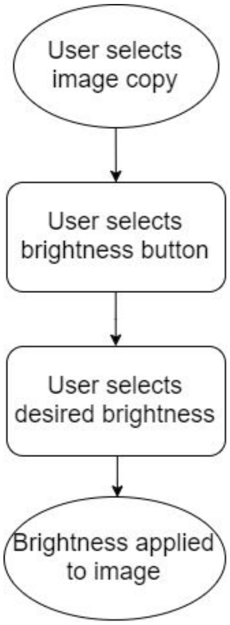
- Pre-condition:
  - Image Copy exists
- Post-condition:
  - Desired filter is applied to image Copy
- Basic path:
  1. User selects desired image Copy
  2. User selects Filter button
  3. Available filters displayed
  4. User selects desired filter
  5. Filter is applied to image Copy

5.1.11 Action: Move Copy


User selects image copy
User presses right and left triggers
Image copy moves location

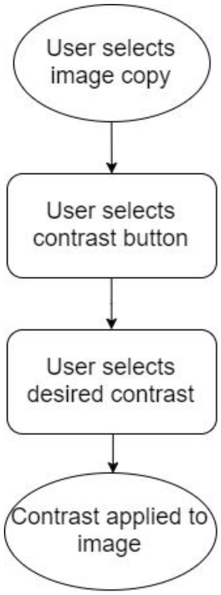
- Pre-condition:
    - Image Copy exists
- Post-condition:
    - Image Copy is moved
- Basic path:
    1. User selects desired image Copy with Left and Right hand triggers
    2. User moves hand in desired direction
    3. Image Copy moves location

5.1.12 Action: Adjust Copy brightness


User selects image copy
User selects brightness button
User selects desired brightness
Brightness applied to image

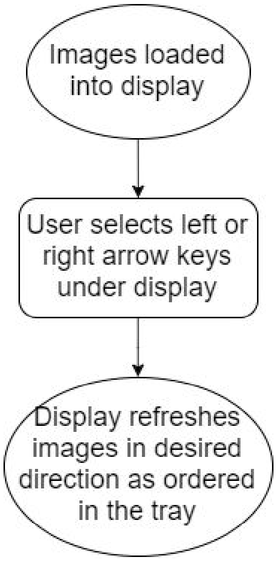
- Pre-condition:
  - Image Copy exists
- Post-condition:
  - Image Copy brightness adjusted
- Basic path:
  1. User selects desired image Copy
  2. User selects Brightness button
  3. Brightness options displayed
  4. User selects desired brightness

5.1.13 Action: Adjust Copy contrast


User selects image copy
User selects contrast button
User selects desired contrast
Contrast applied to image

- Pre-condition:
  - Image Copy exists
- Post-condition:
  - Image Copy contrast adjusted
- Basic path:
  1. User selects desired image Copy
  2. User selects Contrast button
  3. Contrast options displayed
  4. User selects desired contrast

5.1.14 Action: Browse Display images


Images loaded into display
User selects left or right arrow keys under display
Display refreshes images in desired direction as ordered in the tray

- Pre-condition:
    - Tray contains loaded images
    - Display contains current images from tray
- Post-condition:
    - Display shows desired images
- Basic path:
    1. User selects left/right arrow keys under Display
    2. Display refreshes images in desired direction as ordered in the Tray

