



**F3**

**Faculty of Electrical Engineering  
Department of Computer Graphics and Interaction**

**Master's Thesis**

# **UX design of a medical imaging data annotation tool for AI applications**

**Miroslav Müller**

**Open Informatics - Human-Computer Interaction**

**September 2022, May 2024**

**Supervisor: Doc. Ing. Wanda Benešová PhD.**



## Acknowledgement / Declaration

I want to thank my advisor, Doc. Ing. Wanda Benešová PhD., for her professional guidance, frequent consultations she offered to me and helping me stay positively motivated on the work. I want to thank my loved partner Zuzana, my parents and friends, who were always supporting me and believed, that I can successfully finish the work and graduate, becoming an engineer.

I declare that the work presented here is my own and that I have listed all the literature and information sources used in accordance with methodological guidelines on compliance with ethical principles in the preparation of university theses.

In Prague, 20 January 2023

.....

## Abstrakt / Abstract

Vývoj aplikací umělé inteligence (AI) v radiologii, konkrétně pro segmentaci 3D snímků z MRI/CT skenů, vyžaduje přesně anotované datasety. Úkol anotace radiologických dat obvykle připadá na lékařské specialisty, jejichž odborné znalosti jsou nenahraditelné a jejich práce je nákladná. Suboptimální použitelnost a neefektivnost stávajících nástrojů pro anotaci, zejména pro 3D obrazová data, tento problém dále zhoršuje, což zdůrazňuje potřebu zlepšení v této oblasti.

Tato diplomová práce se zabývá těmito výzvami a představuje UX návrh uživatelsky orientovaného nástroje pro anotaci lékařských obrazových dat pro vývoj AI aplikací. Cílem je snížit kognitivní zátěž uživatelů během procesu anotace a tím učinit pracovní postup anotace co nejlépe a nejefektivnější.

Pro dosažení tohoto cíle jsem nejprve provedl komplexní průzkum existujících přístupů a nástrojů používaných pro anotaci obrazů v radiologii, přičemž jsem identifikoval klíčové oblasti vyžadující zlepšení. Následně jsem přijal přístup zaměřený na uživatele (User-Centered Design, UCD) pro vývoj prototypu nástroje pro anotaci. Prototyp byl navržen tak, aby splňoval požadavky vyplývající z důkladného výzkumu uživatelů a byl podroben hodnocení prováděním testů použitelnosti.

Tato práce tedy tvoří robustní základ pro vývoj nástroje, který má za cíl zvýšit celkovou efektivitu a snížit kognitivní zátěž procesu anotace lékařských snímků, a tím urychlit vývoj AI aplikací v radiologii a příbuzných oblastech.

**Klíčová slova:** 3D lékařské zobrazování; segmentace obrazu; anotace datasetů; human-in-the-loop; interaktivní strojové učení; vývoj AI aplikací

**Překlad titulu:** Anotační nástroj (UX návrh anotačního nástroje lékařských obrazových dat pro aplikace AI)

The development of Artificial Intelligence (AI) applications in radiology, specifically for segmentation of 3D images from MRI/CT scans, necessitates precisely annotated datasets. The task of annotating radiological data typically falls to specialist doctors, whose expertise is both invaluable and costly. Moreover, the suboptimal usability and inefficiency of existing annotation tools especially for 3D image data further worsen the issue, highlighting the need for improvements in this domain.

This thesis addresses these challenges by proposing the UX design for a user-oriented medical imaging data annotation tool for AI application development. The goal is to lift the cognitive burden from users during the annotation process, thus making the annotation workflow as smooth and time-efficient as possible.

To achieve such goal, I have first conducted a comprehensive survey of existing approaches and tools used for image annotation in radiology, identifying key areas for improvement. Following this, I adopted a User-Centered Design (UCD) approach for the development of a prototype annotation tool. The prototype was designed to fulfill requirements arising from thorough user research and was subjected to evaluation by conducting usability tests.

This work therefore establishes a robust foundation for the development of a tool ultimately enhancing the overall efficiency and reducing the cognitive load of the medical imaging annotation process, and therefore accelerating the development of AI applications in radiology and related domains.

**Keywords:** 3D medical imaging; image segmentation; dataset annotation; human-in-the-loop; interactive machine learning; AI application development



# Contents /

<b>1 Background</b>	<b>1</b>		
1.1 Image segmentation in medical domain	1		
1.2 Utilizing machine learning techniques	2		
1.3 Motivation	3		
1.4 Goals	3		
<b>2 Foundations of imaging data</b>	<b>4</b>		
2.1 Representing image data	4		
2.2 Viewing image data	5		
2.3 Storing image data	6		
2.3.1 DICOM	6		
2.3.2 NIfTI	6		
<b>3 Related Work and Software</b>	<b>7</b>		
3.1 Open-source software	7		
3.1.1 3D Slicer	7		
3.1.2 OHIF	8		
3.1.3 MONAI Label	8		
3.1.4 Ilastik	8		
3.1.5 Digital Slide Archive and HistomicsUI	10		
3.1.6 QuPath	10		
3.1.7 Mango	11		
3.1.8 ITK — Snap	11		
3.1.9 MITK	11		
3.2 Software Used in Medical Practice	11		
3.2.1 Marie PACS	11		
3.2.2 Philips Intellispace Portal	12		
3.2.3 Siemens Syngo.via RT Image Suite	12		
3.2.4 Varian Eclipse	13		
3.3 Other Commercial Software	14		
3.3.1 RadiAnt	14		
3.3.2 V7 Darwin	14		
<b>4 Methodology</b>	<b>15</b>		
4.1 Utilized HCI and UCD Principles	15		
4.1.1 User Experience and Usability	15		
4.2 Utilized User Research Principles	16		
4.3 Utilized User Research Methods	17		
4.3.1 Qualitative Interviews	17		
4.3.2 Field Research	17		
4.3.3 Interview Evaluation	17		
4.4 User Personas	18		
4.5 User Scenarios	18		
4.5.1 Scenarios in this work	18		
4.6 Hierarchical Task Analysis	19		
4.7 Functional Design	19		
4.8 Prototyping	19		
4.8.1 Used Technologies	20		
4.8.2 Visual Design	21		
4.9 Usability Testing	21		
4.9.1 Core Principles	21		
4.9.2 Actionable Insights	22		
4.10 HCI in AI Image Segmentation	22		
4.10.1 Human-in-the-loop machine learning	22		
4.10.2 Segmentation Specific Methods	23		
4.10.3 Applying HITL Techniques in This Work	23		
<b>5 Results</b>	<b>24</b>		
5.1 Initial User Research	24		
5.1.1 Determining Target Users	24		
5.1.2 Qualitative Interviews	24		
5.1.3 Clarification of Usage by Target Users	26		
5.2 Related Software Exploration and Testing	27		
5.2.1 Comparison Table Idea	27		
5.2.2 3D Slicer and MONAI label	27		
5.2.3 OHIF	29		
5.2.4 Volume label	30		
5.2.5 Mango	31		
5.2.6 RadiAnt	32		
5.2.7 V7 Darwin	33		
5.2.8 Hypotheses and Questions	35		
5.2.9 Quick Mouse Control User Test	36		
5.3 Definition of Personas, Scenarios, and Tasks	36		
5.3.1 User Personas	36		
5.3.2 User Scenarios	36		
5.3.3 Task Analysis	37		
5.4 Prototype Development	37		

5.4.1	Early Design . . . . .	37	A.4.2	Scenario 2 — Radiologist Non-Expert . . . . .	64
5.4.2	An Attempt on Axure RP . . . . .	40	A.4.3	Scenario 3 — Amateur Annotator . . . . .	64
5.4.3	Return to Figma . . . . .	41	A.5	Ideal Application Pass . . . . .	65
5.5	Usability Testing . . . . .	46	<b>B Prototype Development</b>	<b>67</b>	
5.5.1	Usability Test Plan . . . . .	46	B.1	Early Design Specification 1 . . . . .	67
5.5.2	First Usability Test . . . . .	46	B.2	Early Design Specification 2 . . . . .	67
5.6	Prototype Improvement . . . . .	47	B.2.1	Use cases . . . . .	67
5.6.1	Improvements Based on Test Results . . . . .	47	B.3	FigJam drafts . . . . .	67
5.6.2	Export from Figma . . . . .	48	B.3.1	Prototype LoFi 1.0 . . . . .	67
<b>6 Discussion</b>	<b>50</b>		B.3.2	Prototype LoFi 4.0 . . . . .	68
6.1	Literature and Software Research . . . . .	50	B.3.3	Prototype 1.1 . . . . .	68
6.2	Qualitative Interviews . . . . .	50	B.4	Paper Sketches . . . . .	68
6.2.1	Validity . . . . .	50	B.5	Annotation Interaction Methods . . . . .	69
6.2.2	Reliability . . . . .	50	B.5.1	Common Interaction Variations . . . . .	69
6.2.3	Systematic Error . . . . .	51	B.5.2	Interactions . . . . .	70
6.3	Usability Testing . . . . .	51	B.5.3	Possible UI/UX features . . . . .	72
6.3.1	Introducing Quantifiable Metrics . . . . .	51	B.6	Axure RP prototype . . . . .	72
6.4	Further Prototype Development . . . . .	51	B.7	Design Specification 3 . . . . .	73
6.4.1	Visual Design . . . . .	51	B.7.1	Functional Specification . . . . .	73
6.4.2	Suggested Features . . . . .	51	B.7.2	Annotation Flow Diagram . . . . .	75
6.4.3	Guidelines for Human-AI interaction design . . . . .	52	<b>C Prototype Before Testing</b>	<b>76</b>	
6.4.4	Utilizing Interactive Segmentation . . . . .	52	<b>D Usability Test v1.0</b>	<b>80</b>	
6.5	Conclusion . . . . .	52	D.0.1	Test Properties . . . . .	80
<b>References</b>	<b>53</b>		D.0.2	Goals . . . . .	80
<b>A User Research</b>	<b>57</b>		D.0.3	Testing Environment . . . . .	80
A.1	Qualitative Interviews . . . . .	57	D.0.4	Metrics . . . . .	80
A.1.1	Interview Design . . . . .	57	D.0.5	Notes . . . . .	80
A.1.2	Radiology Interview Results . . . . .	58	D.0.6	Test Scenario . . . . .	81
A.1.3	Oncology Interview Results . . . . .	60	D.0.7	Test notes and findings . . . . .	82
A.2	User Personas . . . . .	61	D.0.8	Usability Test v1.0 - Findings Summary . . . . .	82
A.2.1	Defining core needs of personas . . . . .	61	<b>E Prototype After Testing</b>	<b>86</b>	
A.3	3D Slicer Usability Test . . . . .	62			
A.3.1	Collected Insights . . . . .	62			
A.4	User Scenarios . . . . .	63			
A.4.1	Scenario 1 — Radiologist . . . . .	63			

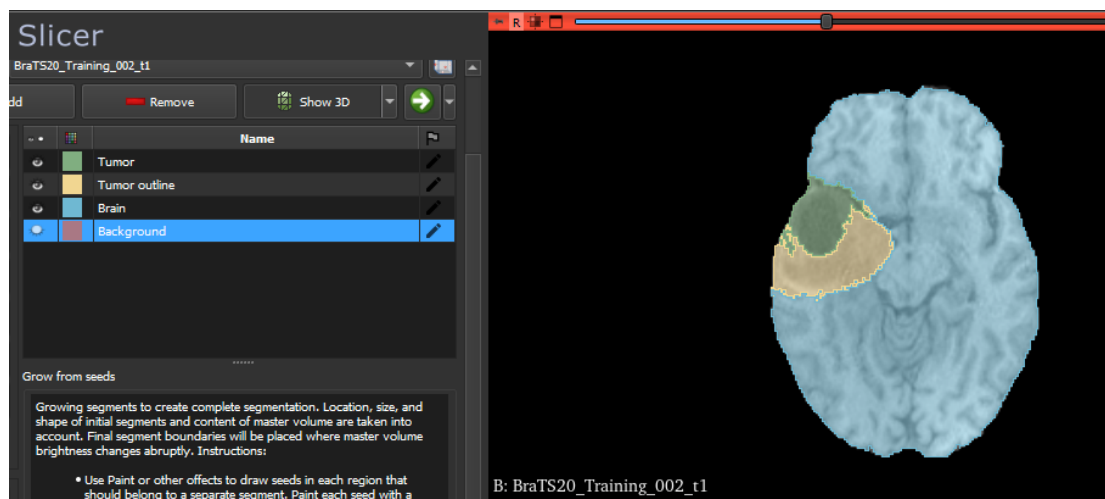
# Chapter 1

## Background

Over the past decade, the field of machine learning, particularly deep learning, has seen remarkable advances, leading to a variety of successful practical applications. In this work, I dive into one of such successful applications in the domain of medical imaging — image segmentation.

### 1.1 Image segmentation in medical domain

Segmentation of images, also known as contouring or annotation in specific contexts, is a procedure with the goal of delineating individual regions of interest (ROI) in an image. In the medical domain, such regions typically correspond to organs, lesions, tumors, and various other anatomical structures [1]. It is an important procedure for analysis and computer-aided diagnosis. Popular medical image segmentation tasks mentioned by Minaee et al. [2] include liver and liver-tumor segmentation, brain and brain-tumor segmentation, optic disc segmentation, cell segmentation, lung segmentation, pulmonary nodules, and cardiac image segmentation. Generally speaking, segmentation is needed for tasks such as visualization of body structures, quantification (measuring volume, surface, shape properties), radiotherapy planning, 3D printing, masking (restricting processing or analysis to a specific region), and more [3].



**Figure 1.1.** An example segmentation of an MRI brain image in 3D Slicer software.

In this work, I use the term “segmentation” when discussing the ROI delineating procedure and its result in general. When the focus is on specialists using a software tool to create segmentations in medical practice, I adopt the term “contouring”, which is broadly used in this domain. Finally, when hired annotators use software tools to create either partial or full segmentations with the purpose of using them as training data for machine learning models, I find “annotation” to be the most suitable term.

The term “medical imaging” then describes the process of capturing images of the interior of a body for clinical analysis [4]. The process utilizes methods such as tomography, ultrasound, magnetic resonance imaging (MRI), and computed tomography (CT) to produce the images. Particularly, MRI and CT are of special interest in the case of this work. The resulting images are a 3D volume, usually represented by slices of 2D black and white images.

The 3D nature of such medical imaging data poses a challenge for efficient segmentation process. As mentioned in relevant works [5] [6] and confirmed during conducted interviews, manual image segmentation of 3D data is a severely time-consuming task, since a significant amount of time is often spent on each 2D slice of the image. Manual segmentation can also lack consistency due to the differences in use of manual annotation tools between annotators.

## 1.2 Utilizing machine learning techniques

Opportunities for automation and the use of machine learning techniques for speeding up the process of 3D medical data segmentation therefore arise. Diaz-pinto et al. [5] claim that deep learning algorithms are the current state of the art for automatic 2D and 3D medical image segmentation. Particularly mentioned are convolutional networks. Nevertheless, in case of supervised learning, the success of training a deep learning model capable of automatic segmentation relies heavily on the availability of an adequate amount of data that has already been annotated. Data that are of high-quality must be created using knowledge and time of high-qualified medical experts. And as mentioned in the previous paragraph, the price for acquisition of such data is often high. As a result, the overall cost for applying supervised machine learning techniques significantly increases. The lack of expertly annotated data in the medical domain and the need for a more efficient creation of datasets was behind the works such as MONAI Label [5] and the creation of additional datasets such as NuCLS [6] or BRATS [7].

A potential solution to the training dataset availability problem could be using an unsupervised learning technique, although that is rather out of the scope of this paper. Nevertheless, there is also the option for using semi-supervised and weakly supervised machine learning techniques, which could both benefit from this work. According to Wang et al. [8], weakly supervised learning encompasses data augmentation, transfer learning and interactive segmentation. Of these methods, the lattermost — interactive segmentation — is also of particular interest in this work. The concept allows annotators to interactively correct the results from an image segmentation model. This is done in a loop by feeding the model with corrections, updating its parameters, and obtaining more accurate segmentation until desired precision is achieved. Finally, semi-supervised learning combines unsupervised and supervised learning together, allowing for having less annotated data.

Apart from training issues, another concern with supervised machine learning used for medical image segmentation might arise once the model is used in inference mode by end users in practice. The users get automated results, where it might be difficult to make further changes to the resulting segmentation. Building on the concept of interactive segmentation, a promising approach to enhance both training and deployment of AI assisted segmentation is to combine immediate predictions from AI models with human-in-the-loop machine learning practices. This way, the user can effectively use AI predictions while allowed to put incentives for the model to further improve the

predictions. Furthermore, by providing the user with well-designed interactions, we can achieve segmentation in a short time using minimal effort. It is therefore essential to create segmentation tools with both automation and usability in mind.

## 1.3 Motivation

As mentioned in previous sections, deep learning methods are significantly advancing the fields of medical imaging in recent years. Researchers are proposing new approaches to medical image segmentation and many are developing specific AI segmentation applications focusing on their domain of expertise [9]. An efficient way for training new task-specific deep learning models is therefore necessary. This work aims to tackle the problem by focusing on improving the efficiency of dataset annotation.

## 1.4 Goals

My work covers a major part of the design phase of an open-source annotation tool. The tool is designated for efficient annotation of medical and biomedical images and the development of AI applications in these domains. I will be referring to the presented tool as “the EFficient ANnotation (EFAN) tool” further in this work. This is a provisional name and is subject to change in a later phase.

In its initial phase, the primary purpose of the EFAN tool is to allow researchers, medics, and annotators to effectively view and annotate medical images. Such fast annotations are meant to be used for training datasets, therefore speeding up and reducing the cost of development of domain-specific AI applications. Multiple annotation methods with different types of annotation data are to be included in our tool, so that their effectivity can be evaluated and compared. The resulting tool is to be used by both amateur and expert annotators from the medical domain, who are frequently hired to help create such datasets. Using the tool should be intuitive, smooth for both categories of users. It has to respect the needs and customs of experts in medical practice, as well as minimize the overall cognitive load needed for annotation.

My aim here is therefore to:

- Analyze how to utilize human-computer interaction (HCI) methods and design patterns in the field of machine learning for the EFAN tool
- Explore and review existing tools and works related both to image segmentation in general and to annotation of images for machine learning
- Conduct appropriate user research between users of existing tools and potential users of the tool
- Design and create a prototype of the tool
- Conduct usability testing and iteratively improve the prototype for higher fidelity
- Prepare a foundation for further development of the tool

There is a parallel ongoing development of AI algorithms to be used on data created using this tool. After the first version of the EFAN tool is developed and AI algorithms are implemented, an eventual next phase is proposed. In this phase, the tool should be upgraded to utilize human-in-the-loop techniques for interactive annotation with live feedback from trained models. After the implementation is complete, researchers, medics, and other end users might use this software for viewing, analyzing, measuring, and fully segmenting medical images as well.

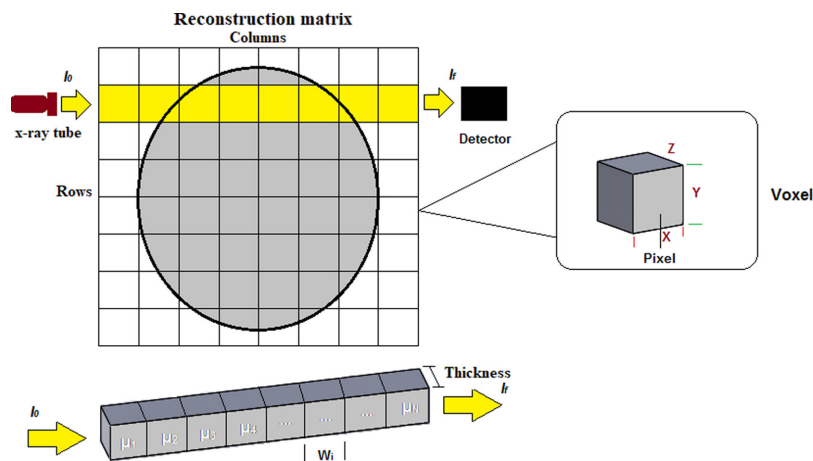
# Chapter 2

## Foundations of imaging data

Before I dive into exploration of related works, I would like to review a few key concepts. This chapter introduces the foundations of viewing and annotation tools used for medical imaging.

### 2.1 Representing image data

Both CT and MRI produce volumetric data. Nevertheless, such data is represented in slightly different ways. CT images are already usually represented in a 3D matrix, allowing easy slicing into three main axes. Corona et al. [10] illustrate how the volumetric image is reconstructed during CT scanning using the following figure 2.1 .



**Figure 2.1.** Schematic representation of a CT image and voxels; 1 to N represent different attenuation coefficients [10]

The CT-scan image obtained from the patient’s body, irradiated with X-rays, is a reconstruction from numerous measurements of attenuation coefficients, which are a measure of how easy bone and/or tissue can be penetrated by X-rays. Radiographic density is then calculated for each voxel using standardized Hounsfield units (HU) based on the degree of X-ray attenuation for a given voxel. This means that suitable viewing brightness presets for specific analysis tasks can be easily defined, since a given type of tissue will always roughly correspond to the same HU value.

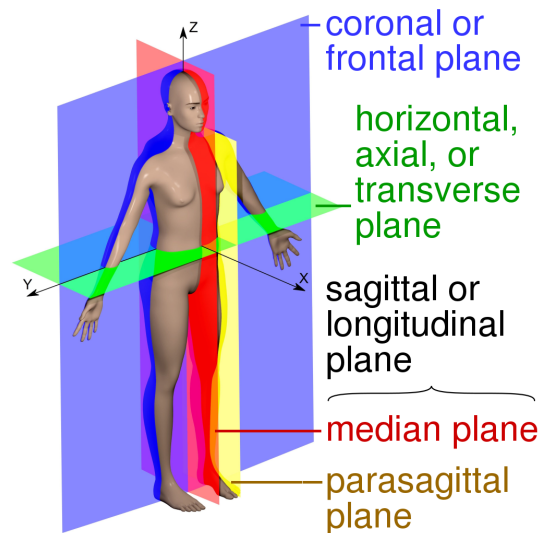
On the other hand, MRI-scans have the advantage of being non-invasive, since no ionizing radiation is produced during examination. For the cost of their longer examination time, they also provide better contrast resolution than CT. MRI-scan images were therefore mentioned by interviewed specialists to be preferred for identification of certain tumors or spotting subtle differences in tissues. The difference from CT-scan images data is, that they are often rather represented using non-isometric voxels and have varying measured intensities based on the procedure’s setting. For example, “T1”

and “T2” types of images are commonly recognized, where fat-containing tissues are brighter in T1 scans and both fat- and water-containing tissues are brighter in T2 scans [11].

