

# Collaborative Segmentation Software

May 19, 2023

## Table of Contents

<b>Table of Contents.....</b>	<b>1</b>
<b>Introduction.....</b>	<b>2</b>
Project Overview.....	2
Existing Work.....	2
<b>Requirements.....</b>	<b>3</b>
<b>Specifications.....</b>	<b>3</b>
<b>Design.....</b>	<b>4</b>
Use Case Diagrams.....	4
User Diagram.....	4
Class Diagram.....	5
Tool Diagram.....	6
Back-End Diagram.....	7
Technologies.....	8
Features.....	9
Limitations.....	9
User Interface.....	10
<b>Prototypes.....</b>	<b>11</b>
Prototype 1.....	11
Prototype 2.....	12
Prototype 3.....	13
Prototype 4.....	14
Prototype 5.....	15
Prototype 6.....	16
Prototype 7 (Final Product).....	17

# Introduction

**Team:** Aidan Remington, Brandon Le, Lahasya Kakkadde Rajanna, Shyam Gnawali

**Teaching Staff:** Kunal Jain

**Client:** Lilla Zöllei

## Project Overview

A browser-based online annotation tool for 3D medical image volumes that enables annotation sharing between users. The sharing feature allows images to be viewed in multiple browsers concurrently, and can sync viewing perspectives and drawings. For a shared viewing experience, changes made to a volume/image are updated automatically when a drawing is finished or a view is changed.

[Website](#)

[GitHub Repository](#)

## Existing Work

[FreeSurfer:](#)

The tool that is currently being used at MGH.

FreeView is an annotation tool part of the FreeSurfer suite which is currently used for annotations.

[NiiVue:](#)

NiiVue is a web-based visualization framework for medical images created by Taylor Hanayik, Chris Rorden, and Christopher Drake.

It has many built-in features and canvas interactions that are helpful for creating multiple drawing functions, image saving, pen changing properties, etc.

# Requirements

The current software (FreeSurfer) does not allow collaborative labeling.

- Allow manual annotation for 3D image volumes
  - Drawing should allow coloring on sections of the image
    - Drawings with different color support: Red, Green, Blue, Yellow, Cyan, and Purple
    - Drawings will be overlayed on the 3D medical volume
      - We can change the section of the medical volume moving to the left, right, posterior, anterior, inferior, and superior
    - Can undo to reverse previous drawing(s)
  - Allow both a single user and multi-user collaborative setting
    - Users should be able to annotate at the same time or separately
    - Changes to the image will be seen by both users
    - Changes to the view will be seen by both users
  - Saving a screenshot of the canvas with image and annotations to local drive
    - A medical image slice with existing annotations can be saved as a .png image
    - Annotations on a slice without a background image can be saved as a .nii file
  - Multi-user Collaboration
    - A canvas currently displaying a medical volume with annotations can be viewed and edited with other users through a shareable URL

# Specifications

This tool is a web-based application written in JavaScript. No software is required to be pre-installed in order to use the tool.

A core feature of the application is being able to manually draw over a given 3D image volume. The NiiVue library contains functions that allow the user to configure their pen by changing the type of annotation tool and changing the color of the pen. It also allows users to undo their annotations, cycle through the slices of the medical image, zoom into the medical image, measure the medical image, save a screenshot of the canvas with the image and annotation, and save a .nii file of just the annotations.

The second core feature of the application is allowing multiple users to collaboratively annotate medical images by displaying them on each of their individual canvases. The Pusher API is used to transfer and update annotation data between users' canvases.