5.2 Actor: DICOM Decoder

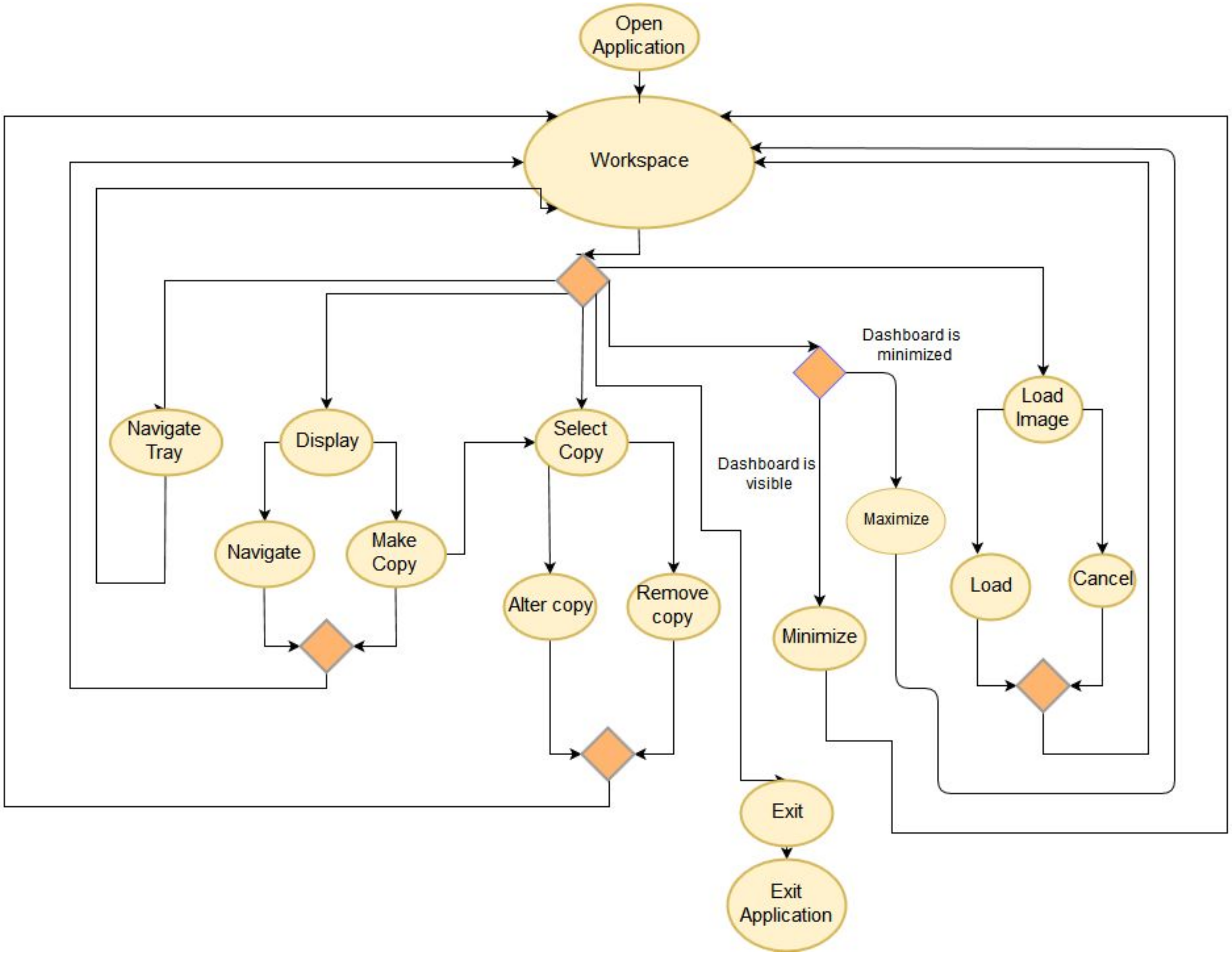
The DICOM Decoder will be an external actor to our system. Its goal will be to search for and decode DICOM files. It will take the deciphered DICOM file and send its contents to our program where they will be stored for use by the user.

5.3 Actor: DICOM File Folder

This File Folder will store DICOM files to be handled by the DICOM Decoder and then sent to our program. Currently this actor will simply be a local folder on a specific computer.

5.4 Overall System Flow

The system will operate in a simple loop, where a User taking action will always return to the base workspace, where they can inspect Copies and the Dashboard by viewing them through the VR headset. This action of viewing images is implied, and not represented here.