## 2.2 Viewing image data

Given the task to display and annotate 3D medical data within the constraints of a 2D screen, the question arises: How can we present such data, so that we can work with it efficiently? The most commonly used approach in software and medical practice researched in this work is to display and work with slices (plane cuts) of the volumetric data. Slicing of the data is typically done according to standard anatomical planes [12]:

- A *transverse plane* (also known as axial or horizontal plane) is parallel to the ground; it separates the head from the feet.
- A *coronal plane* (also known as frontal plane) is perpendicular to the ground; it separates the front from the back.
- A *sagittal plane* (also known as anteroposterior plane) is perpendicular to the ground, separating left from right.



**Figure 2.1.** Visualization of anatomical planes in a human: sagittal (median) plane, parasagittal plane, frontal plane, transverse plane. [12]

Among the medical specialists interviewed, CT-scans were viewed with slices aligned to the mentioned three planes. Meanwhile, MRI-scans were often viewed as slices only corresponding to one of the planes. I am assuming this is due to the mentioned non-isometric nature of the voxels in MRI scans. The slices are typically organized according to the axis along which they were scanned, and can have different thickness based on the procedure’s settings. Nevertheless, many of the analyzed medical imaging systems allow 3D reconstruction even from non-isometric data and therefore viewing such images with slices aligned with either of the three main planes.

In addition to viewing data as slices, the analyzed systems can often use computer graphics algorithms to reconstruct and render a 3D image, which is then projected onto the 2D screen.

## 2.3 Storing image data

There are two main file formats for storing 3D medical image data that I considered in this work — DICOM and NIfTI.

### 2.3.1 DICOM

In medical practice, Digital Imaging and Communications in Medicine (DICOM) is a standard for communication and management of medical imaging information and related data. The DICOM standard specifies network communications, syntax and semantics of commands, directory structures and storage, and, most importantly in our case, a file format for medical images [13]. Image files in this format have a `.dcm` extension and contain both image data and associated metadata such as patient information, imaging parameters, segmentations and other type of annotation data.

### 2.3.2 NIfTI

In the realm of neuroimaging research, the Neuroimaging Informatics Technology Initiative (NIfTI) has established a standard for image data formats [14]. The standard facilitates interoperability and efficiency in handling neuroimaging data and is exemplified by the `.nii` and the zipped `.nii.gz` file extension. The data in this format are easily anonymized. This file format is for example used for the images in The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS) dataset [7], [15], [16]. In this work, I have utilized an image from this dataset for testing and development purposes.



# Chapter 3

## Related Work and Software

In this chapter, I first present relevant existing open-source solutions and related published works. Next, I continue with relevant paid solutions. Finally, I present some currently used tools in radiology and oncology practice in Czech clinical facilities.

### 3.1 Open-source software

In this section, we analyze existing public works and tools that allow viewing of image data, creating annotation and possibly automatic contouring functionalities.

#### 3.1.1 3D Slicer

3D Slicer is a free, open source software for visualization, processing, segmentation, registration, and analysis of medical, biomedical, and other 3D images and meshes; and planning and navigating image-guided procedures. It offers tools for manual contouring with the addition of a few automated algorithms. Additional functionality can be added through extensions [3]. A screenshot from the software is shown on 3.1 .

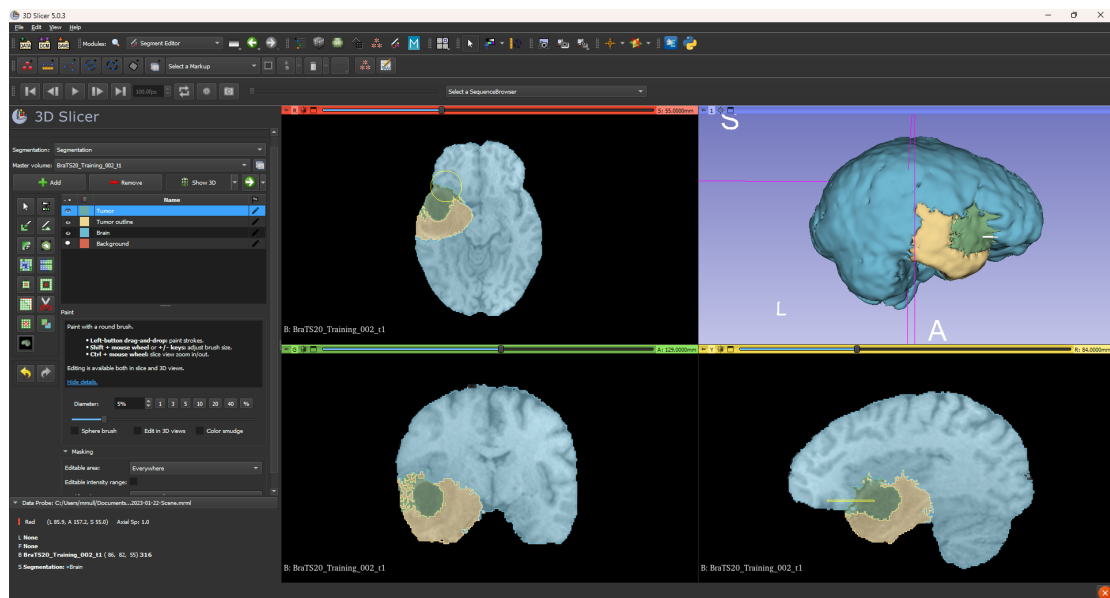


Figure 3.1. 3D Slicer software with default layout and Segment Editor module open.

Due to its popularity and many offered relevant features, I decided to thoroughly test 3D Slicer myself in the research phase. I cover the details in appendix A.3.

### ■ 3.1.2 OHIF

The Open Health Imaging Foundation (OHIF) is an open source, web-based, medical imaging platform. It aims to provide a core framework for building complex imaging applications. It offers viewing, measurement, and segmentation tools. At this point OHIF offers manual segmentation using a brush, but can be also used to annotate the data in the DICOM server via the MONAI Label server.

### ■ 3.1.3 MONAI Label

A significantly related work *MONAI Label: A framework for AI-assisted Interactive Labeling of 3D Medical Images* [5] presents a free and open-source platform that facilitates the development of AI-based applications. It aims at reducing the time required to annotate 3D medical image datasets. The framework connects as a plugin to one of other imaging tools that can provide a GUI — in case of radiology, that is 3D Slicer or OHIF. On the plugin’s GitHub repository, the authors also mention it can be connected to QuPath, Digital Slide Archive and CVAT for use in pathology. Finally, its connection with CVAT can be also used in endoscopy.

The framework itself is running in backend and offers three interactive (DeepGrow, DeepEdit, Scribbles-based) and one non-interactive (Automatic Segmentation) approaches for the GUI [5].

#### Interactive workflow

**DeepGrow** is an interactive segmentation model where the user guides the segmentation with positive and negative clicks. The positive clicks, inside the region of interest, expand the segmentation to include that location, while the negative clicks are used for contracting the segmentation to exclude the clicked region from the region of interest. DeepGrow 2D allows the user to annotate one label on the image one slice at a time, whereas DeepGrow 3D can annotate whole volumes.

**DeepEdit** extends DeepGrow’s click-based segmentation to allow for click-free segmentation inference and click-based segmentation editing. More specifically, it allows the user to perform inference, as a standard segmentation method (i.e. UNet), and also to interactively segment part of an image using clicks as DeepGrow does.

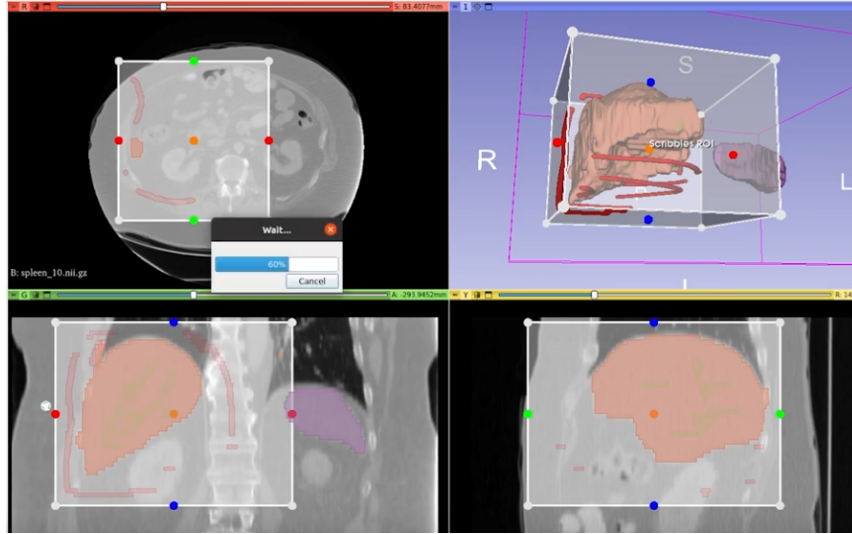
**Scribbles** are free-hand drawings, such as drawing with a pen on paper, which have been widely employed to propose a range of interactive segmentation methods. They provide natural interaction, which most annotators are already familiar with. The scribbles-based interactive segmentation can be used in two different modes. The first mode uses scribbles to generate segmentation labels. The second refines segmentations from a deep learning model using user-scribbles. The scribbles-based workflow is shown on figure 3.2 .

#### Non-interactive workflow

MONAI label also offers a traditional Segmentation model, which is first trained and then only used in inference, providing fully automatic segmentation. The result of the segmentation can then be edited only through manual methods, such as using a brush tool.

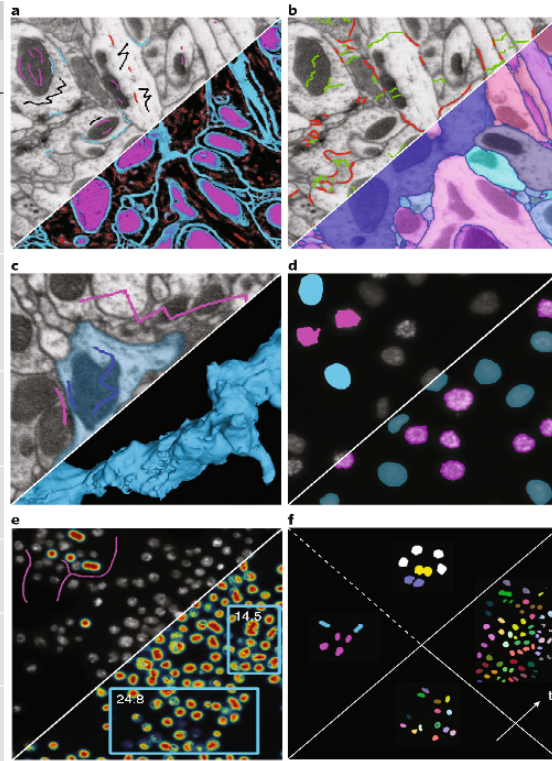
### ■ 3.1.4 Ilastik

The work Ilastik: interactive machine learning for (bio) image analysis by Berg et al. [17] aims to present an easy-to-use interactive tool that brings machine-learning-based (bio)image analysis to end users without substantial computational expertise. The



**Figure 3.2.** Demonstration of scribbles approach from MONAI Label in 3D Slicer

Table 1   Summary of the annotation modes in ilastik workflows			
Workflow name	Input data	Annotation mode	Result
Pixel classification or autocontext (Fig. 1a)	Up to 5D	Brush Strokes	Semantic segmentation: assignment of pixels to user-defined semantic classes such as “foreground” or “cell membrane”
Multicut (Fig. 1b)	2D/3D + channels	Mouse clicks on fragment edges	Instance Segmentation: partitioning of an image into individual objects (instances)
Carving (Fig. 1c)	2D/3D	Brush strokes inside and outside the object	Single segmented object
Object classification (Fig. 1d)	Up to 5D	Mouse clicks on objects	Objects categorized in user-defined classes
Counting (Fig. 1e)	2D + time + channels	Brush strokes and clicks on single objects	Object counts in images and ROIs
Tracking (Fig. 1f)	Up to 5D	Mouse clicks on dividing or merged objects	Object assignment to tracks through divisions
Tracking with learning	Up to 5D	Short tracklets	Object assignment to tracks through divisions



**Figure 3.3.** Figures from the ilastik work presenting annotation workflows and demonstrative images [17]

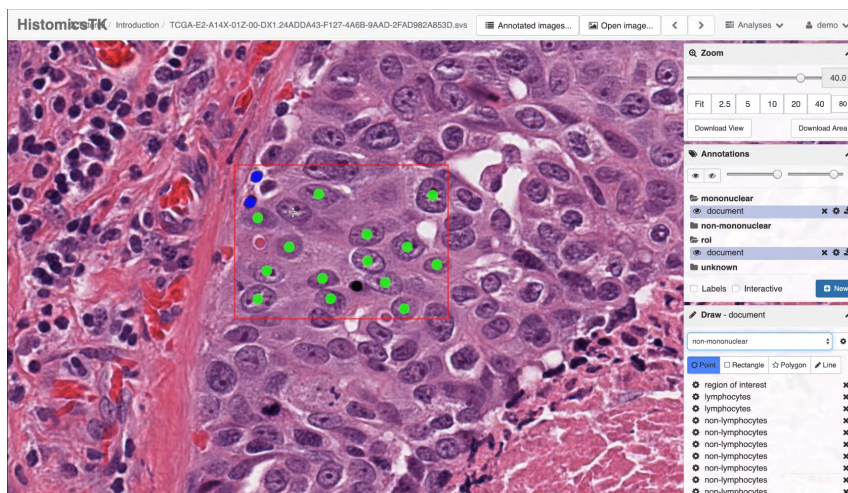
ilastik tool offers multiple workflows which encapsulate multiple established machine-learning-based image processing tasks. A summary of workflows with supplementing demos from the work can be seen in figure 3.3 .

From these workflows, Pixel classification, Carving and Multicut are related to our use. In pixel classification, the user uses a brush to create strokes inside objects corresponding to pre-defined classes. The model then assigns each pixel (or voxel) a corresponding class. Carving and Multicut rather focus on the separation of continuous

regions than evaluating a class per pixel. In carving, positive strokes are drawn inside and negative strokes outside of a region corresponding to a single class/object. The model then outputs a segmentation, delining the target object. Multicut brings this further by allowing multiple classes at once. The user can again use positive and negative strokes, but this time, the user rather delines the borders between structures. Positive strokes mean a class should pass through that area, negative strokes mean there is a division between classes.

### 3.1.5 Digital Slide Archive and HistomicsUI

Digital Slide Archive (DSA) is an open-source web-based platform for managing large collections of histologic images, designed by Gutman et al. [18]. This software therefore stands off from other radiology-focused applications and rather focuses on image collection and annotation in pathology. I mention this software, since the Amgad et al. [6] have utilized this platform for annotation of AI-training datasets in a related way. The annotation is realized inside HistomicsUI, which is implemented as a Girder plugin [19]. The authors then define an annotation as a single rectangle, point, or polygon. An annotation has an assigned style, which specifies its color and attributes and correspond to e.g., a cell or tissue type. Such styles can be exported and imported using a `.json` file. The UI offers navigation between images, zooming and moving among an image, a pane with annotation styles and a pane with the list of all annotations and controls for creating them.



**Figure 3.4.** Annotation using ROI and points of a cell nuclei within the HistomicsUI

### 3.1.6 QuPath

Quantitative Pathology & Bioimage Analysis (QuPath) is an open and extensible software platform for bioimage analysis. This desktop software is also focused on digital pathology applications rather than radiology. I mention this software in respect to previously described Digital Slide Archive to provide an alternative example, that can be used with MONAI label's AI annotation. It allows connecting to public or private servers to import images. The software then allows annotation using various tools, such as the line, polygon, or wand tool, which works similar to the brush tool, except for spreading into neighboring pixels with similar intensities.





**Figure 3.5.** Annotation of a tissue within QuPath

### ■ 3.1.7 Mango

Mango – short for Multi-image Analysis GUI – is a rather simple viewer for medical research images. It provides analysis tools and a user interface to navigate image volumes. There are three versions of mango, each adjusted for a different platform: desktop, browser, and mobile (Apple iPad). I have further explored and reviewed the software in section 5.2.5.

### ■ 3.1.8 ITK — Snap

ITK-SNAP is a free, open-source, multi-platform software application used to segment structures in 3D and 4D biomedical images, originally developed at the University of North Carolina. ITK-SNAP provides semi-automatic segmentation using active contour methods, as well as manual delineation and image navigation [20]. The semi-automatic segmentation consists of three steps. First, thresholding is done. Next, 3D bubble seeds (points) are marked. Finally, an evolution algorithm is started.

### ■ 3.1.9 MITK

The Medical Imaging Interaction Toolkit (MITK) is a free open-source software for the development of interactive medical image processing software. As I have discovered only later during my work, one of such applications is Nvidia’s AI-Assisted Annotation [21], for which I recommend further exploration.

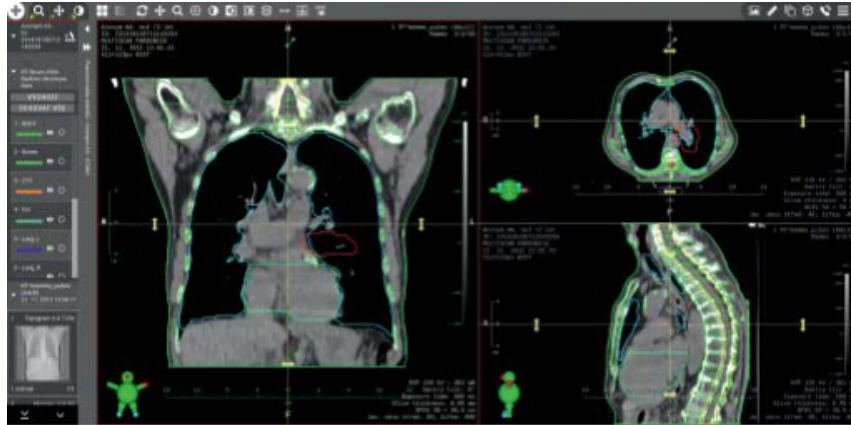
## ■ 3.2 Software Used in Medical Practice

This section lists software I encountered during my own research and during two qualitative interview sessions with medical experts in oncology and radiology departments. The user research results are covered in detail in section 5.1.

### ■ 3.2.1 Marie PACS

MARIE PACS is a solution for electronic processing, archiving and distribution of image data developed in the Czech Republic [22]. The system allows working with and viewing DICOM patient data.

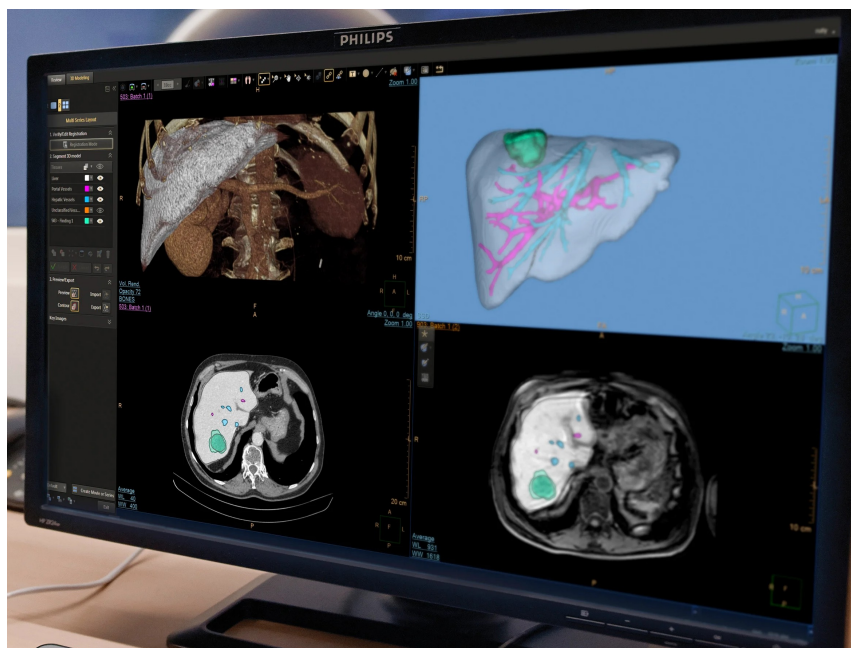
One of the main parts of this solution is the *Marie WebVision* (or also *Marie Server Web*) viewer. The software provides viewing, measuring and contouring functionality and runs in a web browser. A demonstration can be seen in 3.1 .



**Figure 3.1.** A view of a CT scan and the capability to measure contoured 3D volume in Marie WebVision. [23]

### 3.2.2 Philips Intellispace Portal

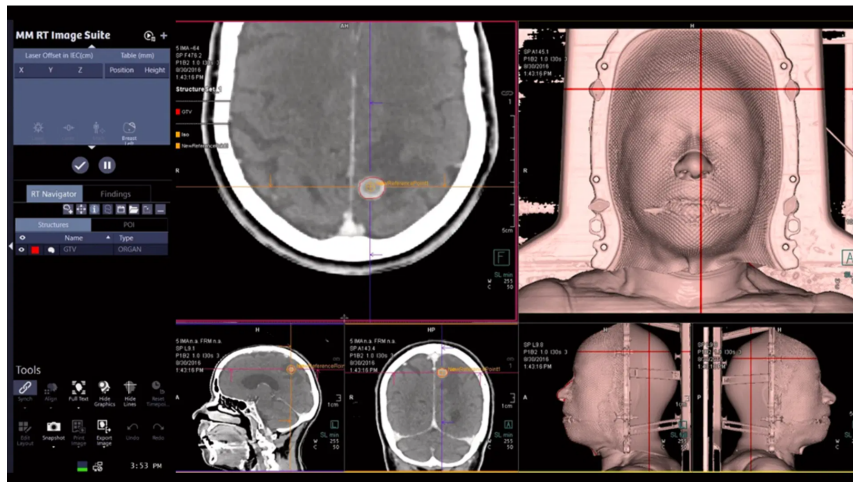
Philips Intellispace Portal consists of multiple modules for different workflows. One of such is a workflow for viewing and contouring image data. It offers automatic algorithms to find and mark various health problems in the data. An image from their newsletter can be seen in 3.2 .