# Design

## Use Case Diagrams

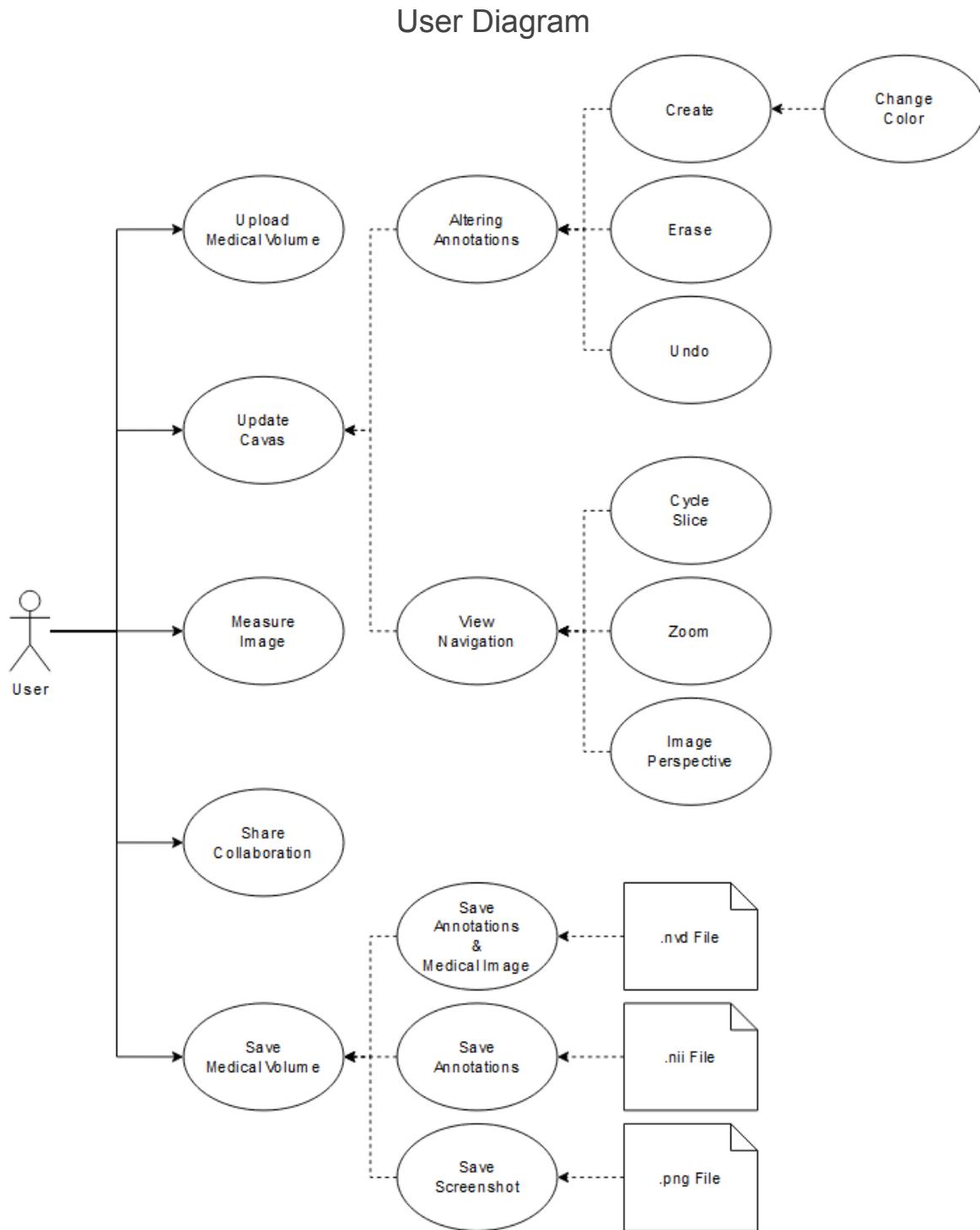


Figure 1: Overview of user actions

## Class Diagram

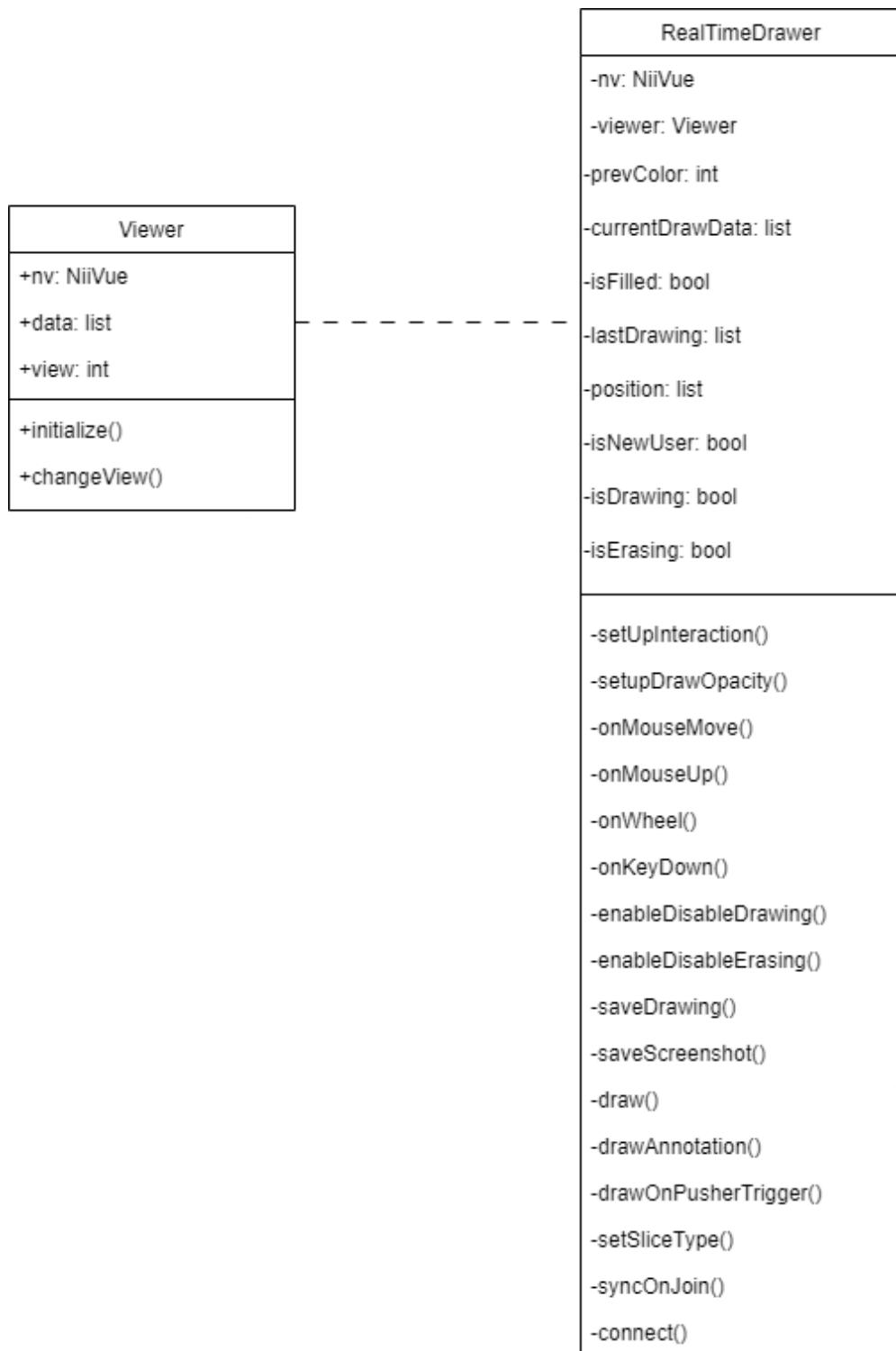


Figure 2: Overview of class attributes and