6 Should-Have Actions and Features

The "Should Have" section lists the functions that may be added in future versions of this project not described in the current deliverable. These are features we hope to implement, given time and budget, and additional features that will either enhance or be necessary to implement these functions.

6.1 Anti-motion sickness features

Since we are dealing with virtual reality, it is important to take into account the side effects of using this emerging technology. Other virtual reality programs have shown that nausea, dizziness, eye strain, and several other health problems can arise when using this technology. Certain individuals have been shown to be more susceptible to these health risks, but there are ways to counteract and lessen these risks. Throughout the project, we will be researching and working on ways to alleviate some of these risks.

6.2 Use of Oculus Rift Touch Controllers

Use of the Oculus Rift's Touch controllers will greatly improve the software's usability. Relying on just a mouse and keyboard places an unreasonable expectation of touch-typing ability on the user. The option to use controllers will be implemented in future deliverables of the project.

7 Could-Have Actions and Features

Due to time constraints, we may not be able to include all of the components that we would like to. These are ideas that can improve the functionality and utility of the program. The following are a few aspects that could be added to the software in future versions if time permits.

7.1 Create animation of time lapsed images

For time series images of the same object, we could potentially create a time-lapse animation for better viewing. This animation could be saved in the workspace for the user to reference.

7.2 Dictate/Write Reports within the virtual workspace

The ability to report on images while using the software would improve a radiologist's workflow. Due to the nature of holding controllers or blind use of keyboard and mouse, being able to type up reports could be difficult in practice. Dictation of reports using the Oculus Rift's microphone could be a feasible alternative to written reports. Methods of implementing this feature will be investigated if time permits.

7.3 Action: Be able to query and download files from the server

Ideally, the system should be able to access and download DICOM files from a PACS. The user would have the ability to query specific files to be viewed and manipulated by the program. Being able to interface with this system is beyond the scope of a 3-month development cycle, but the implications of such an interface will be considered in future iterations of the application's design.

7.4 Create 3D models from DICOM images to view and manipulate

Creating 3D representations of images in DICOM files would assist radiologists by aiding visualization. The user should be able to rotate, move, and zoom in on the model for better viewing. This feature will require additional research into attributes of DICOM standards, and would introduce a new set of user actions into the system.

8 CONCLUSION

This requirements document describes the features that will included in the VR medical imaging processing software. CMPT 371 Team 3will work towards developing, testing, assessing risks, and managing communication between Luxsonic and ourselves for the duration of this project. Our goal will be to provide Luxsonic with a program that incorporates the features specified above. The team will prioritize ease of use, and seek to minimize health risks to users to the best of our abilities.