**Figure 3.2.** Philips Intellispace Portal software with default layout in 3D modelling mode. [24]

### 3.2.3 Siemens Syngo.via RT Image Suite

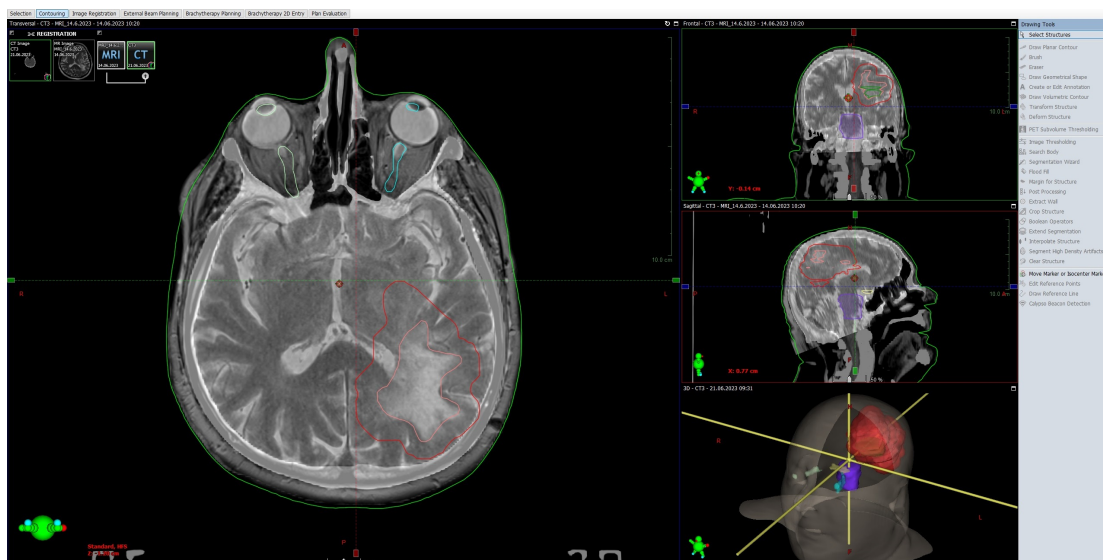
Syngo.via RT Image Suite offers multimodality image viewing and contouring. It allows leveraging image data to get a comprehensive view of patient. It further offers an efficient contouring solution designed for radiation oncologists [25]. AI-Rad Companion can be used for AI-assisted contouring. See a screenshot on figure 3.3 .



**Figure 3.3.** A screenshot of the VSyngo.via RT Image Suite software in Patient marking mode. [25]

### 3.2.4 Varian Eclipse

Varian Eclipse is a software offering solutions for treatment planning in radiation therapy. It is therefore used in oncology practice for precise planning for advanced cancer care. Besides treatment planning itself, it offers workflows for viewing and contouring the image data needed for the planning. An important functionality is fusing MRI with CT images together for taking the advantages of both types of scans. Moreover, it provides AI-assisted functionalities for contouring [26]. A screenshot is shown on figure 3.4 .



**Figure 3.4.** Varian Eclipse; fused MRI and CT; default layout; contouring mode

Note that the Varian company got merged into Siemens [27], so it is possible the name might eventually change in the future.

## ■ 3.3 Other Commercial Software

In this section, I briefly introduce two other paid software, which have been analyzed during this work (see 5.2).

### ■ 3.3.1 RadiAnt

RaidAnt is a PACS DICOM viewer for medical imaging data. It provides basic tools for the manipulation and measurement of images. Its workflow resembles more of the one in clinical practice, due to the use of PACS system and DICOM images.

### ■ 3.3.2 V7 Darwin

V7 Darwin is a web-based commercial tool for annotating image-based datasets for training AI applications. Its main focus is on 2D images, but it has been lately able to work with 3D medical data as well.



# Chapter 4

## Methodology

Aiming to establish a solid framework for the development of the EFAN tool, this chapter begins with a review of human-computer interaction and user-centered design principles. Further, it undertakes an exploration into machine learning-specific challenges and human-AI interaction. Finally, utilized methods and technology in my work are described in detail.

### 4.1 Utilized HCI and UCD Principles

According to Shimodaira [28], Human-computer interaction (HCI) is a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of the major phenomena surrounding them. User-centered design (UCD) then falls into the HCI field. It is an approach to software engineering that focuses on the user in all stages of development.

For utilizing UCD, I've tried to base this work on the following core principles from presentations by Mikovec [29]:

- Find the user needs and take them into account.
- Be in permanent contact with potential users throughout the design process (for example, by iteratively testing the UI prototypes).

To my understanding, the UCD process can be then roughly described as follows. First, requirements are assessed based on user needs. A system is then designed in accordance to those requirements. If possible, design decisions are communicated with users for feedback. Once there is a testable implementation, it is tested with users to see, whether the requirements were met and whether new requirements arise. The process should be repeated multiple times before system deployment.

The system development process can also be broken down into four consequent phases [29]:

1. User research
2. Design
3. Engineering
4. Deployment

Nevertheless, as hinted before, each phase should think of the user in some way. This work covers the user research and design phases.

#### 4.1.1 User Experience and Usability

Ortiz, Juan, and Aurisicchio[30] reviewed theories and models on User Experience (UX) in their work. The authors conclude on the following four key elements required for understanding of UX: a **user** experiencing an **interaction** with an **artifact** in a given **context**. The process is described as subjective, holistic, conscious, emotional and

dynamic. According to ISO 9241-210:2019 [31], we can summarize UX as “user’s perceptions and responses that result from the use and/or anticipated use of a system, product, or service”.

Whilst UX encompasses the entire experience of a user with a system, usability refers to the quality attributes of the system, which indicate how successfully it allows the user to accomplish a specific goal. In the words of the ISO standard [31], it is the “extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use”. According to the presentations by Míkovec [29], good usability can be achieved by maximizing the following properties of the system:

- *Simplicity* — learning how to use the system is easy and fast;
- *Memorability* — the usage is easy to remember and recall;
- and *Efficiency* — once learned, it is fast to use the system (carry out tasks efficiently).

In an alternative definition by Nielsen [32], *Simplicity* is omitted. Instead, *Learnability* is used for describing how easy it is for users to accomplish basic tasks for the first time they encounter the design. The author further includes how many and how severe *Errors* do users make and how pleasant it is to use the system as *Satisfaction* property. I have kept these properties in mind for both design and user testing of the EFAN tool.

## 4.2 Utilized User Research Principles

User research covers a wide area of techniques for collecting information about users. To understand the target users and design an effective tool for them, I followed the recommendations of Franc [33] to utilize both qualitative and quantitative research methods. Qualitative research is focused on gathering insights and formulating questions, which we can then use to form our hypotheses to be tested through quantitative research. Quantitative research then allows hypothesis testing by generalization to the whole population using exact numerical data collected from surveys and controlled tests.

To summarize the main concept:

1. qualitative research gains *insights* and *questions*
2. these are transformed into *hypotheses*
3. which are tested through *quantitative* research
4. and *interpreted* in the context of qualitative data

In my work, I interpret insights as key pieces of raw findings that bring useful information for the prototype’s development. Such insights can carry emotional feelings of users, discovered current technological possibilities or constraints, or, for example, the way users are used to handling specific tasks. I also utilize the concept of writing down questions that arose during user research for further exploration. Finally, I formulate hypotheses which represent my interpreted findings. Due to limited availability of medical experts, I have not utilized any quantitative research techniques and focused primarily on acquiring insightful information through qualitative research and testing hypotheses directly through usability testing on implemented prototype versions.

Franc [33] also outlines how to measure the success of user research through the following properties:

- *Validity* — Did we measure what we are supposed to measure?
- *Reliability* — Does the measurement remain consistent if we repeat it?
- *Systematic error* — Did we correctly use experimental methods and equipment?
- *Random error* — How much did noise affect our measurements?

However, the success of qualitative research, which is rather based on personal opinions and interpretations, cannot easily be measured using the standard criteria often found in quantitative research. Because only qualitative data were collected in my research — due to the limited availability of medical experts and a deliberate choice to seek in-depth insights — assessing these aspects proved challenging for me. An effort to address this was briefly made in chapter 6. The reason I chose not to include quantitative research in my study is also linked to the high standards for quality it demands. There's a risk that numerical results might mislead if the research isn't executed meticulously [34].

## 4.3 Utilized User Research Methods

Here I describe the methods I used for conducting user research.

### 4.3.1 Qualitative Interviews

I have chosen qualitative interviews as the user research method best fitting this case. They allowed to maximize the output from the limited time of medical experts.

### 4.3.2 Field Research

Since the target users for our tool are going to include medical experts and since seeing such experts using existing medical imaging software in practice could provide valuable insights, I decided to utilize aspects from field research during my qualitative interviews. Field studies are also recommended by Nielsen [32] when improving usability is desired.

Babbie [35], as cited in [36], describes field research as a study with the purpose to understand attitudes, practices, roles, organizations, groups, and behaviors in their natural setting. The core technique in this type of research is observation. The authors distinguish between two subtypes of field research: participant observation and direct observation. While participant observation involves the researcher acting alongside participants in the field, direct observation the researcher in a passive role as he only observes participants in the field but doesn't actively participate.

During my research, I have taken on the direct observation practice and let the participants show me how they complete their daily tasks.

### 4.3.3 Interview Evaluation

I have taken either an audio recording, or paper notes combined with photos from the interviews. For further evaluation of these notes, I have chosen to write down key insights. I have personally defined insights as key points, that bring an information value about: emotions of the user, technological aspects of used software, usability of the used software, motivation behind tasks in related domain, and further information on conducted tasks. Further, I have asked myself to pose questions and formulate hypotheses based on the interview results. Such key insights, questions and hypotheses were then used during the development of a prototype for the tool.

## 4.4 User Personas

Franc [33] presents defining personas as a user research method, which was popularized by Alan Cooper. A single persona represents a hypothetical user subgroup. All personas together ideally cover the whole spectra of target users. A persona is defined using a detailed, narrative description, also containing their motivations. I have attempted to create such personas for this work.

## 4.5 User Scenarios

Míkovec [29] defines user scenarios as an informal narrative stories. Such stories are written in a simple, ‘natural’, personal and non-generalized way. They help visualizing how potential users—researchers, amateur annotators and medical professionals in this context—interact with a system to achieve a specific goal or resolve a problem. Moreover, as mentioned in an online guide by Dovetail [37], the focus of scenarios should be on the reasons and motivations why are users doing what they do, rather than describing how in detail.

### 4.5.1 Scenarios in this work

I was originally inspired by a structure of scenarios according to the mentioned guide [37]. The same structure was also described by Salazar [38]. According to the author, scenarios should be centered around one task that is key to the developed system and include the following five elements:

1. an actor;
2. a motivator;
3. an intention or intent;
4. an action;
5. and a resolution.

Salazar further argues that it is a good practice to use a specific persona as the actor carrying out the scenario. Doing so should provide additional context and insight about how different types of users approach tasks similar to which the EFAN tool is about [38]. In addition to the five elements, including problems or pain points that users may experience during their customer journey is advisable for consideration [37].

I have interpreted the previously introduced key elements of a scenario as follows. An actor corresponds to a previously created user persona — including demographics, personal traits, and relevant habits. A motivator should cover the underlying problem or reason why users need a specific solution. An intent describes the goal the user wants to achieve. An action specifies how the user reaches their goals, including the steps they follow. A resolution tells how the user reaches a solution and why do they choose it.

Nevertheless, I have slightly deviated from these standards by creating longer, detailed scenarios which also cover multiple problematic situations that I discovered during user research. I believe this allowed me to better visualize situations that I want to avoid happening for users using the designed tool.

## 4.6 Hierarchical Task Analysis

Task analysis in general can be defined as the broad practice of learning about how users work (i.e., the tasks they perform) to achieve their goals [39]. A task then refers to any activity that is usually observable and has a start and an end point. According to Browne [40], Hierarchical Task Analysis is a type of task analysis which involves collecting information using data collection technique (e.g. interviews, observations), analysing it and then creating a task models. It focuses on the observable behaviour of users.

The author proposes the following guide for HTA creation:

1. Identify users' primary goals and identify main tasks of concern;
2. Design and conduct data collection to elicit information about: the goals that users are trying to achieve;
3. Analyse the data to create specific task models;
4. Generalise across the specific task models to create a generic task model;
5. Check models with users, other stakeholders and iterate.

In the end, I have not followed this guide exactly. More on how I actually evaluated user tasks in section 5.3.3.

## 4.7 Functional Design

Functional design is an approach rather used in project management in general. The Functional Specification Document (FSD) is written by the project's Business Analyst and provides detailed information on how the system solution will function based on what the requested behavior is [41]. Detailed functional requirements, including use cases, system inputs and outputs, process flows, diagrams, and mockups are part of this document. Such functional specification should not define the implementation of the proposed system and rather focus on outer agents and their interaction with the system.

Since the development of the resulting prototype became a rather complicated process, I have utilized some mentioned techniques to write down proposed features in a more structured way. Writing down a custom version of the functional specification document therefore served as a way for logically organizing my design and capturing requirements from interviews and usability testing in it.

## 4.8 Prototyping

Prototyping is a common experimental process where design ideas and concepts can be efficiently tested before a system is passed for actual implementation. I have proceeded by iterative refinements of designs I proposed, frequently looking back on insights gathered from user research and related software testing.

In this section, I further introduce used technologies and their patterns during the development process. Finally, I talk about the importance of visual design in the EFAN tool and its prototypes.

### ■ 4.8.1 Used Technologies

Two main prototyping tools were used, Figma and Axure RP, supplemented by paper sketching and creating diagrams in Diagrams.net.

#### **Figma**

Figma is a set of collaborative web applications for interface design developed by Figma, Inc. I have chosen Figma, since it allows fast prototyping of both web and desktop applications, and it was a tool that I had little previous experience with. I have particularly used Figma design for the prototype and FigJam for initial sketches and notes.

#### **Using Figma Components**

In Figma design, I have explored, learned and utilized the concept of reusable components among the design. A useful aspect was the possibility to use component variants for e.g., different states of a button. Further combining variants with variables, I strived to make the prototype interactive in the primarily tested areas, so that minimal simulation of interactions is needed, therefore providing more exact results of the tests.

#### **Using Design Tokens**

In a later phase of the prototype, I have started using Design tokens within Figma variables. According to an official Figma video explanation [42], a design token represents a single piece of design information (color, text, size, space, font, animation...). The method further utilizes aliasing — nesting the structure of variables, where one references the other. The most common structure is mentioned as being the following:

- Primitive / global tokens
  - Represent what values exist in our system and are usually not applied directly, except for fonts or animations. An example value is a color defined in a raw hexadecimal value with a token named.
- Semantic tokens
  - Represent how the tokens should be used. Its value is typically a primitive token and its name corresponds to its use, such as a primary surface color.
- Component specific tokens
  - Represent where should the token be used, such as on a primary button in a default state. Its value can take a semantic token or a primitive token.

The author further recommends fitting the structure to the system's needs, such as using only use semantic or component specific tokens.

Although applying the design tokens was not crucial for the prototype, it promised a more structured approach to the design and making further adjustments to the prototype more efficient, since values do not have to be manually copied. I have utilized primary tokens and a compromise between semantic and component tokens.

#### **Axure RP**

Axure RP is a rather advanced prototyping desktop software developed by Axure Software Solutions. It also allows developing for web or desktop. I have planned on using Axure RP for the hi-fi version of the prototype. I had to learn how to use this software.

#### **Diagrams.net**

Diagrams.net (previously draw.io) is a cross-platform graph drawing software. Its interface can be used to create diagrams such as flowcharts, wireframes, UML diagrams, organizational charts, and network diagrams. I have utilized Diagrams.net for creating diagrams and brainstorming.

## 4.8.2 Visual Design

During the prototype development a question arose, whether and to what extent propose visual design aspects of our tool early in the prototyping phase. Since easy and efficient learning are considerable attributes of the EFAN tool's usability, I refer to Ghai and Tandon's work on the positive impact of aesthetic visual design on usability of e-learning [43]. The authors claim that adequate use of graphics, typography, and consistency can improve the usability of e-learning tools. I have therefore decided to incorporate a consistent visual aesthetic and the use of icons already in the prototype. Consequently, it could lower the chance of finding usability problems due to poor visual aesthetics and allow stronger focus on problems arising from poor functional design.

The next question to answer was which design guide to use as a base for the prototype. Following on the idea from section 5.4.1, Microsoft's Windows apps design seemed as a reasonable choice. Since they provide a freely available design guide, I decided to follow those guides to provide a consistent design for my prototype.

The prototype's visual design and color palette has thus been significantly inspired by Windows UI (WinUI) 3 Gallery [44], which is a demo app built on WinUI Library [45]. The colors in this app come out of the color design principles for Windows 11 applications [46]. According to the source, color is employed in a way to "help users focus on their tasks by indicating a visual hierarchy and structure between user interface elements". The colors are described as context appropriate and used to provide a calming foundation, subtly enhancing user interactions and emphasizing significant items only when necessary.

The prototype then leveraged a series of open-source icons from MingCute online library [47]. As mentioned, I assume adequate use of graphics should enhance the overall user experience.

## 4.9 Usability Testing

Usability tests are a crucial step in the development process of any tool or product. They involve observing real users interact with the tool to identify any usability issues or areas for improvement. By conducting usability tests, we can gather valuable feedback on how users actually use the tool and what difficulties they may encounter. This allows for the tool to be optimized for usability and user satisfaction.

### 4.9.1 Core Principles

I have conducted usability tests based on principles primarily from remembering lectures by Franc [33], further taking inspiration from other sources [48] [32] [49]. Moderated, in-person testing was preferred to collect qualitative insights into the proposed annotation tasks.

The core aspects for usability tests were choosing the participants, defining test objectives, and designing tasks in such a way to provide information about researched objectives.

I intended to collect both qualitative feedback (user comments, observations) and quantitative data (time taken per tasks, subtasks).



### 4.9.2 Actionable Insights

For evaluation of the results, I have chosen to utilize Actionable Insights based on their definition by TechTarget [50]. Actionable insights happen when humans analyze data to determine an action or a set of actions. In my case, such actions are adjustments of the prototype for better usability and formulation of question for further research. The analyzed data provide the **what**, while insights derived from the analysis of that data provide the **why**. An insight that leads to an action is more valuable than one that simply answers a question. In addition, specific actions are more effective than generic actions. Since I have collected a lot of information from the tests that I wanted to organize in a way to be able to adjust the prototype accordingly, defining actionable insights was a well-fitting approach.

## 4.10 HCI in AI Image Segmentation

This section first explores current practices in integrating HCI principles within systems utilizing machine learning techniques. Following is a section on interactive segmentation,

### 4.10.1 Human-in-the-loop machine learning

Researchers are defining new types of interactions between humans and machine learning algorithms, which can be grouped under the umbrella term of Human-in-the-loop machine learning (HITL-ML). According to Mosqueira-Rey [51], we can define following three approaches in HITL-ML based on who is in control of the learning process.

#### Active Learning

In active learning (AL), the system remains in control of the learning process and treats humans as oracles to annotate unlabeled data. What varies is the strategy of the system to choose data to be labeled by humans.

AL is useful in cases where acquisition of unlabeled examples is less expensive than the labeled ones. This holds true for medical image segmentation. Nevertheless, AL is rather suited for tasks that humans do generally well. This raises questions, whether manual segmentation using humans is an effective task enough. Moreover, users do not like to behave like simple oracles [51].

#### Interactive Machine Learning

Interactive machine learning (IML) can be defined in multiple ways. One definition presents IML as algorithms that can interact with agents and can optimize their learning behavior through these interactions, where the agents can also be human [51]. It works with the idea that humans and computers should work together on the same task doing what each of them does best at any specific moment.

This approach has been proven successful in the area of image segmentation. An expert user can use their knowledge to identify and specify areas which he intends to segment using only pointing or basic annotation tools, while the AI model can try to guess the whole segmentation area based on such inputs.

This approach also has the advantage of being an iterative approach. The user can continuously evaluate the results from the model and give new incentives at places where it needs improvement.

It is also the approach that is probably most fitting for the final version of the EFAN tool due to the proposed nature of minimal annotation interactions.



### **Machine Teaching**

While the previous approach was a cooperation of the teacher and the learner in the learning process, machine teaching (MT) places the responsibility firmly on the teacher. It tries to describe the idea of a teacher who teaches an ML model to an ML algorithm. The teacher is meant to be a human, although algorithms simulating teachers have become more common lately. An example field of application for this approach is robotics. Nevertheless, other fields might use of this approach and image segmentation could also be considered.

#### **4.10.2 Segmentation Specific Methods**

Wang et al. [8] review many currently used methods between researchers. I have briefly introduced the categories the authors specify in section 1.2. Among supervised, weakly supervised, semi-supervised and unsupervised learning, the Interactive Segmentation method falling into weakly supervised learning is of special interest in this work.

#### **Interactive Segmentation**

Building on similar foundations as Interactive Machine Learning, this method is characterized by minimal interactions with the user and interactive feedback. The user first provides simple annotations such as points, lines, or scribbles. The model takes this input and outputs a segmentation. The user then uses similar methods to give further refining input for the model.

#### **4.10.3 Applying HITL Techniques in This Work**

Even though the AI-algorithm part of the EFAN tool is currently still in separate development and subject to change, I have been designing the tool with algorithms resembling interactive segmentation techniques in mind. More specifically, the first milestone was set to design the prototype in a way, such that it allows utilizing multiple simple annotation methods, such as marking points, in non-interactive way. This way, different annotation methods can be first tested for their effectivity and usability on their own. A few promising candidates can then be chosen, for which the AI methods will be customized for. An interactive approach is then proposed as a next milestone.

# Chapter 5

## Results

This chapter describes the progress and the results of my work on the EFAN tool.

Through discussions with my supervisor, I have outlined the progress of my work approximately as follows:

1. Research on the topic of medical imaging, UCD, and HCI principles in AI applications
2. Research related works and existing software with similar purpose
3. Conduct user research between target users
4. Create personas, user scenarios and hierarchical task analysis
5. Design a low-fidelity prototype
6. Conduct usability testing
7. Create a high-fidelity prototype
8. Conduct further iterations of usability testing with effectivity measurements of annotation methods

The results of the first two phases were already covered in chapters 3 and 4. Therefore, a section on initial user research follows. After that, I present a further exploration of chosen related software, including a brief usability test. Next, I attempt on creating personas, scenarios, and task analysis. The remaining three sections then cover the prototype development process, including usability tests.

### 5.1 Initial User Research

This section first describes the challenges that I faced during the initial user research for this work. It follows with the results of qualitative interviews that I have conducted.

#### 5.1.1 Determining Target Users

I have taken on the methodology outlined in section 4 and started user research early in the work. In order to conduct user research, I needed a clearer picture of target users. According to UCD principles, having the user in the center of the development process is critical for making a well usable system. This was challenging for me, since the original users' needs that resulted in the assignment of this work were facilitated to me only through my supervisor. A direct contact with a related medical expert was delayed for hi-fi prototype testing phase, since basic usability testing could be conducted with amateur users and other experts I had contact with as well.

#### 5.1.2 Qualitative Interviews

In this section, I go over two conducted qualitative interviews with representative experts in radiology and oncology, which significantly helped in understanding of the medical imaging domain and to form the initial requirements for the EFAN tool's development.