methods. *RealTimeDrawer* is the main class for annotation making and sharing, while *Viewer* is a supporting class for loading the images.

Tool Diagram

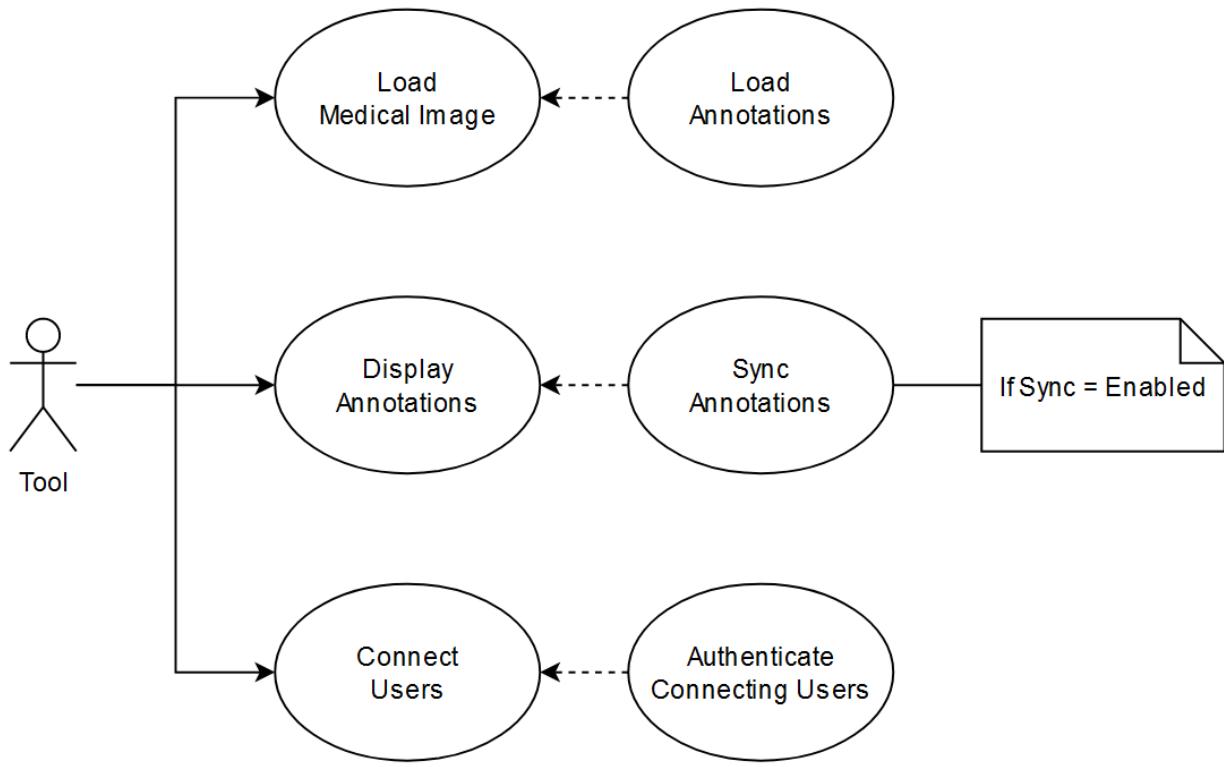
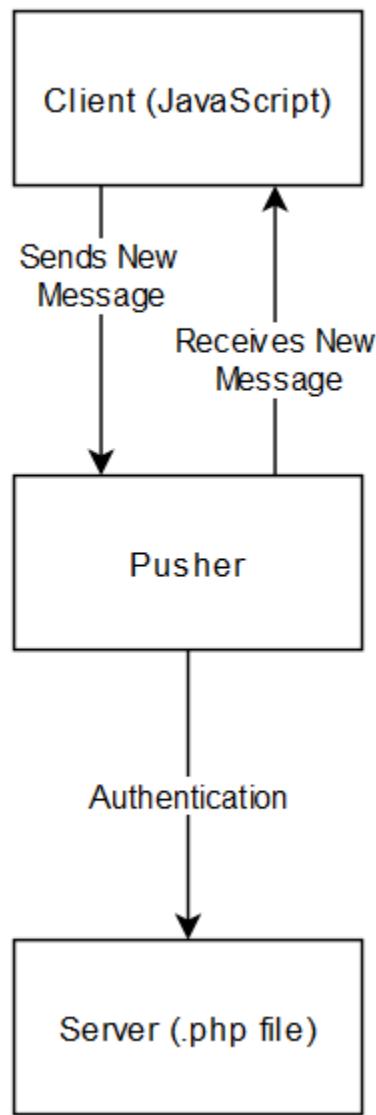