9 Appendix A: Graphical User Interface

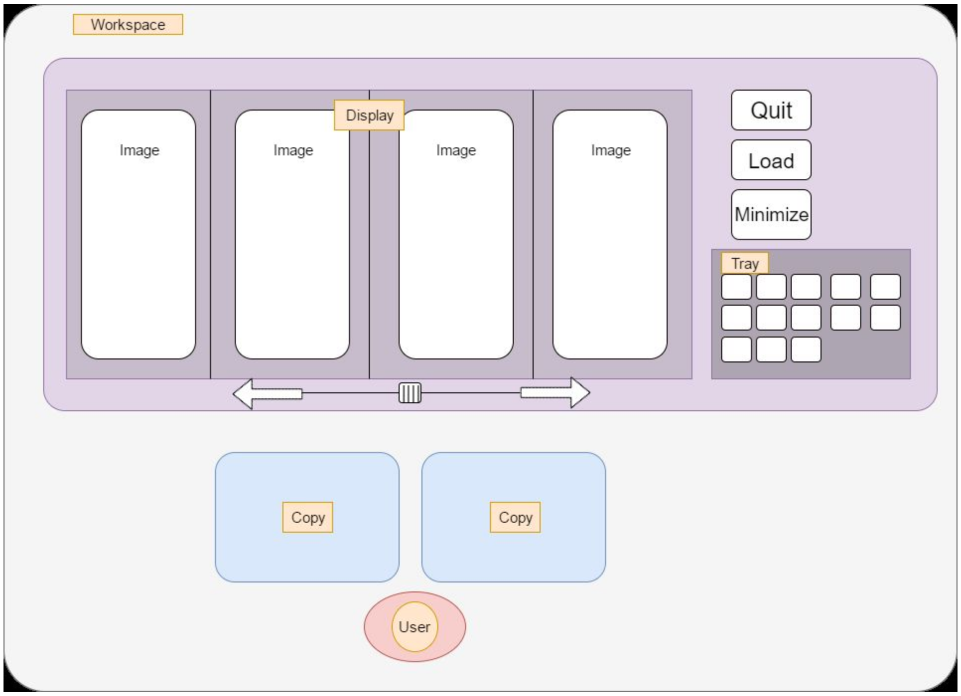


Figure 9.1

The following image represents the workspace environment from a bird's eye view

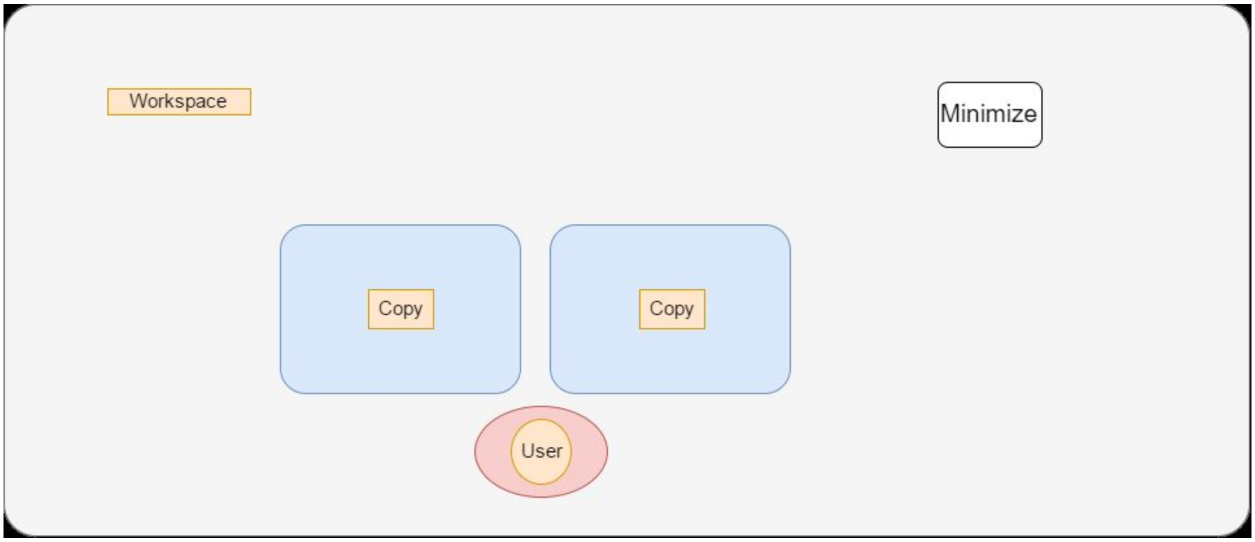


Figure 9.2

When the user selects the Minimize button, the Dashboard will collapse to a single minimize button. The user can use this feature to remove any distractions from their background when inspecting the Copies of their patient's medical images.