### Target Participants

The target participants for this study were set as domain experts who work with medical imaging data, preferably MRI and CT, but not excluding other forms such as x-ray tomography. It is challenging to interview medical experts without proper funding for their spent time, but I have managed to interview two participants meeting the condition through acquaintance contacts. The two participants were from radiology and oncology departments, respectively. These departments were well-fitting for this study because of their relation to medical imaging (including 3D) and since such participants fall into the group of potential expert annotators to be hired for dataset annotation purposes.

### Preparation and Goals

I have prepared a supporting document for the interviewer. The document primarily served as a conversation base for me as the interviewer, as I was in charge of formulating the questions and leading the conversation in a meaningful way. It consists of a guide for the interview consisting of warm-up questions, main questions, and a conclusion. The document further lists other notes for the interviewer, specification of the target participants, and research questions and goals.

I have defined the study goals in a way to:

- learn about common medical imaging tasks in radiology and oncology;
- how are the goals of such tasks achieved through current medical imaging software;
- whether there are usability problems in their current workflow;
- and how relevant is the task of segmentation / annotation of imaging data in their jobs.

The full document including more details can be found in appendix A.1.1.

### Interview Environment and Procedure

Both interviews were conducted at the workplaces of the participants. I was able to sit on a chair next to the participants, while asking questions. After the interviews were complete, I asked the participants to show me the software they use. I asked them to showcase a simple contouring task, ideally representable of an actual scenario from their job. From the interviews, I have collected notes, which I then interpreted into insights. Further, I have formed hypotheses and questions, that I could utilize during prototype and test development.

### Radiology Interview Results

The first interview was conducted in the radiology department of a local hospital. Since the collected insights from the interview are a rather long list, I have included them in the appendix A.1.2. Here I present only the participant's profile and formed questions and hypotheses.

#### Participant Profile

Head of radiology department at a regional hospital, with 20 years of experience.

#### Formed Questions

When analyzing the interview, a few questions for the development of the EFAN tool arose:

- How should we interact with the viewer using a mouse (and keyboard)?
- Which layout should be shown by the default in the presented tool?
- How to present available tools to the user?

### Formed Hypotheses

Resulting hypotheses arising from the interview:

- In radiology, imaging software is rather used for viewing, measuring and analyzing purposes than for image segmentation.
- Nevertheless, segmentation is occasionally utilized for computation of surfaces and volumes of complex shapes, for specialized treatment procedures, or for teaching purposes.
- When auto segmentation is utilized, there is typically no need to edit slight inaccuracies, but rather decide, whether the segmented area actually correspond to an irregularity or it is a false positive.
- Manual segmentation is indeed tedious and time-consuming.
- Software intended for medical use offering advanced auto-segmentation and evaluation features cost a substantial amount of money.
- The most important view is the transverse plane slice. Other planes are sometimes used too. 3D view is good for example for demonstrations / explanations, but is not necessary otherwise.

### Oncology interview results

The second interview was conducted at an oncology department of a local hospital. Similar to the radiology interview, the collected insights and other notes are included in appendix A.1.3.

### Participant Profile

The participant's job position was a physicist. Their job primarily includes radiotherapy planning. That is, receiving 3D medical image data with completed segmentations and planning a radiotherapy treatment based on them. Due to their expertise with the used software including creating segmentations, the participant occasionally teaches medical doctors on how to use contouring software to create proper segmentations for the treatment planning.

### Formed Hypotheses

Resulting hypotheses arising from the interview:

- It is essential to keep undo-ability of user actions in mind when creating an annotation tool.
- Contouring is done solely on the transverse plane slices most of the time.
- Varian Eclipse is a well-designed segmentation software and represents a standard to which oncology experts are accustomed to.

### 5.1.3 Clarification of Usage by Target Users

At first, I struggled to properly understand the assignment and the original target user needs. My first understanding was that the assignment is to propose a tool for medical doctors to be used in their jobs, and such a tool will have an effective AI segmentation functionality. This functionality needs to be designed with care and have good usability with minimal interaction, such that it allows saving time of medical professionals. This is why my research was initially focused more on works such as Human-in-the-loop machine learning: a state of the art [51] and related concepts. I have also conducted the interviews with such reasoning in mind.

Through discussion of the results of the interviews with my supervisor, I have then further clarified the actual desired goal and the meaning of AI-applications in the assignment, such that the tool is rather directed towards researchers and annotators. The

core principles remained, but the shift was in the motivation of medical experts for using the tool. The primary purpose of the EFAN tool was clarified to be enhancing the development of AI applications in medical imaging domains. Such applications might be used by medical professionals in the end, as previously thought, but the current focus is on allowing an efficient development of such applications through the EFAN tool. The current target users are therefore annotators of medical imaging datasets in general. Such annotators can range from amateur volunteers trained by researchers, to medical domain experts including radiologists and other physicians, whose expertise comes at a premium. This shift in thinking also made me realize that I need to think about the problem in context of multiple consequently annotated images in a continuous flow which I have taken into account in further stages.

Nevertheless, medical facilities use image segmentation in patient treatment, creating opportunities for companies to develop specialized software. The interviews with medical specialists reveal that such software is often very costly, even for hospitals. This leaves the possibility for medics to use our software for certain non-critical tasks open.

## 5.2 Related Software Exploration and Testing

To learn how current medical imaging and annotation software works, I have further inspected and tested a few of the previously researched existing applications on my own. This chapter first covers an initial idea for comparing related software. An in-depth review on 3D Slicer based on personal usability testing. Following are brief findings with additional details from my exploration on Volume Label, RadiAnt, Mango.

### 5.2.1 Comparison Table Idea

To be able to constructively compare related software, I have decided to explore the software and create a comparison table where I include the following information about each software: company, license, supported operating systems, supported types of files, integration with other software and tools, PACS functionality, viewing controls, measurement tools, segmentation / annotation tools, AI functionality. Nevertheless, since I started with 3D Slicer which offers a substantial amount of features, the table started growing in columns rapidly. Filling the whole table while organizing it, even by shrinking some columns into broader categories, would therefore cost a considerable amount of time. This is why such idea was abandoned and rather a thorough testing of 3D Slicer was utilized for collecting insights, and other tools were explored, while reviewing their controls and annotation functionality individually.

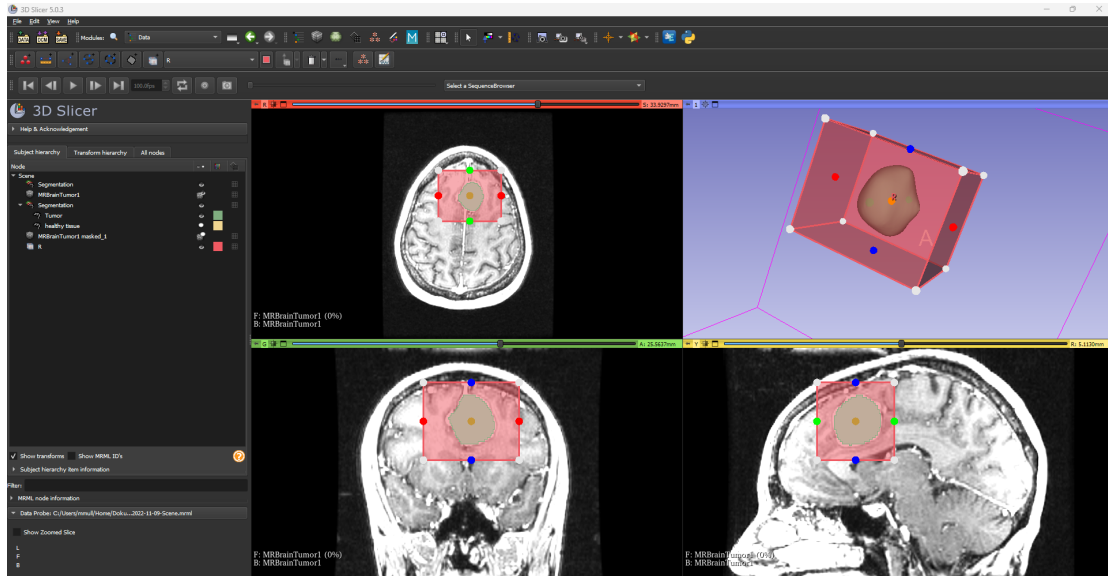
### 5.2.2 3D Slicer and MONAI label

3D slicer matches many requirements for the EFAN tool except good usability. I have conducted a brief usability test myself and written a thorough review on this software. The first test included annotating a tumor in a brain image. I followed with a second test, trying to utilize MONAI label's interactive AI-assisted annotation. I planned annotating the same image for A/B method comparison, but there was no sample brain tumor model available, so I annotated sample liver and aorta images in the end. A document with notes from the tests can be found in appendix A.3. A summary of insights based on these notes follows in this section.

#### User Interface

The user interface can be seen in 5.1 and in previously shown 3.1 . The GUI consists of a top menu with icon buttons and module selector, a left pane with controls of the

currently selected module, and an image viewer. The viewer is split into four views by default. Top-left (red) represents axial plane, bottom-left represents coronal plane, bottom-right represents sagittal plane and top-right shows a 3D view of created segmentation and ROI. Each of the views can be quickly enlarged to fill the whole viewer. The layout can be changed to for example conventional layout, which enlarges the 3D view and keeps the other three next to it, stacked on top of each other. Nevertheless, I was not able to switch which view is going to be the larger one.



**Figure 5.1.** 3D Slicer software with default layout; ROI with tumor annotation; Data module open

## Annotation

Slicer offers two default ways to create annotations. It also offers an already deprecated Annotations module, which I will not cover here. The first way is using the Markups module. This module allows marking points, lines, curves, planes and 3D ROIs onto the image. It also allows measurements of lengths, volumes, and angles. The second way is to use the Segment editor and Segmentations modules. It offers several advanced manual and semi-automatic tools to mark segments in the image. Each segment has a name, a color, and a status (not started, in progress, complete, and flagged) and can be visually represented as a binary labelmap, a closed surface or a fractional labelmap. AI assisted annotation can be achieved through the MONAI label plugin, which appears as a separate module.

## Controls

Control	Action
Scroll wheel	Change slice
Left click + drag	Use tool
Right click + drag	Zoom
Middle click + drag	Pan
Shift + Left click + drag	Pan
Shift + Mouse pointer	Change slice on adjacent views

### Usability Test Results

In the appendix A.3 I present a list of insights that I interpreted from the usability test notes. I have then formed hypotheses and questions, which follow here.

#### Formed Hypotheses

I have formed the following hypotheses:

- In the EFAN tool, it could be beneficial to provide a way to reduce the GUI elements based on context of the current goal the user wants to achieve. Such context selection needs to be clearly presented to the user.
- At startup, the tool should ask the user and guide him through the annotation setup process, so that no external tutorials are needed.
- Make to the user clear, which annotation method we are using and how it should be used.

#### Formed Questions

Questions formed through discussion with my supervisor:

- Should our tool allow textual, or voice notes tied to segmentation classes / ROIs?
- Various mouse controls were used along already explored software. Which controls would fit our tool the best?
- Which views layout should be used by default in our tool?

### 5.2.3 OHIF

Since OHIF was mentioned among the software usable with MONAI label and had a live demo to try, I have briefly explored its basic controls for viewing and segmentation.

#### User Interface

The first opened user interface consists of a study list with DICOM properties shown, from which they can be opened. A study consists of one or multiple related image series - this approach is similar as to software used in clinical practice. Opening a study is done through either **Basic Viewer** or **Segmentation** buttons. The former opens the same view as the latter, just with the segmentation pane being hidden by default.

The viewer's layout is a single view by default, but can be changed to a numerous different ones as can be seen on the screenshot 5.2. It consists of series switcher on the left, the view in the middle, hidable segmentation panel on the right and a menu bar above. On the menu bar, the leftmost buttons returns the user to the study list, in the middle is a selection of tools and on the right are settings and patient information.

Overall, the UI feels clean and modern compared to other tools, with icons clearly indicating their purpose.

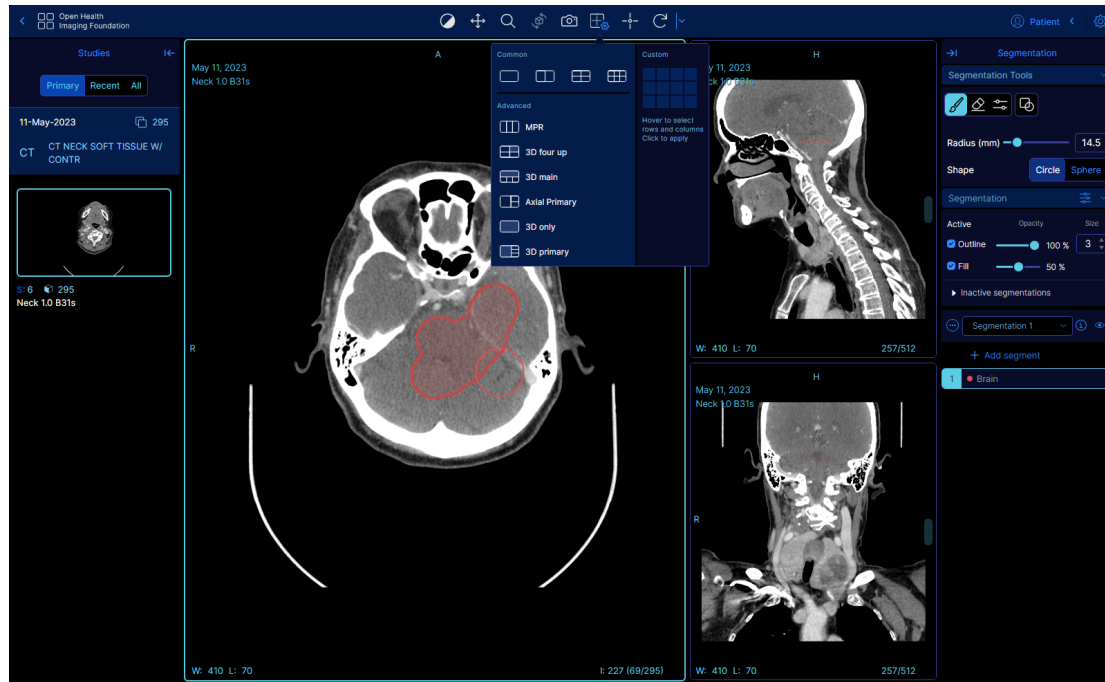
#### Annotation

Annotation, here called segmentation, can be done using a brush tool and an eraser or using a thresholding tool. Further, MONAI label plugin can be added, which adds additional tabs into the segmentation tools pane.

#### Controls

The controls were very simple with the currently selected tool being mapped on left-click. Such tool could be zooming, panning, changing brightness or for example a brush tool for segmentation.





**Figure 5.2.** A sample image with brush annotation with the layouts dropdown open

Control	Action
Scroll wheel	Change slice
Left click (+ drag)	Use currently selected tool
Right click + drag	Zoom
Middle click + drag	Pan

#### 5.2.4 Volume label

Volume label tool was created in a doctoral student's work under the same supervisor. There is a possibility for the EFAN tool project to utilize collaboration during development phase. I have briefly tested the task of tumor annotation in this tool.

##### User interface

It offers a view for each anatomical plane. There are buttons for loading images, saving annotation and annotation itself. The UI is relatively straightforward, with controls on screen and not hidden in pull-down menus.

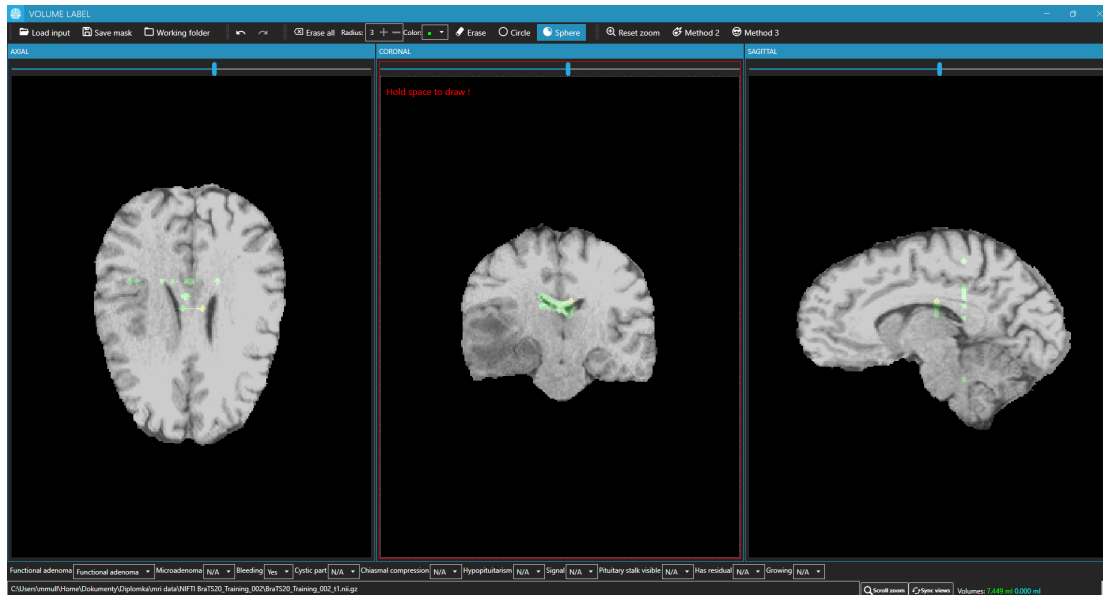
##### Annotation

The software offers circle and sphere brushes and eraser. There are also two buttons that run an AI module to compute a segmentation using different models.

##### Controls

It was relatively intuitive for me to use the software. The toggling nature was however untraditional. I was also puzzled why left click is not used for any action, and it needs to be combined with space to draw. Nevertheless, the GUI conveys this information clearly.





**Figure 5.3.** Volume label

Control	Action
Ctrl	Toggle <b>scroll zoom</b>
Scroll wheel	Change slice
Scroll wheel (scroll zoom on)	Zoom
Shift	Toggle <b>sync views</b>
Mouse pointer (sync views on)	Change slice on adjacent views
Space + Left click + drag	Draw annotation
Right click + drag	Pan

## 5.2.5 Mango

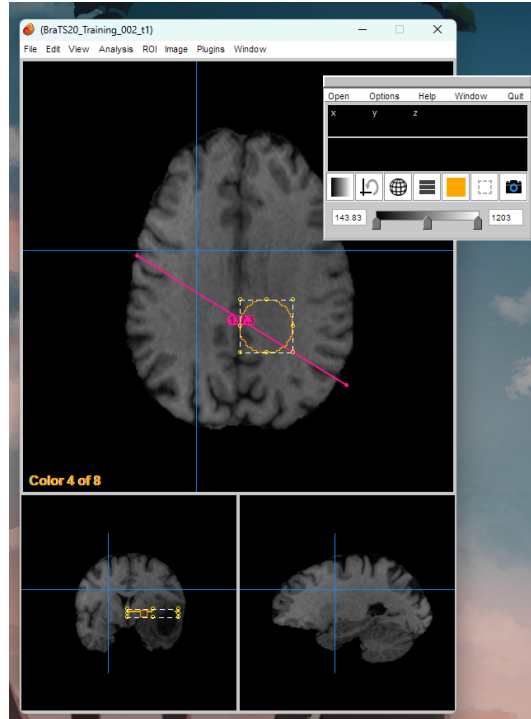
Mango – short for Multi-image Analysis GUI – is a free medical image analysis software by the University of Texas. It is a desktop application designed for viewing, measurement and analysis tasks.

### User interface

The software opens a fixed size window with the main anatomical plane view at the top and two other plane views underneath. A second small floating window offers buttons with image manipulation and ROI creation controls. Icon buttons lack explanations and the GUI is generally hard to understand for a first time user. Many of the software's other actions are hidden in pull-down menus, such as buttons controlling the size of the main window. It took a significant amount of time to learn how to use most of its features.

### Controls

The controls of mango were standing out from other tools. I had a harder time figuring out for example how to pan the view.



**Figure 5.4.** Mango

Control	Action
Ctrl	Toggle <b>scroll zoom</b>
Scroll wheel	Zoom
Left click + drag	Change slice on adjacent views
Right click + drag	Change brightness range
Middle click + drag	Zoom
Shift + Middle click + drag	Pan
Alt + Shift + Left click + drag	Pan

### Annotation

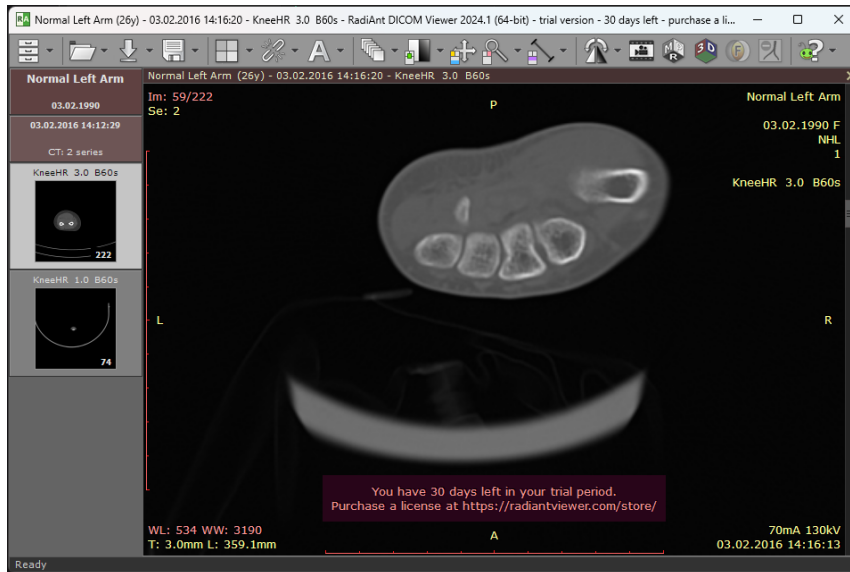
It offers a relatively cumbersome ROI creation experience. Nevertheless, it includes the possibility to edit and refine such ROIs. Additionally, it offers multiple ways to analyze the image — e.g., histogram, measurements for lines and ROI volumes.

### 5.2.6 RadiAnt

RaidAnt PACS DICOM viewer is an example of a paid software. It offers a free trial version, which I briefly explored. It currently lacks support for NIFTI or other image types. Therefore, a DICOM image needs to be used.

### User interface

The interface is clear with icon buttons on top. The main viewing and annotating tools are configurable per mouse button. Such tool buttons show the currently mapped mouse button on a small mouse icon in their corner. This was helpful, nevertheless, I did not understand the blue and purple coloring on those mouse icons. On the left side



**Figure 5.5.** RadiAnt software with a default layout

of the window, a series selector is located. On the right is a view on the current slice. When I ran a “multiplanar reconstruction” function on an image with slices of only one plane, a window with view on all three anatomical planes appeared. It consisted of two small views and a bigger view on the right. A 3D view of the tissue can also be rendered.