Figure 3: Overview of tool actions

The Tool Use Case Diagram displays what actions the tool does itself based on user interactions. The tool will load a medical image given by the user, and will sync the display of the annotations if there is more than one user in the collaboration room. Additionally, the tool will connect the users to the same session if they are authenticated.

### Back-End Diagram



*Figure 4: Overview of back-end interaction*

The Back-End Diagram describes how the tool enables multi-user collaboration through Pusher. A client using the tool will trigger an event by making annotations to an image in an online room shared with another client via session keys. Any such edits to an image's annotations will be sent to the server, and then Pusher will update the annotation canvas for all other clients in the room.

## Technologies

- JavaScript is the sole language as it can handle everything needed on its own
- [Pusher.com](#) API transfers messages between clients that can be used for multi-user collaboration
- [NiiVue library](#) contains methods for loading and rendering NIFTI image data in a WebGL 2.0 context.
- Supported Images:
  - Voxel-based formats: [NIFTI](#), [NRRD](#), [MRtrix MIF](#), [AFNI HEAD/BRIK](#), [MGH/MGZ](#), [ITK MHD](#), [ECAT7](#).
  - Mesh-based formats: [GIFTI](#), [ASC](#), [BrainSuite DFS](#), PLY, [BrainNet NV](#), [BrainVoyager SRF](#), [FreeSurfer](#), [MZ3](#), OFF, [Wavefront OBJ](#), [STL](#), [Legacy VTK](#), [X3D](#).
  - Mesh overlay formats: [GIFTI](#), [CIFTI-2](#), [MZ3](#), [SMP](#), STC, FreeSurfer (CURV/ANNOT)
  - Tractography formats: [TCK](#), [TRK](#), [TRX](#), VTK, AFNI .niml.tract
  - DICOM: [DICOM](#) and [DICOM Manifests](#)

## Features

- Medical Image uploading
  - Users can add an image from their local drive to a web-browser canvas by dragging the file to the URL search bar
  - Compatible [file types](#)
- Annotation of Medical Images
  - Brush mode can be toggled on and off
  - Pen colors can be selected through a table; supported colors include Red, Green, Blue, Yellow, Cyan, and Purple
  - Erasing mode can also be toggled on and off
  - Undo button removes the previous annotation made on the canvas
- Viewing of Medical Images
  - Can cycle through slices a medical volume currently being viewed in the canvas
  - The perspective from which a medical volume is being viewed can be changed with a keystroke
  - Magnification zooming on an image is controllable with scrolling
  - Distance measurements across an image can be turned on, and are scaled according to Imperial millimeter units
- Synchronization
  - Multiple users see annotations and image view changes concurrently
  - Sync can be toggled on and off
- File Saving
  - .nvd file saves the medical image with annotations
    - Can be uploaded again to continue annotating
  - .nii file saves the annotations
  - .png file saves the current image slice with annotations

## Limitations

- [Pusher Rate Limits](#)
  - Only 10 requests allowed per second
  - Size limit of 10 KB
  - Allowing higher rate of requests per second will improve syncing
- Sync
  - Only one channel open which means different images will have same annotations
  - Syncing is not perfect, there is an offset at times
    - Refreshing the page is a simple solution to solve the offset
  - Erased annotations sync slow
  - Annotations created with view sync off will be removed when view sync is turned on
- Pen width cannot be changed

## User Interface

A hot-key menu is displayed on the bottom left of the screen with three clickable settings: 'Select Color', 'Opacity', and 'Sync View'. A canvas is displayed with the 3D medical volume loaded at the center of the screen. The sync view icon lights up when syncing is on, and is gray when syncing is turned off. There is also a sliding bar for selecting the opacity level of the pen.

Please refer to the project's [GitHub repository](#) which lists the controls as well as the description of each setting.