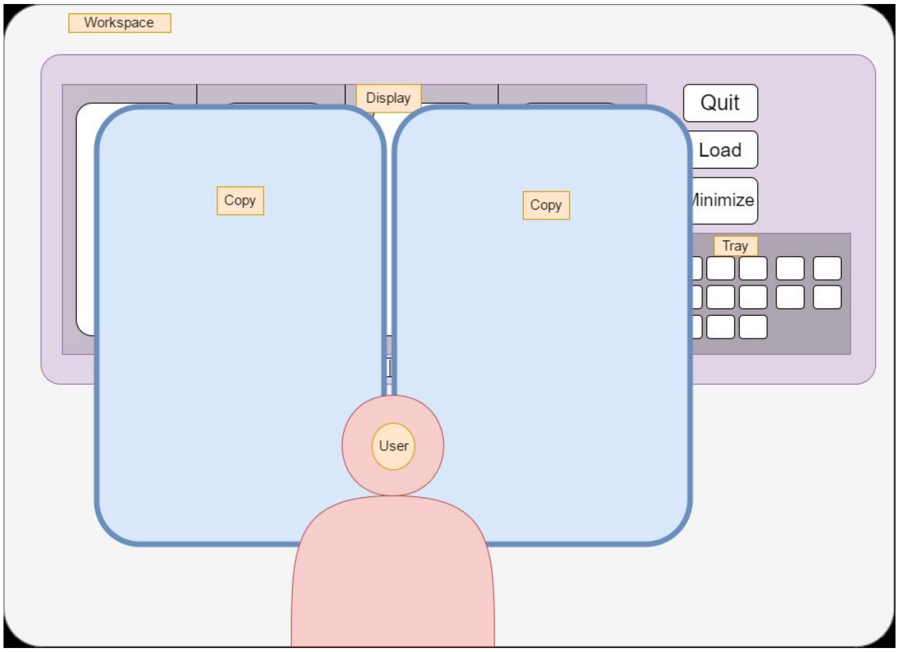


Figure 9.3

This image represents the workspace from the users perspective. The Copies will be in front of the Dashboard as they will be undergoing examination.