### Annotation

RadiAnt only offers measurement and analysis tools.

### Controls

The controls were not too differing from other tools in clinical care and were easy to use.

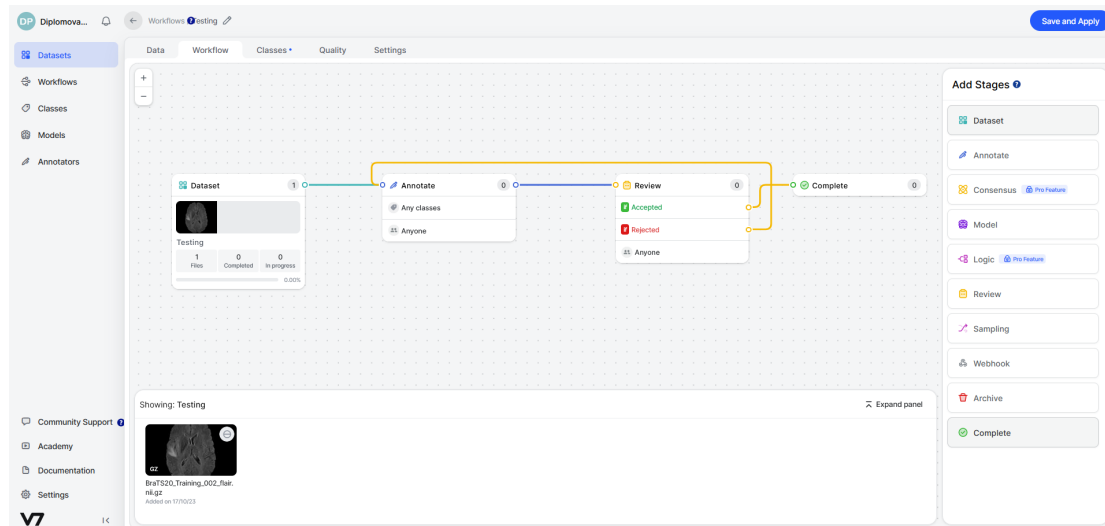
Control	Action
Scroll wheel	Change slice
Left click + drag	Pan
Right click + drag	Zoom
Middle click + drag	Change brightness range

## 5.2.7 V7 Darwin

V7 Darwin is a web tool focused on annotating image datasets for AI applications. This is therefore a promising tool to explore. Since I managed to get access to a demo through my education account, I have been able to test the tool on my own.

### User Interface

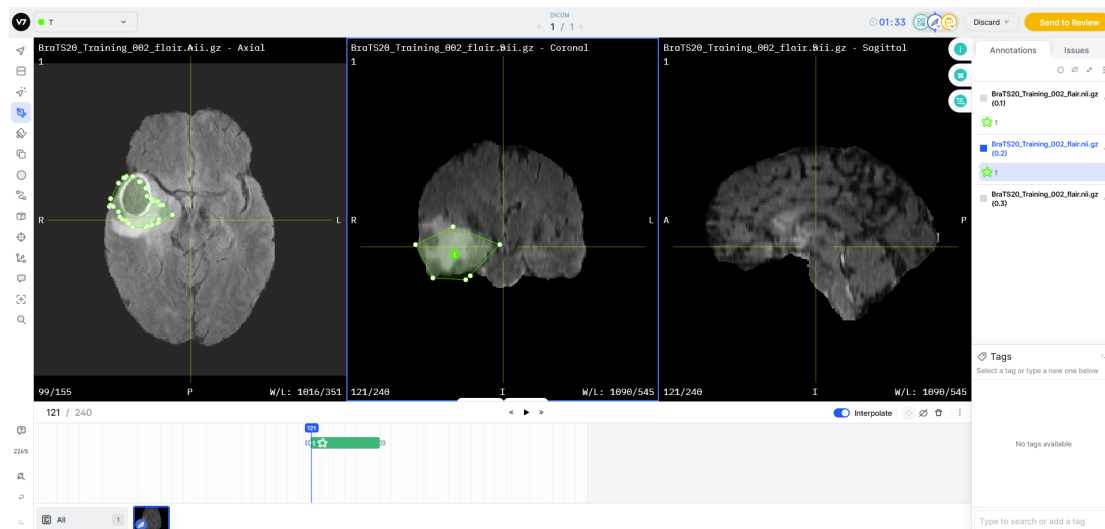
The GUI is intuitive and clean on first looks. The main difference in compared to other inspected software is its logical structure. A workspace has five tabs: Datasets, Workflows, Classes, Models, and Annotators. Datasets are then at its core. Each dataset has an assigned Workflow, annotated Classes and other settings. Workflows are visually designed and define what is being done by who in what order (see fig 5.6



**Figure 5.6.** V7 Darwin workflows

). Models then contain AI models and Annotators provide an overview on people and progress.

During annotation, a toolbar is located on the left side, while annotations and their classes are located on the right side. Changing brightness is possible from a popup settings pane under the settings icon. A unique feature is the timeline at the bottom (see 5.7). The timeline shows the span of annotations through a given slice in a parallel axis. This feature supplements the navigation between slices and allows moving and resizing annotations along that axis. In the top-right corner is a button which submits the annotation for next stage defined in the workflow. The image can also be discarded from the dataset from there.



**Figure 5.7.** V7 Darwin viewer with brain annotation

### Annotation

The following annotation class types are available: Bounding Box, Cuboid, Ellipse, Keypoint, Line, Mask, Polygon, Skeleton, and Tag. The default is Polygon and can be used with for example Brush or Auto-Annotate tool. The Auto-Annotate tool then

allows running a model on a slice by selecting a box ROI. The annotation is then stretched in parallel direction for a certain number of slices. If the ROI is resized on a different slice, the model runs again and adjusts the annotation for that slice. In-between slices are then interpolated. The overall experience was mediocre. I assume this is due to the tool focusing primary on 2D annotation rather 3D.

### Controls

The basic controls are relatively straightforward, utilizing the left mouse button for panning or currently selected tool. I have also experienced a weird behavior when trying to pan the image using the middle button, where the brain shifted away and I was not able to find it again.

Control	Action
Scroll wheel	Change slice
Left click (+ drag)	Use currently selected tool
Right click + drag	Zoom
Middle click + drag	Pan (buggy)

## 5.2.8 Hypotheses and Questions

In this section, I make conclusions and form questions based on the results above.

### Mouse Controls for the EFAN tool

The basic controls significantly differ among tested software. A question arises for the development of EFAN tool — whether to define specific tools for zooming, panning and brightness change to be used with left click and drag, compared to utilizing different mouse buttons for basic viewing capabilities and having left-click tools only for specific tasks, such as measurement and annotation. A third option would be to utilize both combined, such as in OHIF. I present a comparison table between what I have evaluated as the most common mouse controls among open-source and paid software, and controls of three previously introduced clinically used systems.

Action	Common Software	Marie Web-Vision	Syngo.via	Varian Eclipse
Change slice	Scroll	Scroll	Scroll	Scroll
Use selected tool	Left	Assignable	Left	Left
Zoom	Right / Ctrl + Scroll	Left / Ctrl + Scroll	Tool	Ctrl + Scroll
Pan	Middle	Middle	Tool	Middle
Change brightness	Tool	Right	Middle	?

Additionally, Marie WebVision allows binding of individual buttons to tool such as OHIF and RadiAnt. Scrolling for changing the viewed slice is relatively standard, apart from a few exceptions. An important fact for consideration is that the EFAN tool is going to be focused on annotation. Assuming that, the left mouse button — representing a main action — should be primarily utilized for using annotation tools. Zoom, pan and changing brightness remain rather complicated. From those, only panning seems to be mapped on the middle mouse button most of the time.

### Annotation Progress

The introduction of datasets, workflows and different stages allows a more structured management of the whole annotation process.

#### ■ 5.2.9 Quick Mouse Control User Test

I have asked an amateur user without previous experience with neither graphical nor medical imaging software to sit down in front of 3D Slicer and assigned them basic viewing tasks. For zooming, the user intuitively scrolled the mouse wheel. Since this has not worked, they searched for a magnifier icon or +/- signs. For changing the slices, the scroll wheel has been already discovered and therefore used. For panning, first tried was left-clicking and dragging. Next, they searched for a move tool. Lastly, they tried arrow keys.

Such test also hints that a comparison usability test is desirable. Since medical experts are indispensable representants of end users and since I have planned reaching out to such experts again at a later usability testing phase, I have planned the test of this question for that phase as well.

### ■ 5.3 Definition of Personas, Scenarios, and Tasks

Created user personas and scenarios, further expanded by task analysis and use cases definition, were utilized throughout the development process of the prototype.

#### ■ 5.3.1 User Personas

In this section, I would like to mention the personas that I initially described. They are presented as whole user groups with different expertise and knowledge. Their descriptions, following a specification of user needs for each group, can be found in A.2.

#### ■ 5.3.2 User Scenarios

I have created two primary scenarios and two additional user scenarios. The entire scenarios can be found in A.4. I strived to ground the stories in the actual information that I collected from interviews and usability testing of related tools. I further tried to pinpoint problematic situations which could be used as negative examples to avoid in the design of the EFAN tool.

The first scenario focuses on the experience of a radiologist with clinical software in a rather sporadic situation when segmentation is needed. Since I have not previously interviewed an expert in a dataset annotation situation, but rather with tasks in their profession in mind, the described task is also part of his profession. The experts hired for annotations are expected to come from similar backgrounds, therefore possibly having similar previous experiences with medical imaging software.

The second scenario covers a short story about a radiologist intern, briefly illustrating a job of a less experienced medical specialist. It takes on further information from the radiology interview and demonstrates a different radiology task. It points out, that even professional medical software can have usability problems. For this scenario, I have not described the completion of the task itself due to time constraints and rather focused on the next scenario.

The third scenario finally focuses on dataset annotation. It is a story of a student who volunteered as an annotator. Since I have previously tested the usability of 3D

Slicer, the scenario therefore takes on the results of this usability test. It describes multiple problematic situations in detail.

The fourth scenario was planned to be about an oncologist creating contours for further use of radiotherapy treatment planning. Due to time constraints, the availability of other scenarios, and shift of focus rather on annotators of datasets, it was cancelled.

### ■ 5.3.3 Task Analysis

I originally planned on creating an HTA for the prototype, but I misunderstood its principles and created a description which rather resembled an ideal application pass at first. The description of an annotation task in an ideal pass can be found in the appendix A.5. Even though this was not ideal, it helped me get a rough idea how the final prototype might work. Since I have further outlined use cases and tasks inside notes of the prototype's draft, I have not followed on a true HTA.

I have collected user data primarily about image segmentation itself. Collecting information on dataset annotation was rather difficult task. For that, I have used discussions with my supervisor, researching related works, and exploring dataset annotation software.

## ■ 5.4 Prototype Development

This section describes the development process of the prototype for the EFAN tool until the first usability testing in a structured and reasonably chronological way. The iterative prototyping process described here was informed by qualitative interviews with medical specialists, encapsulating their insights and pain points regarding tools in medical practice. The development was further informed through preliminary testing of related tools. The chapter consists of three major stages of the development, before the first usability tests were possible to be conducted.

### ■ 5.4.1 Early Design

This stage covers numerous drafts and specification made for the prototype.

#### Determining Intended Platform

While the preceding chapters laid a solid groundwork for initiating the development of the prototype, another aspect remained for brief consideration: the platform on which the final system will operate. Non-questioned was that I am crafting an application tailored for use in a desktop environment. The question was therefore whether to design for a desktop application or a web-based application. If it were a desktop application, then an intended operating system should be also considered. I have not come to a decisive conclusion regarding this matter, so I let the final decision for future developers. However, I have discovered that participants from interviews were using Windows as an operating system in their jobs. If we add the fact that Windows is the most commonly used system worldwide [52] and that many annotators might therefore have a good knowledge of the system as well, we can assume that designing the app with Windows users in mind is a reasonable approach. I have followed on this idea primarily when designing the visual aspect of the prototype.

#### Design Specifications

I have started by following gathered insights and hypotheses from user research to write down a rough design specification for the tool. It was based on the use cases



and requirements discovered in the qualitative interviews from radiology and oncology departments. The full draft can be found in B.1. After goals of the work have been further clarified, a second draft with additional information and adjusted use cases followed and can be also found here B.2.

### Key Functional Aspects

The design in this phase consisted of two main screens — a file selection screen and a viewer screen.

#### File selection screen

The file selection screen serves as a streamlined way to choose an image for annotation. Commercial software in medical practice as previously researched utilize a full picture archiving and communication system (PACS). Development of such system would require a significant amount of additional effort and is out of the scope of the current work. Moreover, such system uses workflow based on patients and individual studies, where related scans are grouped together, which is an unnecessary approach for dataset annotation. Therefore, a simple file browser screen, allowing to select and annotate specific files, was proposed.

#### Viewer screen

The viewer screen then offers the main functionality for working with medical images. Since the goal for this work is designing a tool that respects common usage patterns from commercial practice, so that medical experts can utilize their experience with existing software, I have based the viewer's basic layout on such software. I have further refined this decision by taking into account information gathered in the interviews. The participants often look at only one view at a time, and the other are used less frequently. The most essential view for medical experts for analysis is the transverse plane. Taking this information into consideration, the prototype's screen shows three views of the same image by default. The primary view shows the transverse plane on the left and two smaller views of the other planes on the right on top of each other. The views have a color-coded border, similar to Varian's image viewing software: blue for transverse plane, green for coronal plane and red for sagittal plane. The layout will be customizable, with the option to maximize only one of the views and keep other hidden.

### Considered Use Cases

In the second specification notes, I have considered the following use cases for the tool:

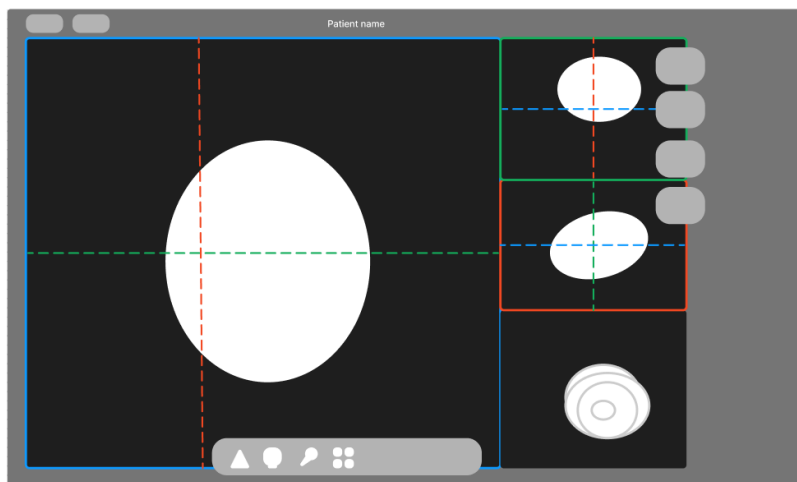
- Definition of annotation classes for a dataset
  - class names
  - class colors
- Annotation of a whole folder with defined classes
- Annotation of individual file with defined classes
- Annotation of individual file from scratch (including class creation)
- Viewing of annotated file
- Viewing unannotated file

These use cases considered a design, where annotated classes could be predefined for a folder to allow easy handover of data from researchers to annotators. Such folder would also contain a file with classes for annotation and the images to be annotated.

I have further discussed real-world dataset annotation examples with my supervisor and decided on a representative task to be tested through the prototype: annotation of a brain tumor on a MRI image.

### FigJam Drafts

The first prototype draft was made using FigJam, where I also created sticker notes with use cases and tasks. Links and previews of all three FigJam versions can be found in appendix B.3. On figure 5.1 I present the core part — the viewer screen GUI draft. It was inspired by medical software used in practice while adding the possibility for a 3D model view, aligned on the right under other non-primary views. An interesting aspect that seemed helpful in navigation through the slices are the dashed lines, indicating current slice positions of other views. I have illustrated two ways of how to present available tools and settings to the user. On the right is a tool pane inspired by Varian Eclipse. A pane on the right is also typical for V7 Darwin and OHIF. The bottom toolbar with annotation tools was inspired by FigJam’s toolbar itself. Through positive personal experience with FigJam’s toolbar, I proposed to test such modern way of picking annotation tools.



**Figure 5.1.** FigJam draft outlining the viewer layout

### Paper Sketches

I have created numerous notes and sketches on paper during the process. On figure 5.2 I present the most important one, at which I tried out and personally tested different layouts of views, panes, and tools. The remaining sketches, including a few additional GUI elements, can be found in appendix B.4.

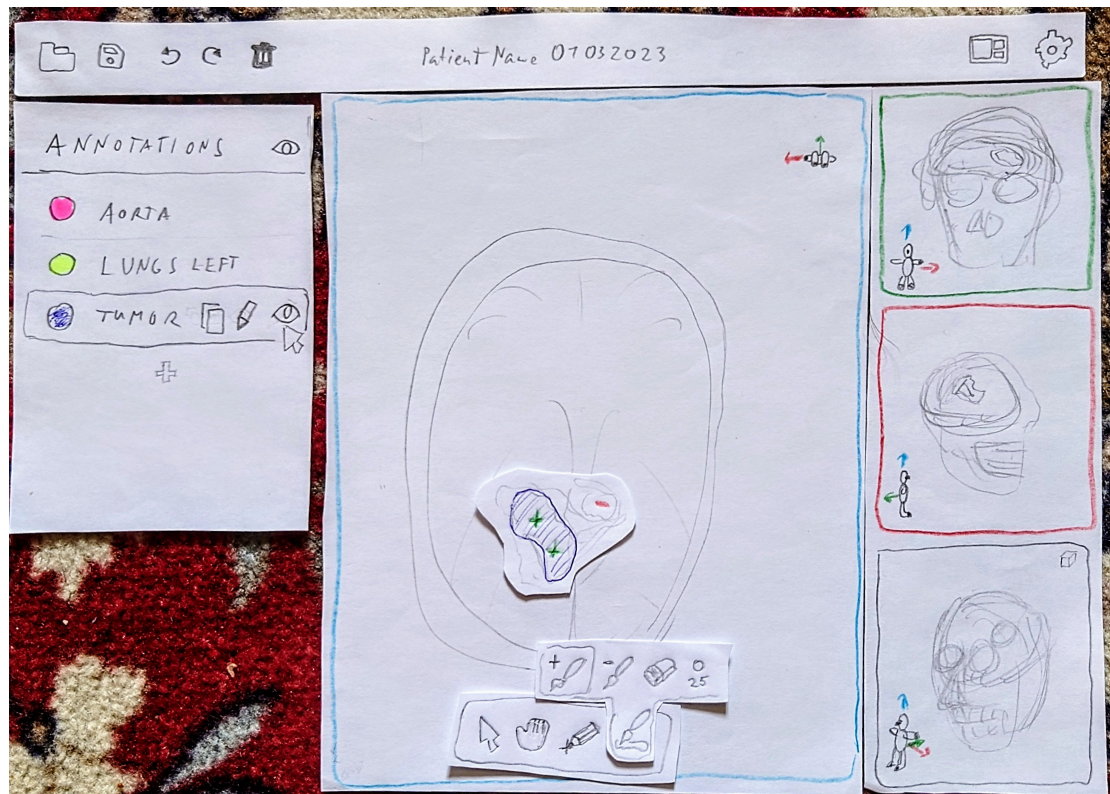
### Proposed Annotation Methods

For the first design, I have proposed the following four annotation methods / tools:

- Points (specify inside/outside)
- Edge points (specify edges)
- Scribbles (specify inside/outside)
- Lasso (specify edges)
- Brush 2D/3D (specify inside)

Such annotation tools were designed with a single annotated structure and its volume in mind. The methods are described in detail including images in appendix B.5.

I have also originally thought of providing the annotation tools for the user as a choice during the annotation process. This later turned out to be rather biased idea, since the need is to create an application for the development of AI applications. These



**Figure 5.2.** Paper sketch; Annotation pane on the left, main plane view in the middle with annotation tools floating

applications will determine the methods desirable for specific dataset annotation. The choice of annotation methods should therefore be determined rather per dataset or per image.

### Figma Interaction Limitations

Nevertheless, during my exploration of using Figma for the development of a lo-fi prototype, I have discovered an issue. Figma lacked the possibilities for introducing necessary interactions for simulating image annotation. Even placing visual markers as points on an image seemed as a difficult task, not mentioning the unavailability to produce the effect of scrolling between individual slices of the 3D volume. I have therefore paused the development of the prototype in Figma and moved on to Axure RP. According to my research of feature, Axure RP seemed as a better software to achieve testable annotation interaction.

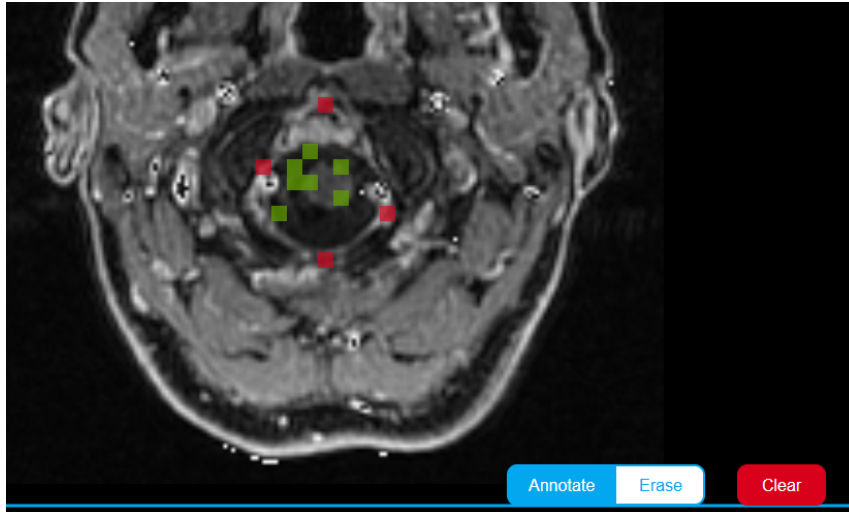
#### 5.4.2 An Attempt on Axure RP

After switching to the development of an Axure RP prototype, I have set the goal to first implement the point annotation tool interaction and move on from that.

This was challenging even in Axure RP, since reacting to events from the scroll wheel can only be registered when actual vertical scrolling happens. I have found a way to hack this behavior and implemented changing slices by allowing the whole view scroll by a negligible amount. Slices were imported as individual images and assigned to individual states of a dynamic component. Each of these states contained a grid repeater with checkboxes, which allowed to be checked, unchecked or disabled. I have then changed the styles of such checkboxes to be either green, transparent or red, respectively.