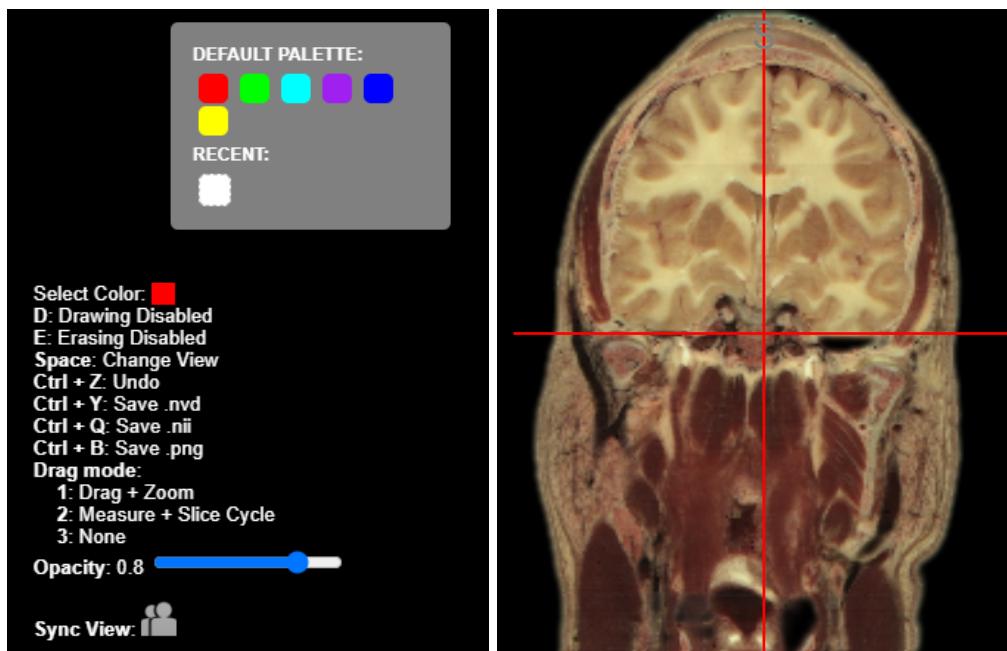


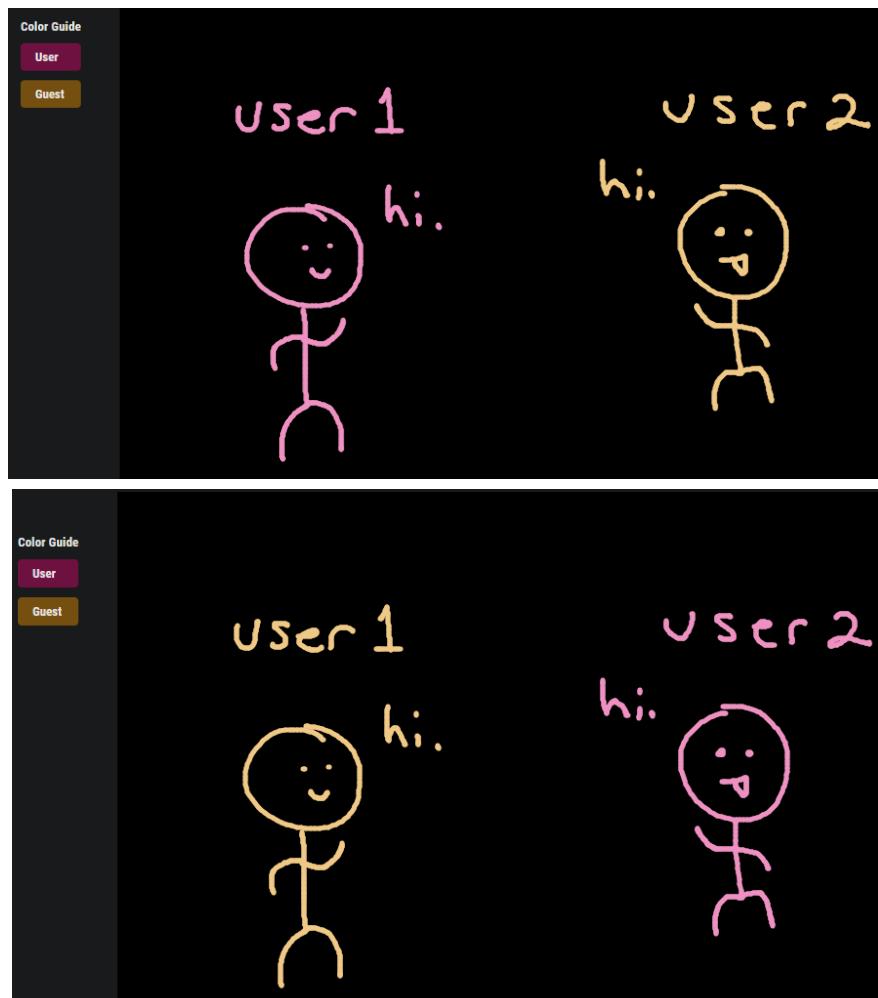
Figure 5: Hot-key menu and canvas display

# Prototypes

## Prototype 1

We developed the first prototype by using this [sample](#) from Pusher's website. It was a mock-up to show how the annotation feature would work. The sample also included a collaborative component for two people to draw and display what the other person was drawing.