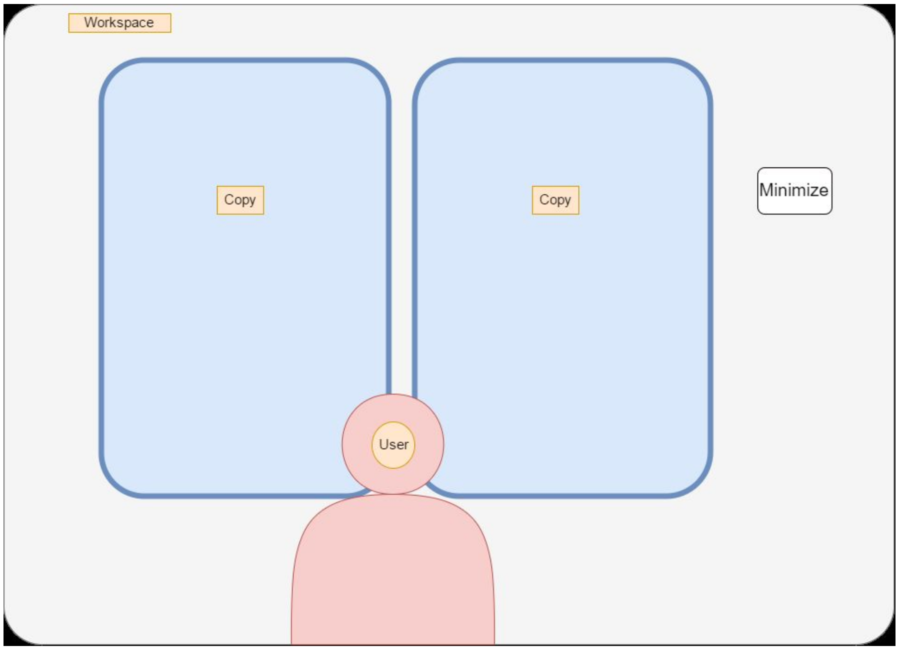
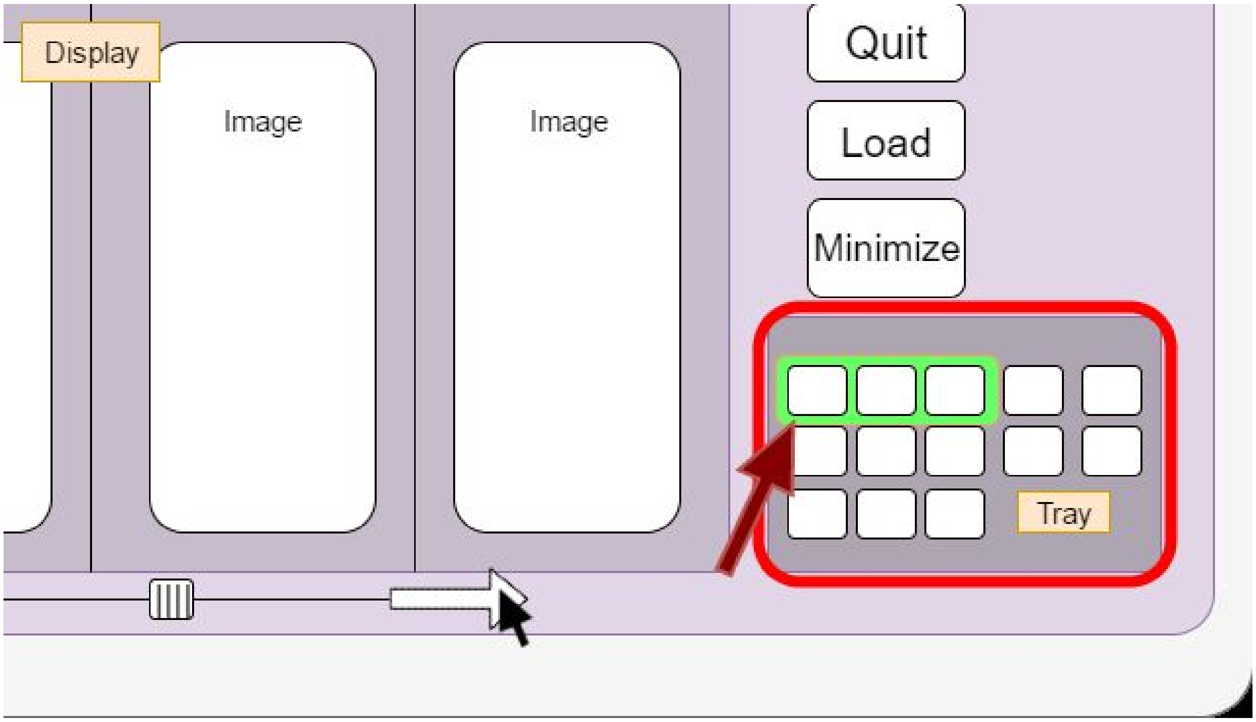


Figure 9.4

When the user selects the Minimize button, The user will be able to remove all distractions from the background and inspect the Copies selected.



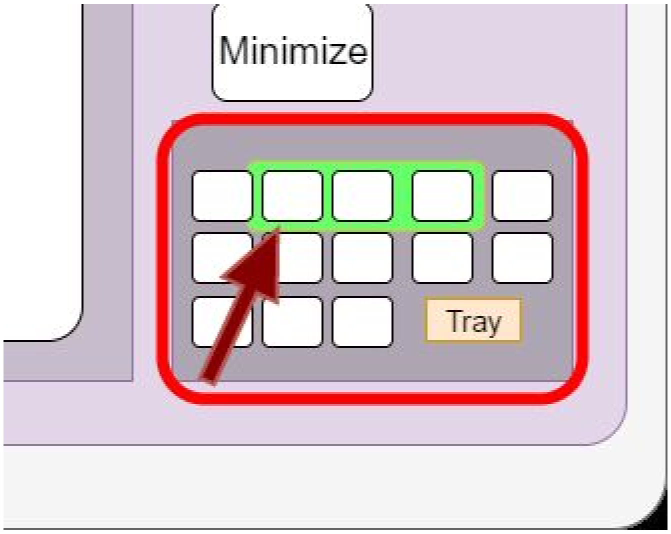


Figure 9.5 & 9.6

The images in the Display will be highlighted in the Tray. When the user navigates through images in the Display the Tray will also adjust the highlighted area to match the display. This way the user knows exactly where they are in the folder.