This way, I achieved an interactive prototype with the following features:

- changing slices using scroll wheel
- adding positive and negative annotation points by left and right clicks
- erasing points one by one or all at once



**Figure 5.3.** Point annotations in Axure RP prototype

This implementation, nevertheless, had a substantial issue — significantly low performance and expandability. The issue lied in the use of many states with many checkboxes inside those repeaters. Even after reducing the number of interactive states and checkboxes, running the prototype required over a minute of waiting to load and navigation between slices was not smooth. This would not be an issue preventing testing of the prototype as it was, but it was an issue for further expansion of other annotation methods, since working with the prototype in Axure RP became very slow.

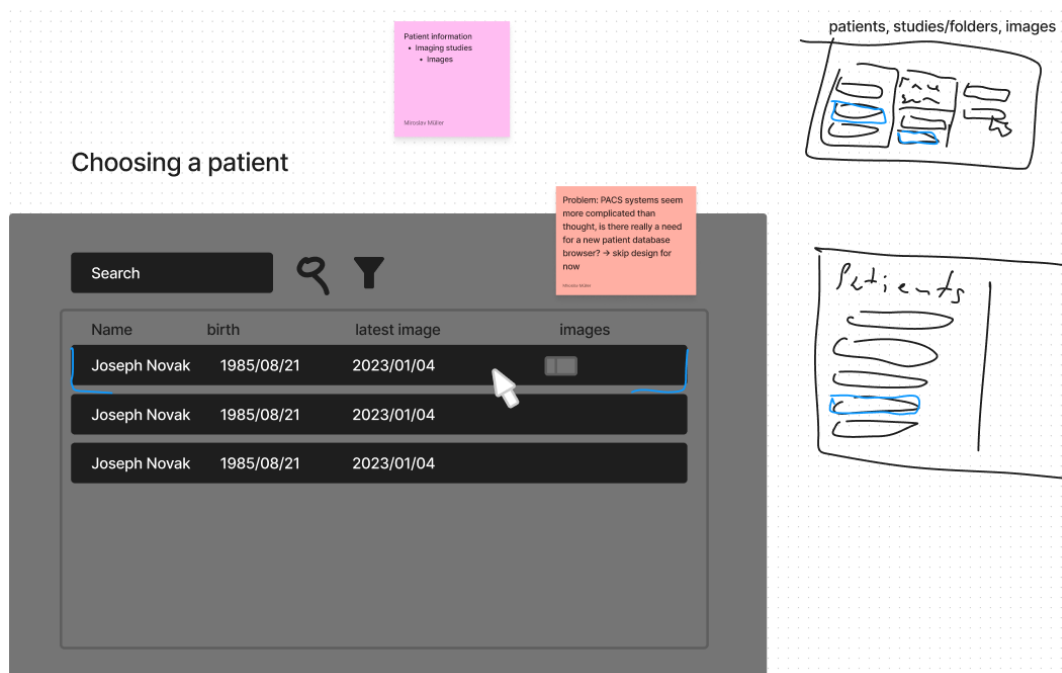
I assumed testing only point annotation interaction on its own does not answer any questions about what is and is not efficient and usable. I have decided to not continue with attempts to implement annotation interaction through prototyping tools, and rather focus on evaluating the overall dataset annotation process and usability of the prototype's GUI. Comparison testing of different annotations was planned for high-fidelity stage, where manual programming is used to create a simple testable application. I have tried coming back to this prototype a few times, but was rather discouraged to use it due to the technical difficulties it posed.

### ■ 5.4.3 Return to Figma

Since the attempt on developing a testable annotation interaction in Axure RP failed, I have decided to return to Figma, where I have previously started designing the application. Reasons for using Figma over Axure RP were more experience and better previous experience in prototyping. This section describes the development of the prototype until the first usability testing.

#### Developing File Selection

I have also briefly returned to previously mentioned FigJam Drafts in section B.3 and focused on drafting a file selection screen. It was at this time when I finally decided



**Figure 5.4.** Draft of a PACS-like screen in FigJam

not to design a PACS-like patient based system, but rather a simpler file based system, which could function either locally or on a remote server.

### Revised Functional Specification

At this stage, I once more revisited related works and results from user research. Assuming we want to allow the development of AI applications for radiology in general, annotating more than one class per image is desirable. Therefore, I have set the goal to implement the ability to annotate more than one class in an image.

I have further elaborated on the use case of a whole dataset annotation. I have brainstormed the logical structure of the EFAN tool and proposed an updated design through a draft functional specification in Diagrams.net, including annotation state flow diagram, GUI layout drafts for annotation screen, popup window for choosing the annotating method, and a proposal of a file picker available directly from the viewer.

The functional specification and annotation flow pages can be seen in appendix B.7.

### Final Logical Structure

Following on the functional specification, I have settled on the following top-level logical structure in the end.

Entities and their attributes:

- Folder (Dataset)
  - List of Image Files
- Image File
  - Annotation (optional)
- Annotation
  - Current Status
  - Annotation Method
  - Annotated Classes



- Class Preset
  - Name
  - List of Classes
- Class
  - Name
  - Color
- Annotation Method
  - Name
  - Description

**Folder (Dataset)** - This unit represents a folder (or dataset) that contains files for annotation, since researchers will need a way to pass down image files for annotation. This use case requires more user research from dataset annotation practice. Nevertheless, a folder is assumed sufficient for now. The actual implementation is rather a technical detail.

**Image File** - Starting from the image files, each file can have an assigned annotation. In the real world, this could be for example represented by a separate file with a different extension or by attributes directly implemented into existing formats such as NIFTI. Nevertheless, I would not like to make conclusions about implementation details here and keep the design only from what the user will see and work with.

**Annotation** - Each annotation has a status, an assigned annotation method based on the AI model used, and assigned classes for annotation. It also contains all annotation data, such as marked points. An annotation can be created and deleted per file. Empty annotations with only status (and optionally annotation method and classes) can be also created beforehand, therefore enabling marking of specific files needing annotation or revision.

**Class Preset** - A class preset is a reusable unit of a set of classes. Class presets should be stored externally and allow to be reused for different datasets. It should be also possible to share presets between researchers and annotators.

**Class** - A class represents a type of tissue being annotated. Classes can represent for example tumors, lungs, pulmonary nodules, or bones. Each class has therefore a name and a color of annotation, for easy distinguishing.

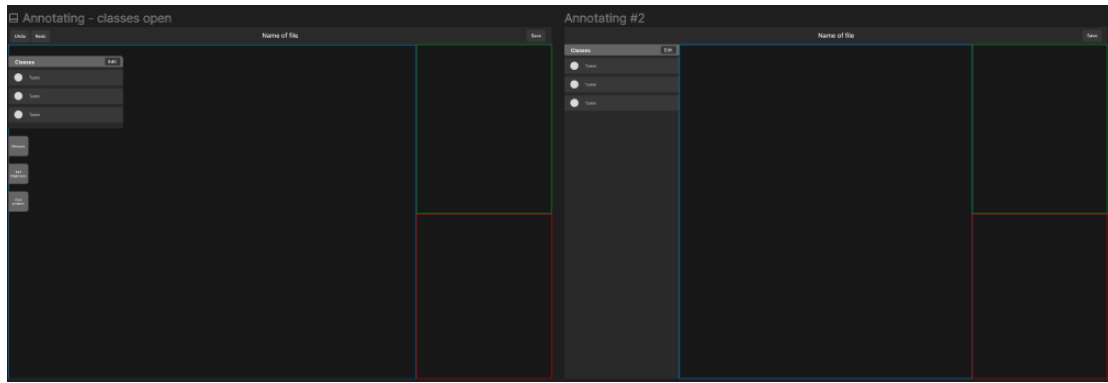
**Annotation Method** - Annotation methods correspond to tools available for annotation. They should also exist externally, such as class presets. Currently, it is not certain, whether and how annotation methods could be changed, added or removed. As previously mentioned, I propose a set of six annotation methods for this prototype (counting Brush 2D/3D as two).

**Status** - An annotation can take on a status. I suggest three default statuses for now, with the possibility to allow user-created statuses in the future. The current statuses are:

- *To be done* — marks a file that has yet to be annotated
- *In progress* — marks a file where annotation has begun
- *Complete* — marks a file where annotation is complete

### Annotation Tools Pane

Fitting lists of annotation classes into a popup menu from a floating toolbar located in the middle thus seemed unsuitable. I have therefore considered two options for incorporating annotated class selection.



**Figure 5.5.** Left - Floating tool pane; Right - Full height tool pane

I have decided to first follow with the floating approach, as shown on the left of figure 5.5 . Assuming the main image view is the central point to the annotation, reducing the size of other GUI elements is desirable.

I have then continued with the prototype development, basing it on ideas from the mentioned functional specification.

### Encountered Figma Problems

The variables feature in Figma is seemingly a new feature, sometimes still resulting in visual glitches and unstable behavior. When I combined it with component variants and interactions, for example, hovered buttons often got stuck in the hovered state. I have spent time on trying to fix these glitches before usability testing, but I was not successful in all.

### Prototype Ready For Usability Test

Here I present a preview of the final design of the prototype before usability testing was conducted. Screenshots of all parts in more detail can be found in the appendix C

### Main Menu and File Selection Screen

I have introduced the concept of main menu into the prototype. The main menu consists of four tabs, selectable in the left pane: Files, Annotation classes, Annotation methods, and Settings.

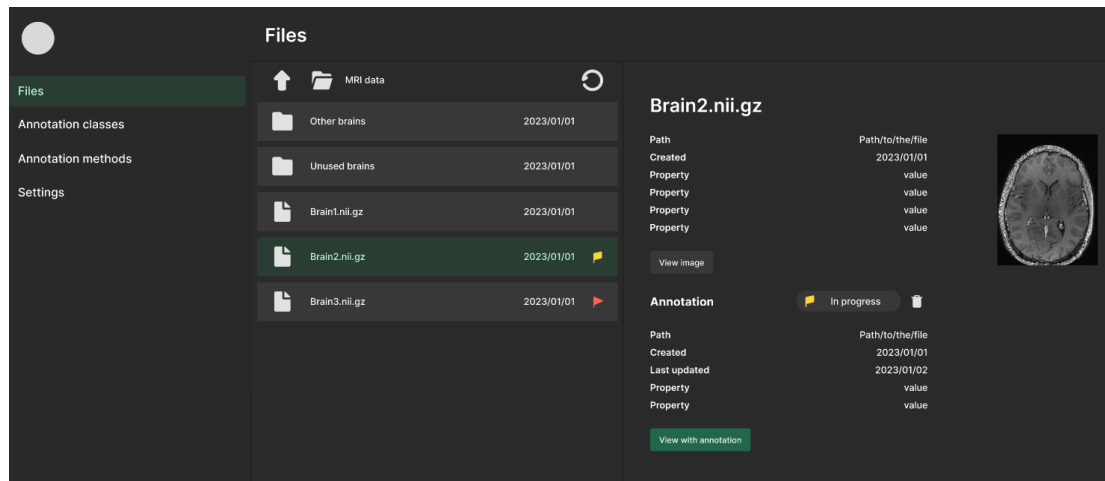
The middle pane then represents the currently open tab. The optional right pane then shows detail information about currently selected item. A screenshot can be seen on 5.6 .

If a user selects a file without assigned annotation, “View and create annotation” button can be pressed. A cancelable popup then appears, asking for selection of a class preset to be annotated. After confirmation, next popup appears, asking for choosing an annotation method. The methods can be further adjusted from there. An example is the setting a fixed number of annotated points. This feature is a result of a discussion with my supervisor, from whom I got the information that a currently developed AI application by a different student considered using only a fixed number of points as input. Finally, the user is presented with the viewer screen, ready for annotation.

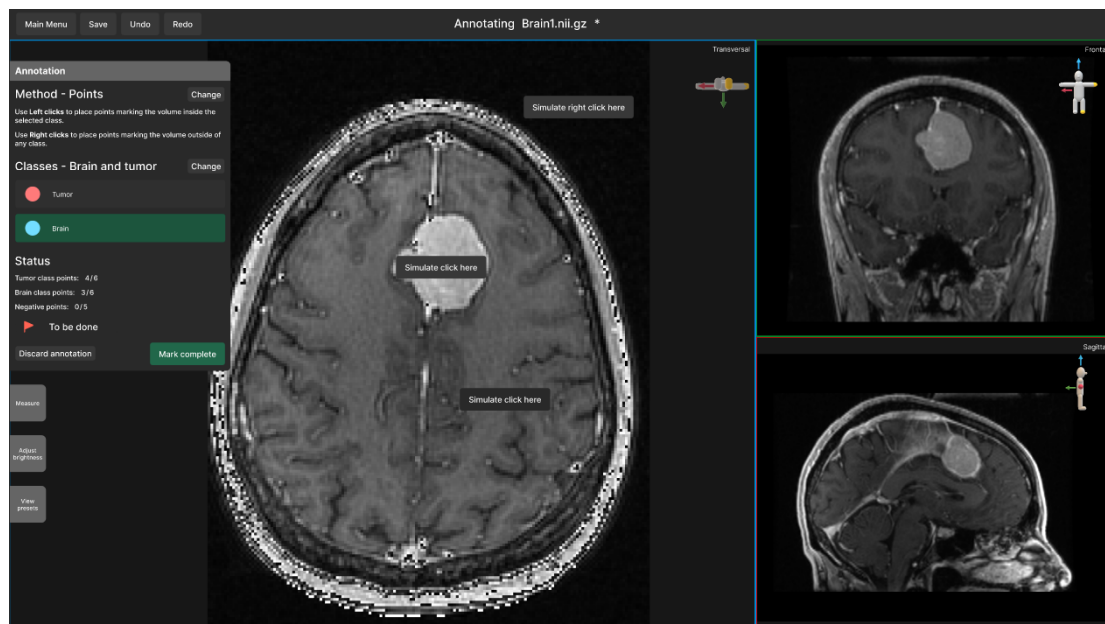
### Viewer Screen

The viewer screen follows the previous drafts. It consists of default three-view layout with floating tools on the left. At the top, we can find a bar with the file name and options to go to main menu, save the annotation, undo, and redo. A screenshot can be seen on 5.7 .





**Figure 5.6.** File selection screen



**Figure 5.7.** Viewer screen with annotation tool pane open; A single brain slice is shown; Buttons that simulate point annotation are located on the annotated image

If no annotation is assigned, then the annotation pane will be collapsed by default and the title will say “Viewing filename” instead of “Annotating filename”. The title further shows a star behind the filename if unsaved changes have been made to an annotation. An annotation can be created by unwrapping the annotation tool pane and clicking a “Create annotation” button.

### Annotation Tool Pane

The pane is composed of the selected annotation method, annotated classes and its current status. A description for the method is shown here, and it can be changed from here. Classes can be edited as well. A single class is always selected from the list of currently annotated classes. Lastly, the status shows the current number of placed points and the annotation status, also discardable or changeable from here. The section also contains a primary button with “Mark complete” on it. This changes the status to complete and changes to a non-primary button with “Go to next file to be done” for

easy access to the next file, without needing to go to the main menu. The next file is a following file with a status other than Complete.

### Simulated Annotation

Since no actual annotation interaction was implemented, I have provided the user with one, two or three buttons on the image, depending on the configuration. Clicking on a button simulates a left click with the selected class, adding a number to the corresponding class in the status section.

## 5.5 Usability Testing

This section describes how the interactive prototype underwent rigorous usability testing sessions. Feedback from these sessions was analyzed and used to refine the prototype further, ensuring that the final design meets the users' requirements and expectations.

### 5.5.1 Usability Test Plan

Through discussion with my supervisor, I have initially outlined the tests as follows:

1. Pilot test (Figma + Axure RP)
  - Figma Tasks:
    - Walk through the program for a single image (no real annotation)
    - Walk through the program for multiple images (no real annotation)
  - Axure RP Tasks:
    - Use the Point annotation tool to annotate a tumor (only + points)
    - Use the Point annotation tool to annotate a tumor (both + and - points)
2. Updated test (Figma + ?)
  - To be written based on the results of the first test
3. Comparison and performance test (Figma with embedded web prototype or other software)
  - Prototype Tasks:
    - Annotate using following methods:
      - Points
      - Edge points
      - Lasso
      - Brush
      - 3D Brush
  - 3D slicer
    - Same tasks using similar tools

### 5.5.2 First Usability Test

The goals of the first test were to evaluate whether a basic PC user can intuitively understand the GUI, navigate through the GUI, open image files, manage already annotated files, and complete multiple annotation tasks in succession. In summary, I aimed to determine if amateur users can easily learn and understand how to use the tool. Additionally, the test sought to identify weak and problematic spots in the overall process to enhance usability and efficiency. The test scenario and testing document can be found in appendix D.

### Pilot Test

I have first conducted a pilot test to evaluate the testing process itself. The test showed a few minor bugs in the prototype, which I fixed and which helped me better prepare for the next usability tests. Since the test itself was not significantly changed and the results contained a substantial amount of useful information, I have included the results from this test in the final evaluation.

### Evaluating Test Results

From each test, I have made a screen recording including audio. All participants agreed to be recorded. For each test, I have created a file containing information on: the testing environment, participant's profile, test log including answers to questions. After the test, I have watched the recording and filled in the test log. After the test log was complete, I have added positive and negative findings sections. I followed with evaluating valuable findings and insights from the test log and fill them as bullet points into those sections.

Finally, I have made a Findings summary document, containing a long table with all unique insights. Each insight was assigned the following additional information:

- Actionable insight
- Users — from how many (and which) users this insight came from
- Severity — how severe do I interpret this insight to be
- Implication for Prototype Revision — proposed changes in the next version of the prototype

The full table can be found in appendix D.0.8.

## 5.6 Prototype Improvement

In this section, I describe how I further improved the prototype after the usability tests, and describe next steps that I took to prepare the prototype for next tests.

### 5.6.1 Improvements Based on Test Results

The usability test provided much more information than expected. After going through previously mentioned findings, I have started making changes in the prototype.

#### Renaming Classes to Structures

The users were confused between classes and class presets. I have therefore decided to revisit the topic of naming of class presets and classes.

After reviewing results from medical interviews, I found that there are other differences in naming conventions apart from the segmentation process being called contouring in medical practice. A good example from the Varian Eclipse software, which is focused on contouring, is that the segmentation classes are rather called “structures”. I assume this is due to the nature of what is being segmented — body structures and tissues. Since our tool is going to be also used by medical experts and only for medical image segmentation, I conclude that using a name corresponding to their practice is more suitable. Therefore, the annotation classes were rather renamed to structures in the EFAN tool.

I have then focused on adding clear indication on what is meant by a structure preset and a structure in the GUI. I have improved the popup for selecting structures for annotation by adding clear indication of presets and structures, and slightly reformulating

the title. I have further added small color dots to each preset, indicating structures does it define.

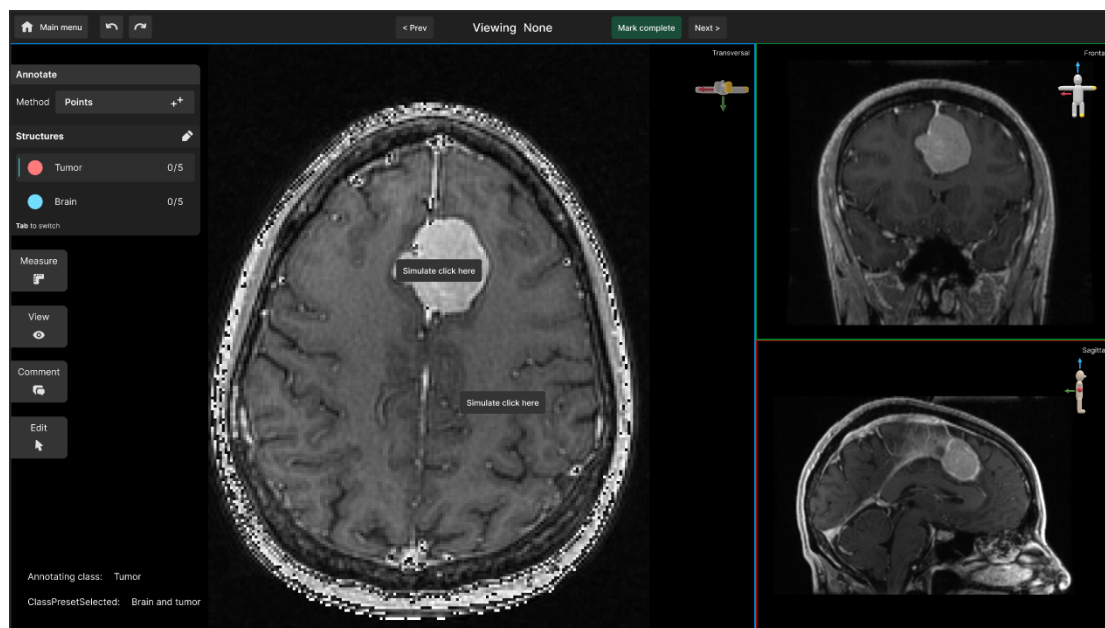
### List of Changes

Here I list further realized changes in the prototype in a compact format:

- manual saving was cancelled, autosave after each change is assumed
- “View” and “View with annotation” was merged into one button “Open Image”
- added icons to tools and buttons
- reworked both popup windows for clarity
- updated feedback on what item from a list is selected
- “Mark complete” button and a status flag were moved to the top bar next to the file name
- annotation tool pane was made more clean
- using a keyboard shortcut for easy switching between classes was added, including a hint in the annotation pane

### Final Design Showcase

The whole final design can be found near the end of this work in appendix E. For illustration of changes, I showcase the annotation screen on figure 5.1 .



**Figure 5.1.** Updated viewer screen with annotation pane open

### ■ 5.6.2 Export from Figma

At this phase, I have already reached the limitations on what is possible with Figma in comparison to what is desired to be tested. I have thus decided to move from Figma to code simple implementation. This step needed an export from Figma, which, contrary to my expectations, turned out to be a time-consuming and difficult task.

I had to decide on the export plugin, the export language and library. After thorough consideration, I have chosen to export the prototype as an HTML website built on React.

## React

React is a free and open-source front-end JavaScript library for building user interfaces based on components. It is focused on creating Single Page Apps (SPA's). In SPA's, the server only returns a single HTML page to the client. Navigation within that page is then handled by React running in the browser.

The export plugin from Figma was able to output HTML, CSS, and even React files with components. Nevertheless, no currently available plugin that I explored was able to output code with implemented interactions. The whole interactive functionality would have to be coded from scratch. Moreover, the export from the free plugin I used was rather based on exporting frames and not the entire application.