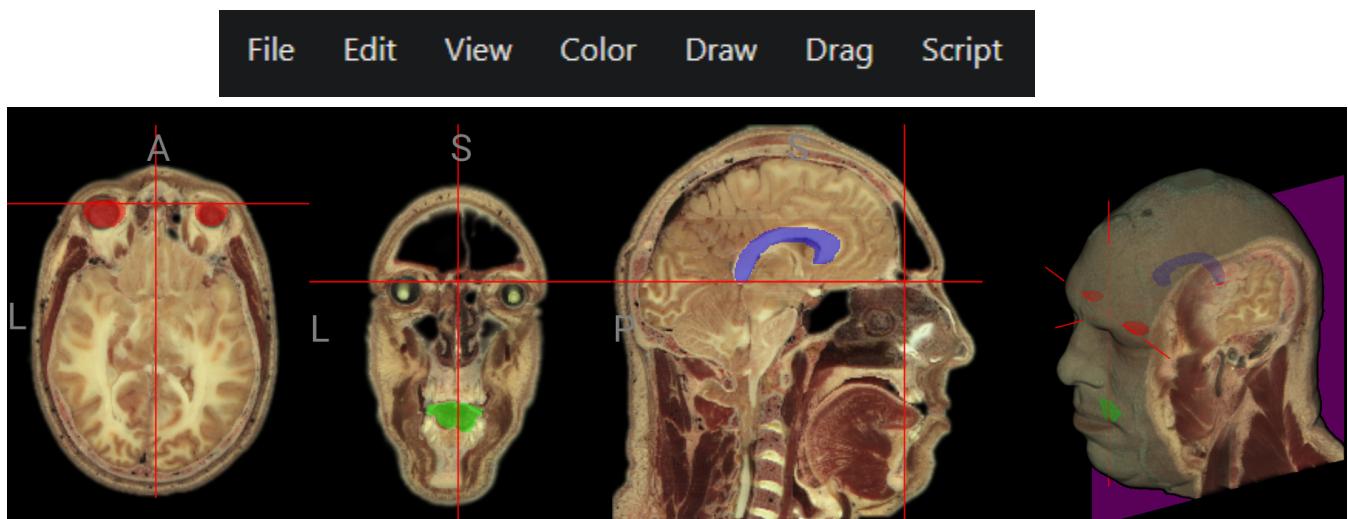
- JavaScript
- Express
- Node.js
- Pusher
- React



## Prototype 2

The second prototype we developed by using a [sample](#) from NiiVue's demo page. We used the sample to conduct tests on various 3D medical volumes, evaluating their interactions with NiiVue's library. Through this prototype, we were able to assess NiiVue's capabilities and plan how we can incorporate them into future development stages.

- JavaScript
- NiiVue
- React



## Prototype 3

The third prototype we developed to showcase our proposed collaboration system. The website's main page would provide users with the option to create or join a room, and upon creating a room, a unique hash key would be generated that users could copy and share with others to allow access to join their room.

As part of this prototype, we also included a basic canvas feature that allowed users to draw for us to test the collaborative annotation capabilities of the system.

- JavaScript
- Express
- React
- Socket.io

### Create Room

User1

35cd5448-1ea4-416a-22da-1

[Generate](#)

[Copy](#)

[Create Room](#)

### Join Room

User2

35cd5448-1ea4-416a-22da-1727297f8eae

[Join Room](#)

User1 - (You)