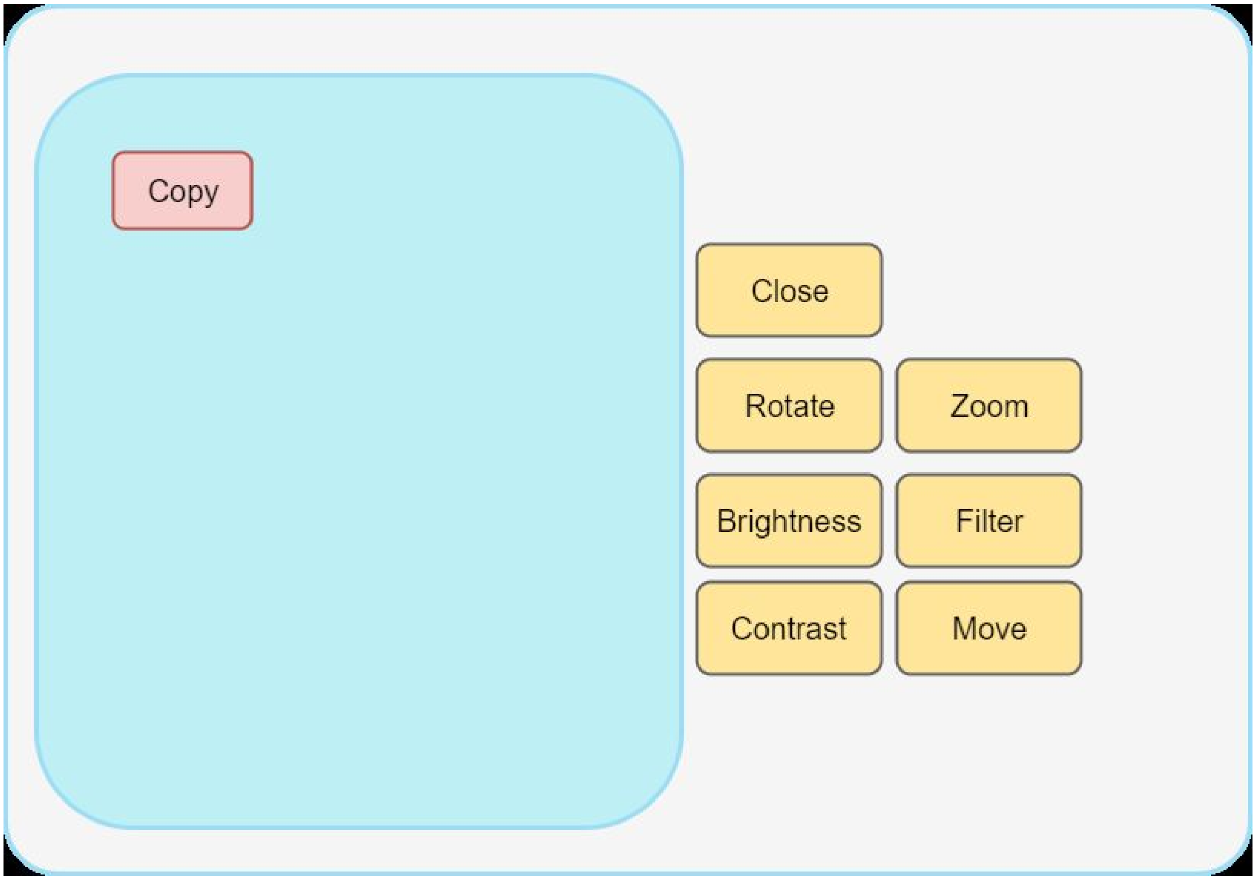


Figure 9.7

When a user selects a Copy, the following buttons will appear next to the selected Copy. The user can then use these buttons to interact with the image and adjust its visual settings.