Even though there was not much time for this work remaining, exporting the prototype was still considered better than building the app from scratch. For the purpose of this work, I learned the basics of React from a comprehensive playlist of video tutorials [53]. I have then followed with the export itself. For interested readers, an external appendix file called `React notes.md` contains my notes from when I learned React and started exporting the prototype.

Nevertheless, at this point, I have run out of available time for the practical part of this work. It therefore marks the end of my development of the prototype.

# Chapter 6

## Discussion

### 6.1 Literature and Software Research

I have thoroughly investigated works and software related to segmentation of medical images. Nevertheless, the count of related software was rather overwhelming, so it was not possible to deeply analyze everything. MITK and Nvidia's autosegmentation [21], which I discovered in a later phase of the work, remained as a non-explored yet promising work.

### 6.2 Qualitative Interviews

In this section, I attempt to evaluate the quality of conducted interviews.

#### 6.2.1 Validity

During qualitative interviews, I interviewed two medical experts while focusing on gathering information through open questions and investigating the characteristics of their work with current medical imaging software. Since both of them were of significant position and had years of experience, they could additionally provide information about their whole workplace and mediated information about other related job positions as well. This served as a solid basis for understanding how commercial software is used in practice.

Nevertheless, due to time capacity reasons and misunderstanding previously mentioned in section 5.1.3, I have not conducted interviews with actual hired annotators of datasets — neither expert annotators nor amateurs. The collected data about them are only through many consultations with my supervisor, exploring related works, and dataset annotation software. Moreover, the interviews were conducted with the EFAN tool being an end product for medical experts in mind. The discovery of this problem made some needs and use cases formed from those interviews, such as for measurement, irrelevant, since they were related to regular tasks of medical experts in their profession, rather than in the situation of dataset annotation. This complicated the interpretation of interviews, since I had to manually select information relevant only to dataset annotation.

#### 6.2.2 Reliability

The two participants from qualitative interviews provided plenty of information to work with. Completely avoiding the risk that the collected data were subjective to those participants and that they would not apply to other potential participants would cost additional effort, which was evaluated to be unnecessary. Nielsen [54] argues that the relation between the number of tested users and found usability problems is logarithmic. The author therefore argues it is crucial to have at least some participants as opposed to zero, and that further raising of the number provide only less and less beneficial data.

### ■ 6.2.3 Systematic Error

During qualitative interviews, I have utilized some aspects of the field research method. Babbie [35], as cited in [36], mentions the risk of impact of the presence of the researcher on the participants during field research. The subjects might have altered their behavior due to my presence, possibly more due to the combination of an interview with observation. Since I let the subjects show me how they work with the software on their own, the subjects might have chosen examples what they think are worthy of showing, but not really representative. As for the interview part, I have focused on posing open explorative questions without using any suggestive questions.

## ■ 6.3 Usability Testing

Usability testing was conducted to collect insights about basic intuitiveness of the designed tool.

### ■ 6.3.1 Introducing Quantifiable Metrics

Although planned, the prototype did not get into a phase where I was able to properly utilize quantifiable metrics other than counting for how many tested users a specified actionable insight occurred. Measuring the time of test tasks and counting of errors could further help quantify test results. Using such quantification, a comprehensive comparison of the proposed annotation methods could be achieved.

Utilizing an eye tracker was also recommended to me by my advisor. Nevertheless, I did not find the prototype to be in a situation where the benefits of such measurements would overcome the associated time-cost yet.

## ■ 6.4 Further Prototype Development

In this section, I make suggestions based on the current state of development for further researchers to consider.

### ■ 6.4.1 Visual Design

I have proposed a visual style reminding of modern WinUI 3 apps. If the proposed prototype's styling matches further development criteria, I recommend following Microsoft's Fluent 2 design [55]. The website covers main principles of the design language. When Windows is selected as platform, it provides a link to an elaborate Windows UI 3 Figma UI Kit, which I only discovered late when writing this work.

### ■ 6.4.2 Suggested Features

For lasso and other scribble-like annotation, I suggest defining an easy way to rotate the image is possible, e.g., by holding a modifier key while clicking and dragging with a mouse button. The rotation feature was used in the HistomicsUI and in the work by Amgad et al. [6] and is also a common feature among graphical software, where strokes need to be drawn with mouse.

I also strongly recommend realizing a usability test on mouse controls and keyboard shortcuts, since this is a key part of the interaction.



### ■ 6.4.3 Guidelines for Human-AI interaction design

In the beginning, I intended to utilize the Guidelines for Human-AI interaction design by Microsoft [56]. They are a set of 18 generally applicable best practices for designing human-interaction with AI-based products and features. The Guidelines are divided into four groups, based roughly on when they are the most relevant to interaction with an AI system: upon initial interaction, during interaction, when the AI system is inevitably wrong, and over time. The guidelines are part of Microsoft's Human-AI eXperience (HAX) toolkit.

Nevertheless, since I was designing the tool without a human-AI loop incorporated yet and rather focused on general usability, there was no direct need for utilizing those guidelines or toolkit. Therefore, I have only read through the guidelines and recommend following on the toolkit once the development integrates a human-AI interaction loop.

### ■ 6.4.4 Utilizing Interactive Segmentation

Similar to the paragraph above, I believe utilizing a true interactive segmentation should be a priority in development following this work. The current prototype supported only non-interactive UI. Nevertheless, the proposed annotation methods are laid down such as iterative and interactive segmentation should be made possible.

## ■ 6.5 Conclusion

The development of a user-oriented medical imaging data annotation tool for AI applications in radiology is an important task for enhancing the efficiency and accuracy of image segmentation processes. This work has addressed the challenges of current annotation tools by laying out the UX design foundation for a new annotation tool through a user-centered approach.

I had overcome many technical, organizational and personal struggles during the whole process, and learned many valuable lessons. Such struggles and the need for perfection of my designs lead to a slower pace in the entire process, which prevented me from being able to conduct additional usability and efficiency tests. However, the foundational work completed thus far provides a strong basis for subsequent testing and refinements in the design.

In this work, I have namely: researched the topics of medical image segmentation and dataset annotation; surveyed existing medical image viewing and annotation tools; reviewed general and AI-specialized user-centered principles and methods; conducted careful user research among target users; defined user personas, scenarios, and requirements needed for the development of a prototype; iteratively improved the prototype's design from paper sketches up to a solid interactive prototype in Figma; conducted usability testing of the prototype; further the prototype while preparing it for next usability tests.

The results of this work provide a valuable UX framework for future development of the EFficient ANnotation tool, serving as a user-friendly 3D medical image annotation tool, accelerating AI development in radiology.

## References

- [1] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment Anything in Medical Images. 15 (1), 654. DOI 10.1038/s41467-024-44824-z.
- [2] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. *Image Segmentation Using Deep Learning: A Survey*. <http://arxiv.org/abs/2001.05566>.
- [3] *Image Segmentation — 3D Slicer Documentation*. [https://slicer.readthedocs.io/en/latest/user\\_guide/image\\_segmentation.html](https://slicer.readthedocs.io/en/latest/user_guide/image_segmentation.html).
- [4] *Medical Imaging*. [https://en.wikipedia.org/wiki/Medical\\_imaging](https://en.wikipedia.org/wiki/Medical_imaging).
- [5] Andres Diaz-Pinto, Sachidanand Alle, Vishwesh Nath, Yucheng Tang, Alvin Ih-sani, Muhammad Asad, Fernando Pérez-García, Pritesh Mehta, Wenqi Li, Mona Flores, Holger R. Roth, Tom Vercauteren, Daguang Xu, Prerna Dogra, Sebastien Ourselin, Andrew Feng, and M. Jorge Cardoso. *MONAI Label A Framework for AI-assisted Interactive Labeling of 3D Medical Images*. <http://arxiv.org/abs/2203.12362>.
- [6] Mohamed Amgad, Lamees A. Atteya, Hagar Hussein, Kareem Hosny Mohammed, Ehab Hafiz, Maha A. T. Elsebaie, and et al.. NuCLS: A Scalable Crowdsourcing Approach and Dataset for Nucleus Classification and Segmentation in Breast Cancer. 11 giac037. DOI 10.1093/gigascience/giac037.
- [7] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lenczi, Elizabeth Gerstner, Marc-André Weber, and et al.. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). 34 (10), 1993–2024. DOI 10.1109/TMI.2014.2377694.
- [8] Risheng Wang, Tao Lei, Ruixia Cui, Bingtao Zhang, Hongying Meng, and Asoke K. Nandi. Medical Image Segmentation Using Deep Learning: A Survey. 16 (5), 1243–1267. DOI 10.1049/ipr2.12419.
- [9] Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N. Chiang, Zhihao Wu, and Xiaowei Ding. Embracing Imperfect Datasets: A Review of Deep Learning Solutions for Medical Image Segmentation. 63 101693. DOI 10.1016/j.media.2020.101693.
- [10] Jorge Corona, Daniela Rodriguez-Delgado, John Dairo Henao Penenrey, Juan Carlos Sandoval, and Carlos Poblano-Salas. Design and Fabrication of a Customized Partial Hip Prosthesis Employing CT-Scan Data and Lattice Porous Structures. XXXX DOI 10.1021/acsomega.0c06144.
- [11] *MRI Interpretation*. [https://www.radiologymasterclass.co.uk/tutorials/mri/t1\\_and\\_t2\\_images](https://www.radiologymasterclass.co.uk/tutorials/mri/t1_and_t2_images).

- [12] *Anatomical Plane*.  
[https://en.wikipedia.org/w/index.php?title=Anatomical\\_plane&oldid=1191622737](https://en.wikipedia.org/w/index.php?title=Anatomical_plane&oldid=1191622737).
- [13] DICOM Standards Comitee. *DICOM - Introduction and Overview*. National Electrical Manufacturers Association (NEMA).  
<https://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf>.
- [14] R.W. Cox, John Ashburner, Hester Breman, Kate Fissell, C. Haselgrove, C.J. Holmes, J.L. Lancaster, D.E. Rex, S.M. Smith, J.B. Woodward, and Stephen Strother. *A (Sort of) New Image Data Format Standard: NiFTI-1*.
- [15] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S. Kirby, John B. Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing The Cancer Genome Atlas Glioma MRI Collections with Expert Segmentation Labels and Radiomic Features. 4 170117. DOI 10.1038/sdata.2017.117.
- [16] Spyridon Bakas, and et al.. *Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge*.  
<http://arxiv.org/abs/1811.02629>.
- [17] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I. Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. *Ilstik: Interactive Machine Learning for (Bio)Image Analysis*. 16 (12), 1226–1232. DOI 10.1038/s41592-019-0582-9.
- [18] David A. Gutman, Mohammed Khalilia, Sanghoon Lee, Michael Nalisnik, Zach Mullen, Jonathan Beezley, Deepak R. Chittajallu, David Manthey, and Lee A. D. Cooper. The Digital Slide Archive: A Software Platform for Management, Integration, and Analysis of Histology for Cancer Research. 77 (21), e75-e78. DOI 10.1158/0008-5472.CAN-17-0629.
- [19] *DigitalSlideArchive/HistomicsUI*.  
<https://github.com/DigitalSlideArchive/HistomicsUI>.
- [20] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability. 31 (3), 1116–1128. DOI 10.1016/j.neuroimage.2006.01.015.
- [21] *Fast AI Assisted Annotation and Transfer Learning Powered by the Clara Train SDK*.  
<https://developer.nvidia.com/blog/annotation-transfer-learning-clara-train/>.
- [22] *MARIE PACS OR-CZ*.  
<https://www.orcz.cz/en/marie-pacs>.
- [23] OR CZ. SÍLA INFORMACE MEDICAL SOLUTIONS 2021 MARIE PACS BY OR.
- [24] *Philips Introduces IntelliSpace Portal 11 at ECR 2019*.  
<http://www.itnonline.com/content/philips-introduces-intellispace-portal-11-ecr-2019>.

- [25] *Syngo.via RT Image Suite*.  
<https://www.siemens-healthineers.com/cz/radiotherapy/software-solutions/syngovia-rt-image-suite>.
- [26] *Eclipse Treatment Planning System*. Varian Medical Systems, Inc.  
[https://varian.widen.net/s/vfddt6qgn2n/eclipse\\_brochure\\_rad11118\\_may\\_2023](https://varian.widen.net/s/vfddt6qgn2n/eclipse_brochure_rad11118_may_2023).
- [27] *Our Combination with Siemens Healthineers | Varian*.  
<https://www.varian.com/about-varian/our-combination-siemens-healthineers>.
- [28] Hiroshi Shimodaira. *Principles of Human Computer Interaction - HCI Lecture 1*.  
[https://www.inf.ed.ac.uk/teaching/courses/hci/1112/lects/01\\_Principles.pdf](https://www.inf.ed.ac.uk/teaching/courses/hci/1112/lects/01_Principles.pdf).
- [29] Zdeněk Míkovec. *User Interface Design - B4M39NUR*.
- [30] Nicolas Ortiz, Carlos Juan, and Marco Aurisicchio. A Scenario of User Experience. 182–193.
- [31] ISO. *Ergonomics of Human-System Interaction Part 210: Human-centred Design for Interactive Systems*.
- [32] World Leaders in Research-Based User Experience. *Usability 101: Introduction to Usability*.  
<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [33] Jakub Franc. *Psychology in HCI - B4M39PUR*.
- [34] World Leaders in Research-Based User Experience. *Risks of Quantitative Studies*.  
<https://www.nngroup.com/articles/risks-of-quantitative-studies/>.
- [35] Earl R. Babbie. *The Basics of Social Research*. 4th ed ed. Thomson/Wadsworth, ISBN 978-0-495-09468-5.
- [36] Health Liaison Librarians. *LibGuides: Qualitative Study Design: Field Research*.  
<https://deakin.libguides.com/qualitative-study-designs/field-research>.
- [37] Dovetail Editorial Team. *A Step-by-Step Guide to Designing Effective User Scenarios*.  
<https://dovetail.com/ux/user-scenarios/>.
- [38] Kim Salazar. *Scenario Mapping: Design Ideation Using Personas*.  
<https://www.nngroup.com/articles/scenario-mapping-personas/>.
- [39] World Leaders in Research-Based User Experience. *Task Analysis: Support Users in Achieving Their Goals*.  
<https://www.nngroup.com/articles/task-analysis/>.
- [40] *Using User Data – Hierarchical Task Analysis (HTA) – Lilly Browne*.  
<https://blogs.city.ac.uk/browne-inm452-2016/2016/12/25/1-6-using-user-data-hierarchical-task-analysis-hta/>.
- [41] Rosy Alvarez, and et al.. *Functional Design*.  
<https://uit.stanford.edu/pmo/functional-design>.
- [42] *The Difference between Variables and Styles*.  
<https://help.figma.com/hc/en-us/articles/15871097384471-The-difference-between-variables-and-styles>.

- [43] Akanksha Ghai, and Urvashi Tandon. Analyzing Impact of Aesthetic Visual Design on Usability of E-Learning: An Emerging Economy Perspective. 12 (2). DOI 10.18870/hlrc.v12.i2.1325.
- [44] *WinUI 3 Gallery*.  
<https://github.com/microsoft/WinUI-Gallery>.
- [45] Karl-Bridge-Microsoft. *Windows UI Library (WinUI) - Windows Apps*.  
<https://learn.microsoft.com/en-us/windows/apps/winui/>.
- [46] jwmsft. *Color in Windows 11 - Windows Apps*.  
<https://learn.microsoft.com/en-us/windows/apps/design/signature-experiences/color>.
- [47] MingCute Design. *MingCute Icon - Carefully Designed Icon Library*.  
<https://www.mingcute.com/>.
- [48] Assistant Secretary for Public Affairs. *Usability Testing*.  
<https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>.
- [49] World Leaders in Research-Based User Experience. *Usability Testing 101*.  
<https://www.nngroup.com/articles/usability-testing-101/>.
- [50] *Actionable Insights*.  
<https://www.techtarget.com/searchcustomerexperience/definition/actionable-insights>.
- [51] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. Human-in-the-Loop Machine Learning: A State of the Art. 56 DOI 10.1007/s10462-022-10246-w.
- [52] *Desktop Operating System Market Share Worldwide*.  
<https://gs.statcounter.com/os-market-share/desktop/worldwide>.
- [53] (225) *Full Modern React Tutorial - YouTube*.  
<https://www.youtube.com/playlist?list=PL4cUxeGkcC9gZD-Tvwfod2gaISzRiP9d>.
- [54] Jakob Nielsen. *Why You Only Need to Test with 5 Users*.  
<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [55] *Fluent 2 Design System*.  
<https://fluent2.microsoft.design/>.
- [56] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. *Guidelines for Human-AI Interaction*. In:  
<https://www.microsoft.com/en-us/research/publication/guidelines-for-human-ai-interaction/>.

# Appendix A

## User Research

### A.1 Qualitative Interviews

First, I present the interview design document used during the interviews. Finally, both interview results follow.

#### A.1.1 Interview Design

I have prepared a supporting document for the interviewer. Its contents are rewritten from original Czech version to English version in this section.

##### Target Participants

The participants had to be medical doctors with knowledge and previous experience of work with a form of medical imaging data (preferably MRI, CT, but possibly x-ray or other).

##### Interview Notes

I had specified a few additional notes in the document:

- Plan the interview at participants workplace ideally.
- First, conduct the interview, then observe common medical imaging tasks of their job in practice.
- Introduction:
  - Who am I and what am I doing in my work?
  - What do I want from the participant?
  - What can I offer back?
- Help create a comfortable environment during the interview.
- Ending:
  - Thanks
  - Contact information

##### Research Questions and Goals

Next, I specified the following research questions and goals:

- What are their tasks and goals they want to achieve through medical imaging software?
- How well is the software usable?
- How much do they work with mouse and keyboard?
- How is their currently used software different from previous ones (if any)?
- How difficult was it for them to learn how to use this software?
- Where in their workflow is currently their biggest hitch?
- Do they have any interest in segmentation / annotation of data (if they currently use viewer only)?

## Interview Questioning Guide

The interview itself consisted of warm-up questions, main questions, and a conclusion. The main questions were chosen as follows:

- Imagine last week in your work, give an example of a task you have done. / What did you do today?
- Which software do you use for your work?
- For how long?
- Which other software did / do you use?
- When you acquire a new image, what is the first thing you do with it / how do you start?
- What do you do next?
- Is there anything that took you longer to learn?
- What takes you most time?
- At which window / part do you look the most and why?
- Do you control everything by mouse?
- Do you know any shortcuts or tricks you use?
- When you are finished with an image, what do you do?
- Does anybody else look at your results? What specifically?
- Tell me more about an example from last week.
- Could you describe the task, step-by-step?
- Did you stumble upon a problem?
- Name one thing in your work you would appreciate to change.
- Would you appreciate the possibility to make notes / annotations into the images (if not currently present)?

### ■ A.1.2 Radiology Interview Results

This section covers notes and insights from the first qualitative interview that was conducted at the radiology department of a local hospital.

#### Radiology Interview Notes

A voice recording of the interview has been recorded. I have rewritten the notes from the interview to a file which can be found included under the name `Qualitative Interview 1 - raw notes in Czech.md`

#### Radiology Interview insights

By analyzing my notes from the interview, I have summarized key findings into the following insights.

##### **Emotional insights:**

- Participant likes the variety in tasks that their job offers and that he has an option to choose what he wants to do in radiology. It can vary from image evaluation and report writing to emergency treatments under live-screening from radiology devices.

##### **Technology insights:**

- They capture images using various technologies. Apart from x-ray and ultrasound, they can capture multiple 2D image slices or a 3D image from MRI, and a 3D image CT.
- MRI images show soft tissues more clearly and can be more detailed than CT.



- CT images have intensity values standardized in Hounsfield units and have a wider range in general. Various predefined brightness / intensity windows (for evaluation of brain, bones, etc.) can thus be selected.
- The mouse controls of software differ between utilized software.

#### **MARIE WebVision software insights:**

- It runs in a web browser.
- It is used by radiologists for basic viewing of 2D and sometimes 3D data.
- Working with it seems to have a few functional problems which are degrading overall user experience (e.g. user can start exploring slices before a dataset is fully loaded, after it loads fully, the first slice become suddenly selected, ignoring current state).
- Has a pencil tool for contouring.

#### **Phillips CT software insights:**

- Allows more advanced features (like auto-segmentation) than MARIE and is used for special checkups.
- Participant used manual segmentation by pencil / lasso on two slices and then let the software to interpolate.
- Participant is satisfied with this software.
- The default view for contouring includes a big transverse plane section. On the right of it are two other plane sections.

#### **Rapid AI software insights:**

- Offers advanced solutions for e.g. identification of strokes.
- Licence for a year costs a significant amount of money. When there are only a few patients for a year who would need such inspection, the value-cost ratio is too low for the hospital to pay for it.
- Not all radiologists remember and/or use all features, when such occasion occurs, one needs to remind or relearn them. The same for newly introduced features.

#### **Profession insights:**

- Viewing and analysing is the primary task in radiology, segmentation is used only in very specific scenarios.
- In the images, they try to find:
  - inflammations
  - tumors (pulmonary nodules, gut, liver, ...)
    - harder to find in liver (low contrast, vascular system interfering)
  - cysts
  - other sites of irregularities (changes in image)
- They often measure irregularities using a ruler tool. Sometimes it is needed to compare with measurement from previous screening and evaluate whether the irregularity is changing size.
- Before segmentation, it is needed to select a template with classes for segmentation or create own.
- After auto-detection of pulmonary nodules, they check whether the segmentation is correct and whether it is not just a false-positive. Manual editing of such segmentation is only a rare case.
- 3D view is rarely used.

- Other medical professions, such as cardiologists and surgeons often cooperate with radiologists. They also use medical imaging software to view scanned data and read reports written by radiologists.

### ■ A.1.3 Oncology Interview Results

The second interview was conducted at the oncology department of a local hospital. Instead of making a voice recording for later analysis, I have written down notes on paper directly during the interview.

#### Oncology Interview Notes

Most of the notes that focus on the Varian Eclipse software as well as how the participant uses it were rewritten into digital from which I include as an external file under the name `Qualitative Interview 2 - raw notes in Czech.md`.

#### Oncology Interview Insights

I have summarized key findings into the following insights:

##### Technology insights:

- DICOM is a standard format for patient data.
- The process of creating segmentations in this domain is called contouring.

##### Varian Eclipse software insights:

- It offers a selection of workflows: selection, contouring, external beam planning, brachytherapy planning, brachytherapy 2D entry, plan evaluation
- The default view for contouring includes a big transverse plane section. On the right from it are three sections stack above each other. First two are other two planes. The third is a 3D view of image regions.
- During interviewer's contouring try, the behavior of changing slices at all views at once software slowed down our segmentation process a bit.
- It offers limited undo options — the participant frequently saved the work and used saves for reverting changes.