h e l l o

User2

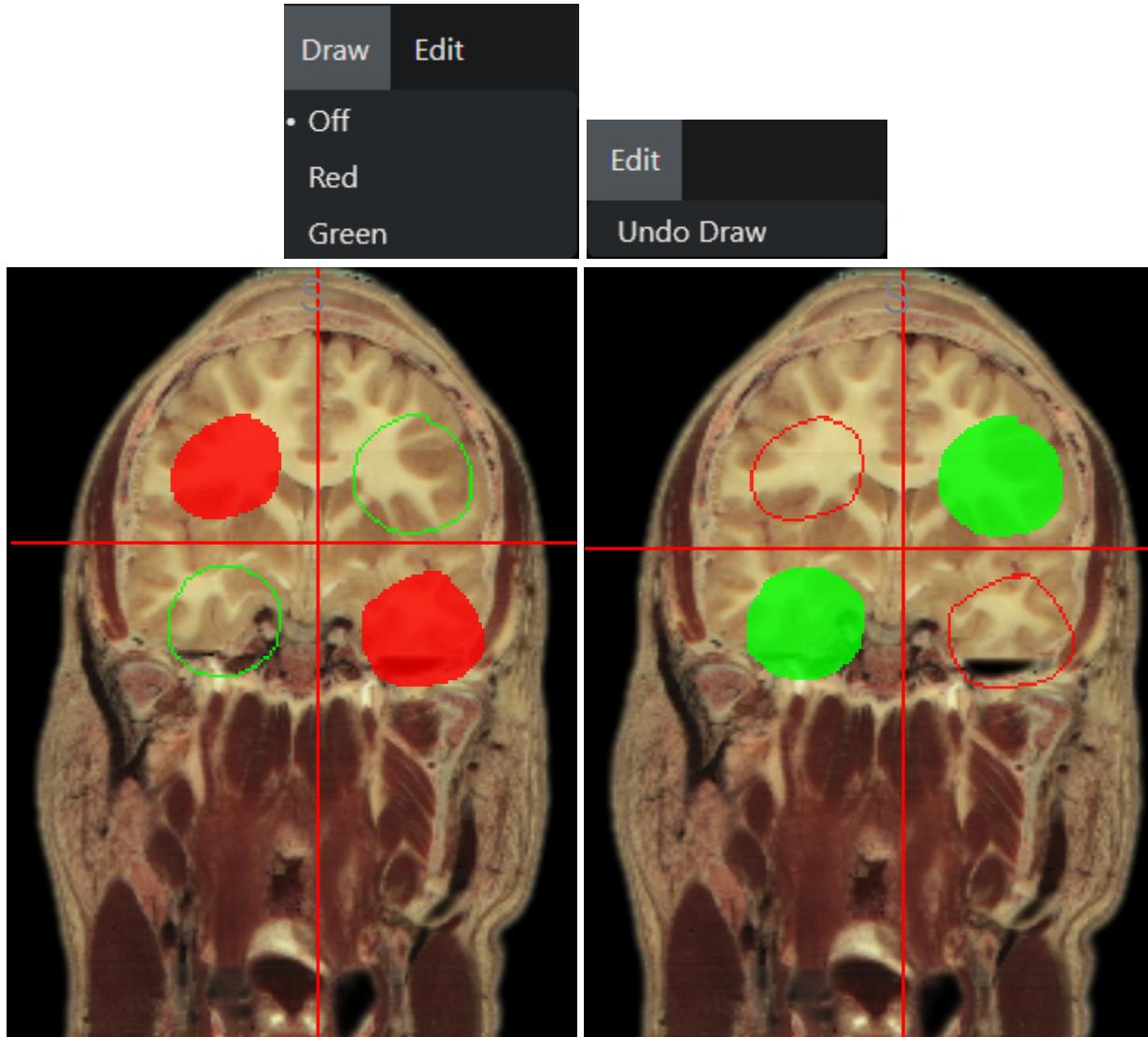
h e l l o

## Prototype 4

The fourth prototype was an integration of the first and second prototypes. It utilizes NiiVue's library to access their drawing and undo features for creating and removing annotations. To enable collaboration among users, we integrated Pusher's API to sync the annotations across all medical images.

During the testing phase, we encountered an issue with the annotation fill points generated by NiiVue, as they were not synced with Pusher's API. This caused other users to see only the outline of the drawing. Unfortunately, the fill point data generated internally by NiiVue was too large to send to other users, which resulted in system performance issues.