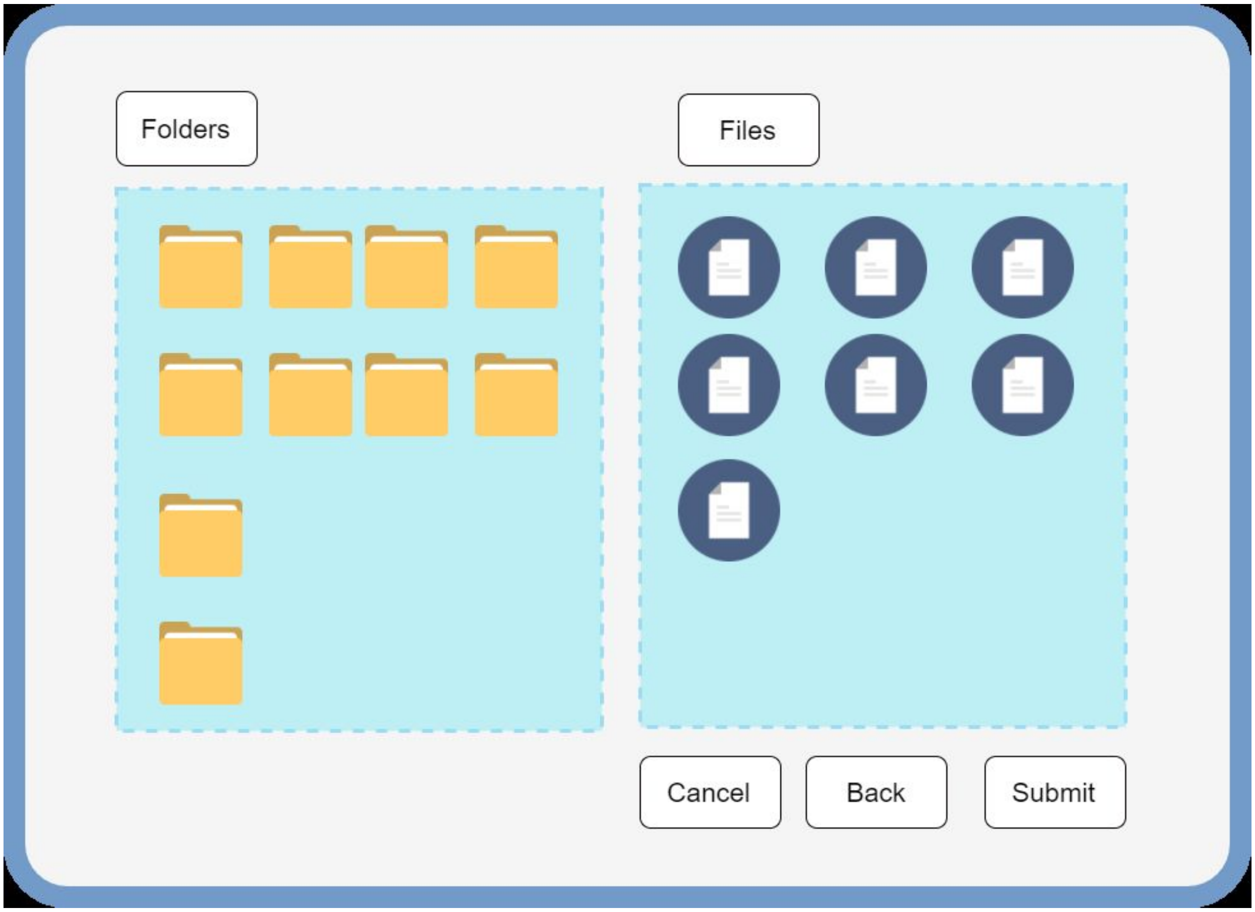


Figure 9.8

When the user selects the Load button in the workspace the following browser UI will pop up. The user can use this to navigate through their filesystems to import the files they want to inspect