- JavaScript
- Pusher
- NiiVue

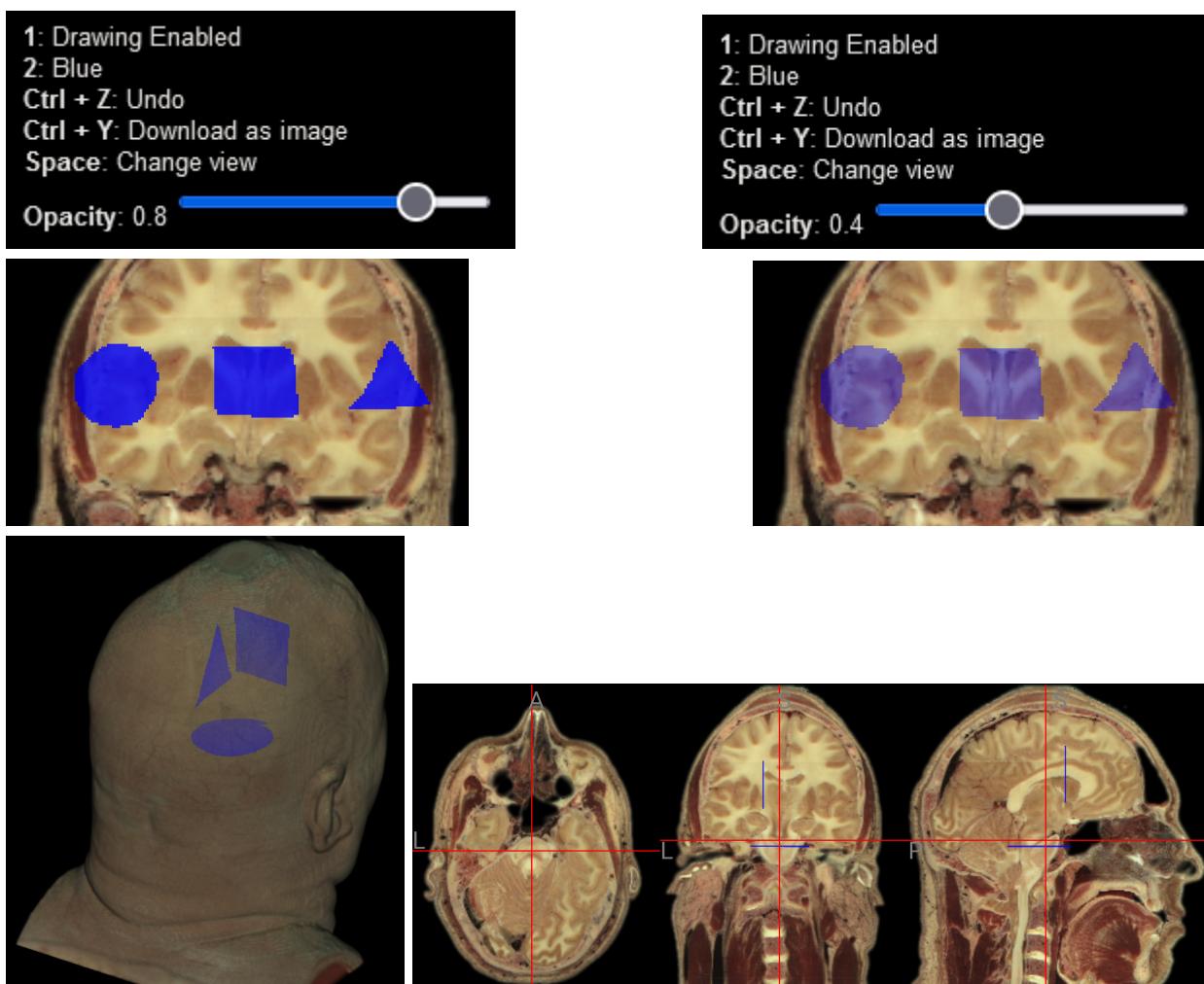


## Prototype 5

The fifth prototype was a branch of the fourth prototype. We eliminated the need to click through the user interface to modify settings, instead opting for hotkeys for a more streamlined user experience. To further declutter the interface, we removed the toolbar from the top of the screen and placed it into a condensed UI at the bottom left corner, which displays the controls for the keyboard.

Building on the prototype's features, we integrated additional functionalities using NiVue's library. These included a drawing toggle, pen color cycle, an ability to save a screenshot of the current image slice with annotations as a .png file directly to the user's local drive, an ability to cycle through different perspectives of the 3D medical image, and the ability to adjust the opacity of annotations for improved visual clarity.

- JavaScript
- Pusher
- NiVue



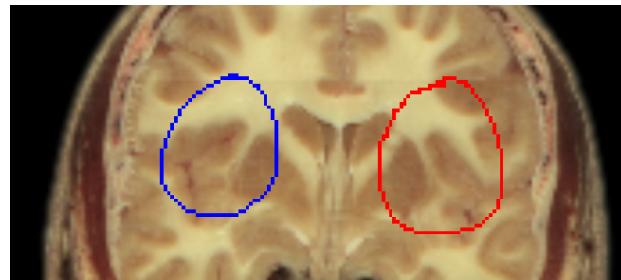
## Prototype 6

The sixth prototype was a further development from the fourth prototype with functionalities from the fifth prototype being incorporated later on. As with the fifth prototype, we focused on simplifying the UI, removing the toolbar and relying on hotkeys for easier access.

This prototype was designed to address issues related to user experience. One major improvement was the addition of a pen color palette, which was kept as a UI element instead of relying on hotkeys, as the latter could be cumbersome with multiple colors to cycle through. To streamline the annotation process, we also removed fill points, offering users a simple brush tool that is more intuitive and commonly used.

In addition, the sixth prototype also introduced a new feature that allowed users to upload medical images directly to the canvas using a data parameter in the website's URL. This was particularly useful in cases where the user did not have the file stored locally on their drive, enabling quick and easy access to the image for annotation.

- JavaScript
- Pusher
- NiiVUe



Example Links:

- <https://oopsuwu.github.io/Collaborative-Segmentation-Software/?data=https://oopsuwu.github.io/Collaborative-Segmentation-Software//samples/FLAIR.nii.gz>
- <https://oopsuwu.github.io/Collaborative-Segmentation-Software/?data=https://niivue.github.io/niivue-demo-images/visiblehuman.nii.gz>

## Prototype 7 (Final Product)

The seventh prototype is the final production prototype currently. It integrates the best features from both the fifth and sixth prototypes regarding the annotation tool and canvas.

Among the many improvements added to this version, the sync feature is now controllable allowing users to work independently. An eraser tool was also created to correct small annotation errors. Image slice and perspective sync was also added as a developing feature.

Additionally, using the right mouse button with the mouse wheel now offers different modes, such as dragging + zooming and measurement + slice cycle, adding more versatility to the tool. We've also added the capability to save the medical image with the annotations created as a .nvd file, this allows the user to create additional annotations later on further expanding the range of possibilities for users.

The sync toggle does not allow separation between the view sync and the annotation sync.

- JavaScript
- Pusher
- NiiVUE