##### Profession insights:

- Radiotherapy treatment planning consists of the tasks of:
  1. opening image viewer
  2. inspecting countered classes
  3. opening a treatment planner
  4. making irradiation calculations
  5. planning the treatment
- There is usually a need for contouring multiple organs / image regions. There is a structure which should be radiated and then structures where radiation should be minimized. Structures are differently prioritized based on allowed irradiation.
- Contouring is done mostly solely on the transverse plane slices.
- The order of actions during segmentation is following: choosing patient → choosing structure classes from a template → selecting appropriate brightness window → segmentation
- Full low-precision contouring of a spinal cord class by the interviewer using a brush and interpolation took about 5 minutes.
- A lot of automated evaluation, measurement, or counting is done directly on the CT machines — the software is at the operating computer.
- Flow of patients and task differences between radiology and oncology departments:

- In radiology department, they rather create written descriptions, measurements, etc. If a tumor is found, they send the patient to oncology department. The oncology department has its own imaging equipment, software and staff, who then create contours for radiotherapy treatment.

## ■ A.2 User Personas

The expected end users of the developed tool are amateurs, as well as medical experts, primarily from radiology or related departments. This why the following groups were described based on conducted interviews.

### Amateur

In the view of this work, an amateur is a person who does not have any or has little previous experience with software for viewing or annotating medical data. This person is only briefly instructed by a professional how to recognize desired structures in the image data and how to use a specialized software. Their motivators for this job will primarily be gaining money, but can also be learning new skills or getting a good feeling from belief of helping a good purpose.

### Radiologist

The radiologist is a medical doctor who has completed the appropriate post-graduate training and interprets medical images. He is used to communicating these findings to other physicians using a report or verbally. The radiologist primarily only views, analyzes and interprets medical data, but is not an expert on data annotation / segmentation. Nevertheless, he knows how to approach such problem. A specialized radiologist might occasionally also use imaging to perform minimally invasive medical procedures.

### Cardiologist (external specialized physician)

Cardiologist is a person who deals with disorders of the heart and the cardiovascular system. Their job includes medical diagnosis and treatment planning of congenital heart defects, coronary artery disease, heart failure, valvular heart disease and electrophysiology. This person might view medical data when needed.

### Oncologist

Oncologists job is similar to the radiologist, except that the main focus is tumors. This person was trained with medical software and both views and annotates medical data for further treatment.

### Physicist (oncology)

A physicist is a scientist who specializes in the field of physics. Our specific case involves a physicist working at an oncology center. Their job includes creating plans for patient irradiation from annotated / segmented medical data. This person views annotated medical data and follows with the creation of radiation plans based on them.

## ■ A.2.1 Defining core needs of personas

I further continued by briefly summarizing core needs of each group.

### Radiologist — primary target

- quick access to patients data
- getting / choosing desired visualisation
  - therefore quick analysis, detection of problems

- quick measurement of anomalies
- occasional annotation and quickly reminding himself the usage
- report creation for other specialist
- notes in images???

#### **Cardiologist (external physician)**

- quick access to patients data
- quickly remind himself the usage of software / (re)learn
- getting / choosing desired visualisation
  - therefore quick analysis, detection of problems
- measurement of anomalies
- report creation

#### **Surgeon**

- quick access to patients data
- quick check of image before / during surgery for reference
- get live feed of image data

#### **Oncologist - primary target**

- quick access to patients data
- getting / choosing desired visualisation
  - therefore quick analysis, detection of problems
- annotation of anomalies and healthy tissues
- report creation?

#### **Physicist (oncology)**

- quick access to patients data
- quick analysis of annotations
- quick (convenient) creation of plan for irradiation

#### **Amateur**

- quick learning of available tools / convenient exploration

## **A.3 3D Slicer Usability Test**

I have conducted a usability test of 3D Slicer on myself. Taken notes from the progress and results of the testing can be found in an external file with name `3D Slicer usability test - raw notes.md`.

### **A.3.1 Collected Insights**

From the notes, I have interpreted the following insights:

#### **Technology insights:**

- Apart from DICOM format, Neuroimaging Informatics Technology Initiative (NIFTI) is a very popular format in imaging software.
- Slicer saves its scenes as custom `.mrml` files.
  - Such scene files are internally represented in XML and contain predefined or custom nodes that make up the scene

- Annotations from the Markups module are saved in additional `.mrk.json` files.
- Segments from the segmentation modules are saved in additional `.seg.nrrd` files by default. NRRD stands for “nearly raw raster data” and serves as a format supporting scientific visualizations and image processing applications

### **3D Slicer software insights:**

- The GUI was cluttered with many buttons and felt overwhelming.
- I had to learn the concept of modules that is used in this software by my own exploration and by watching a tutorial.
- After I learned how to use the modules, I felt less overwhelmed since they allow showing only related buttons in the left pane on the screen. Nevertheless, there were still buttons I did not use or properly understand still visible on the screen.
  - I felt I like the idea that I would first choose a context of a task I want to do, which would narrow down the menus and UI elements on the screen.
- As an inexperienced user not knowing proper ways of doing segmentation in 3D Slicer, I have encountered multiple struggles and even software errors during my testing.

### **MONAI label insights**

- Setting up the framework requires using a command line.
- Additional work is needed for utilizing sample models.
- Annotation within 3D Slicer required an additional amount of cognitive load, since using the model through the GUI was not clear to me at first.
- While annotating, I had to make sure to click apply.
- It was difficult to understand what is the meaning of and the difference between SmartEdit, Deepgrow and Deepedit terms. The work mentions Deepgrow and Deepedit as separate models, but the GUI showed a dropdown with “SmartEdit (Deepgrow)” in a dropdown selector, and then the default “Model: Deepedit” in a nested selector.

## **A.4 User Scenarios**

In this section, full texts of user scenarios that I have created follow.

### **A.4.1 Scenario 1 — Radiologist**

Martin is a radiologist in a main regional hospital and has multiple years of experience working with radiology images. His job is primarily about evaluating medical images and writing medical reports. He focuses mainly on CT scans.

On a regular workday, Martin is assigned a patient to be examined using CT. He reads a report by a fellow radiologist. The patient has pain in his chest and problems with breathing. His x-ray examination suggests undesired pulmonary nodules, so he was recommended for a CT scan of his chest. Martin’s assistant conducts the CT scanning with the patient and saves the data to an internal system. The radiologist is now about to evaluate the results and recommend treatment.

Martin opens up a dedicated CT software that the hospital has a license for. He first searches for the patient’s data in the hospital’s database. He then opens the 3D image and selects a brightness preset appropriate for lung analysis. Next, he selects automatic pulmonary nodule analysis in the software and lets it run. The software recommends three places for the doctor to check. They are automatically segmented volumes which might be pulmonary nodules.

Martin uses his mouse and keyboard to control the image view. He checks all three volumes and discovers, that one of them is not a pulmonary nodule, so he removes this segmented volume. He checks the image on his own to see whether there are other suspicious places, but does not find any. Since the remaining two modules are well enough segmented and editing them is tedious, he continues to read their sizes. One of the nodules seems not dangerous, but the other one has an unusual shape, and its volume is over a certain threshold. Martin thinks this one can be dangerous and should be further checked out.

Martin writes a report about his findings. He recommends that one of the nodules should undergo a biopsy to check whether it is a tumor. He then sends his findings to the patient's treating physician.

#### ■ **A.4.2 Scenario 2 — Radiologist Non-Expert**

Anna, a radiologist intern at the local hospital, needs to measure the area of nerve roots in the spine where a herniation is encroaching and reducing the available space. Accurate quantification of this area is required for patient diagnosis and treatment planning by senior radiologist. Anna opens the Marie WebVision viewing tool and selects the desired MRI scan. As the image slices are slowly loading one by one, she scrolls between the slices to the desired region and wants to start measuring. She then encounters a problem: the view suddenly switches to the first slice. She is surprised and feels a slight anger towards the computer. She decides to remain calm and returns to completing her task. After she is finished, she discusses the results with a senior radiologist. She mentions the encountered problem and is told, that she has to wait for all the slices to load first if she wants to avoid the sudden switch. Anna remembers this advice and will expect this inconvenience next time.

#### ■ **A.4.3 Scenario 3 — Amateur Annotator**

Thomas is a medical student who volunteered to help with training an AI brain tumor detection model. He hasn't worked with 3D medical imaging software yet, but has general anatomical knowledge. As a student, Thomas is interested in brain tumors and would like to learn more about them with this task. He has previously seen a few 2D slices from 3D images with brain tumors.

Thomas is assigned a PC where an open-source image computing software with a special AI module is installed. A professional demonstrates how to open the software, run the AI module, and complete training on one image to him. Thomas is a little overwhelmed since the software looks more complicated than he expected. The professional leaves Thomas with an MS Word note that summarizes the workflow shown.

After the brief showcase, Thomas moves on to try it on his own. He first wants to start up the AI module. He copies a command from the professional's note, starts a command line, and tries to paste it using Ctrl-V, but it does not work. Luckily, he remembers to paste the command using right click instead. He presses enter, and the module seems to start. He feels a bit like a hacker with all the unknown things happening in the command line.

Hoping that the AI module started successfully, Thomas runs the viewing software. Feeling overwhelmed, he looks at the note, telling him to select a specific module. He finds the Modules dropdown and looks for the AI module. After a while, he finds it and selects it. Next, he presses a refresh button according to the note. A few new options now appear in the module window. Thomas clicks the "Next Sample" button and a first brain image is loaded. Thomas is now relieved, that he was able to set it up.

He then continues to the actual training. He uses mouse and keyboard to control the view and searches for a tumor, but he does not find any. Since he was instructed to skip such cases, he presses “Next Sample” again. In the next sample, he is able to localize something that resembles a tumor. Thomas now knows he needs to annotate it. He looks for the tools the professional used, but cannot see them. He finally discovers them under a collapsed Segment Editor menu.

Thomas sees a “tumor” label. He looks at the note, which suggests creating a “background” label so that he can use the “grow from seed” tool. After creating such a label, he clicks on something that looks like a paint tool. He then proceeds to paint some areas around the tumor and inside the tumor. At first, it is hard for him to be precise, but he discovers, that he can make one of the views bigger — after that, it is easier for him. When he makes a mistake, he uses the Undo shortcut. After a few slices roughly painted, he finds the Grow from seed tool.

Following the note, he presses Initialize. He can then see that a full semitransparent annotation appeared. Thomas already sees some imperfections, but is now not sure, whether he should click Apply and then edit the results, or continue editing before applying. He decides to continue editing and returns to the paint tool. The first problem is, that he would like to paint smaller. He looks around for a while. Eventually, he finds a slide bar and the word “Diameter” next to it and uses it to make the brush smaller. As he moves through the slices, he notices that the semitransparent annotation already changed after he painted more strokes. He did not notice at first, since it took some time to recalculate. Now that he knows, he continues to add more strokes until he is satisfied.

Finally, he returns to the Grow from seed tool, and presses “Apply”. The annotation is now more opaque, and the background layer disappeared from view. He remembers that the professional could also see a 3D preview. He returns the view to the 4-views layout, but sees the last view empty. After some time searching, he sees a “Show 3D” button, which seems already pressed. He tries to press it once, now it looks unpressed. Confused, he presses it the second time and a 3D view appears. On it, Thomas finds another imperfection.

Thomas looks for an eraser to delete a part of the tumor label. He finds it easily and manually erases a part on a few slices. He notices that the background layer remained, but is only hidden. Just to make sure, he clicks the eye to show it. He notices, that as he deleted a part of the Tumor annotation, the Background annotation did not change, so there is now a hole on that place with no label. Slightly worried, he decides that this should not be a problem, since the main goal is to have a good tumor annotation.

Lastly, according to the note, he presses Submit label. His first annotation is finished. Thomas decides to take a little break and then continue with the Next Sample.

## A.5 Ideal Application Pass

The following text was planned to evolve into hierarchical task analysis, but ended up as an early ideal application pass for the tool.

**User:** A professional with knowledge to find and analyze specific patterns in 3D medical images.

**Problem:** The user needs to identify problems and desired formations in the images. If a desired formation is identified, its 3D volume is annotated. The annotation and the overall GUI navigation process should be as time-efficient as possible.



**Task:** Annotate tumor using point annotation

**Plan:**

1. **Open an image file**

- a. Drag and Drop file
- b. Browse file
- c. Click Annotate
- d. Choose Point annotation method
  - i. Hover to view tooltip (optional)
  - ii. Choose not to show again (optional)
  - iii. Click point annotation
- e. View Help
  - i. Choose not to show again (optional)
  - ii. Close Help

2. **Find tumor** (free order)

- a. Change view layout (optional)
- b. Adjust image brightness (optional)
- c. Change 3D slice (optional)
- d. Zoom and pan (optional)
- e. Look for formation

3. **Annotate tumor** (if tumor found)

- a. Click to create points
- b. Check if point number satisfied

4. **Save annotation** (if annotation created)

- a. Click Save annotation
- b. Browse folder
- c. Check save annotation into a separate file (optional)
- d. Click save

**Questions:**

- How to save annotations? File? Same folder? Automatic saving?
- At the end, either choose another file, or close?
- How to manage image files in general?
- Not necessary to always annotate, but when yes, the way towards annotation should be easy
- Two options when opening a file - View / Annotate buttons
- Should multiple files in one session also be considered?
- Is there a need for continuous workflow?

# Appendix B

## Prototype Development

### B.1 Early Design Specification 1

This appendix represents my first draft design notes. These notes roughly correspond to functional specifications for the tool formed based on the requirements from interviews, followed by hypotheses and questions for further research. The specification was created with expected use in radiology department in mind, before the ultimate goal for the tool was clarified. Notes ending with a question mark were inspirational and subject to further inspection, as since the mentioned features might not be needed in the final product.

Since the specification is of lower formal quality, it was moved to an external file `Functional specification (early 1).md`.

### B.2 Early Design Specification 2

This appendix follows up on the previous specification by adding additional notes and use cases that I constructed after further consideration and gathering more information. The notes can be found in `Functional specification (early 2).md`, while the use cases are listed here.

#### B.2.1 Use cases

After the usage clarification, I have defined a new set of use cases:

- Definition of classes for annotation for a folder (dataset) or a file
- Annotation of a whole folder with defined classes
- Annotation of individual file with defined classes
- Annotation of individual file from scratch
- Annotation of individual file from scratch (fully manual)
- Viewing of annotated file
- Viewing of unannotated file

### B.3 FigJam drafts

This appendix contains links and previews to shared versions of created FigJams.

#### B.3.1 Prototype LoFi 1.0

A preview of the draft can be seen in B.1 and the original can be found on the following url:

- <https://www.figma.com/board/YyxUrPHFzcDstJNBCaycrX>



Figure B.1. Early prototype FigJam draft

### B.3.2 Prototype LoFi 4.0

A preview of the draft can be seen in B.2 and the original can be found on the following url:

- <https://www.figma.com/board/sNOFm0InuzidNKrSNzcUQU>

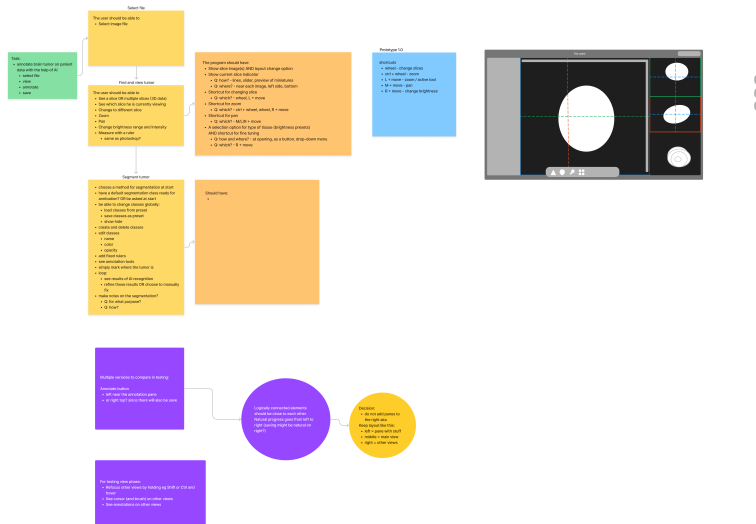


Figure B.2. Newly adjusted version of the previous FigJam draft

### B.3.3 Prototype 1.1

A preview of the draft can be seen in B.3 and the original can be found on the following url:

- <https://www.figma.com/board/y7YrTSyfUQ4v8W0jCqteUi>

## B.4 Paper Sketches

This appendix contains additional two paper sketch images not covered in the work. The images are on figures B.1 and B.2

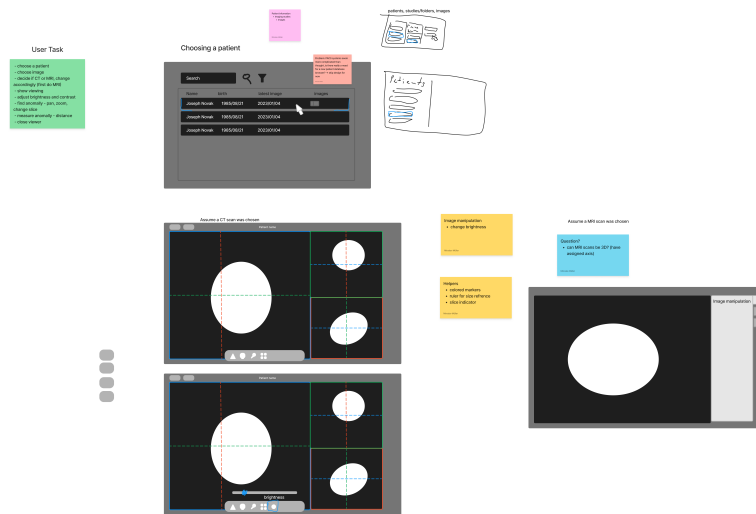


Figure B.3. FigJam drafts after returning to Figma development

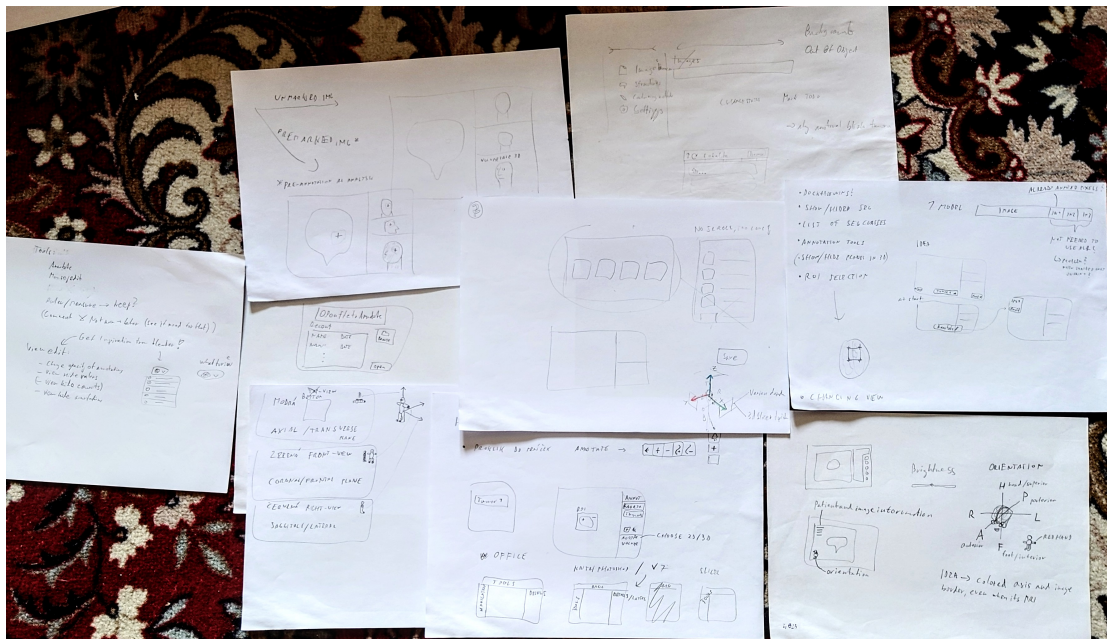


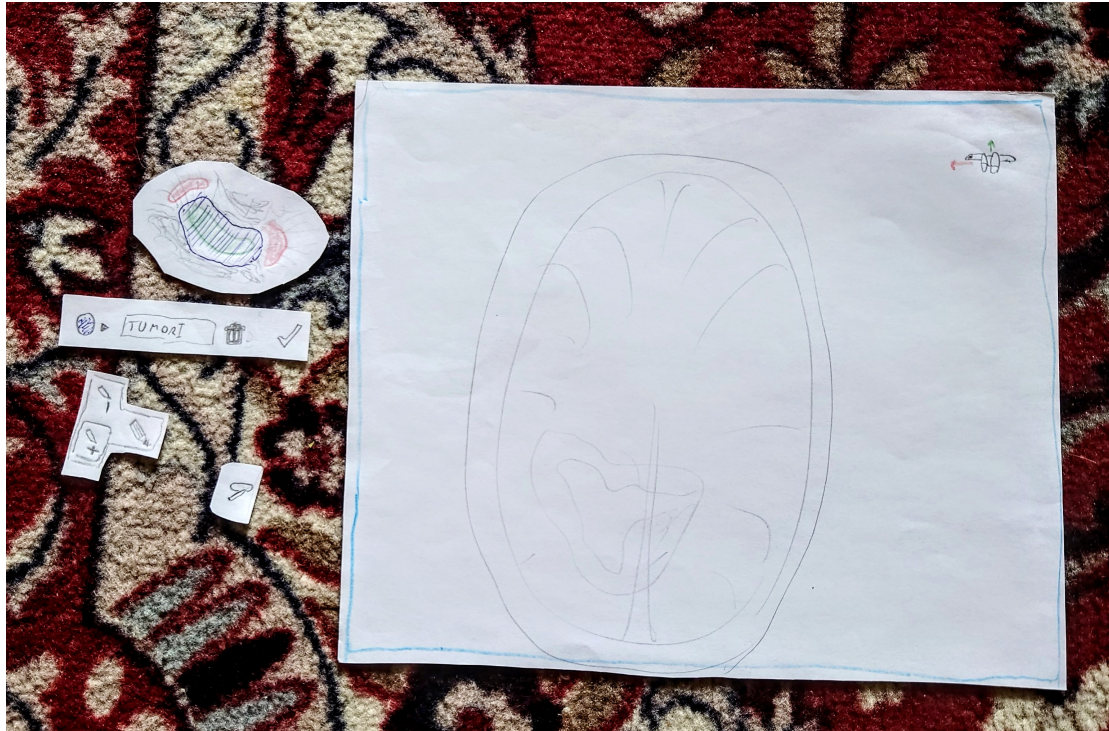
Figure B.1. Overview on various paper sketches for the prototype

## B.5 Annotation Interaction Methods

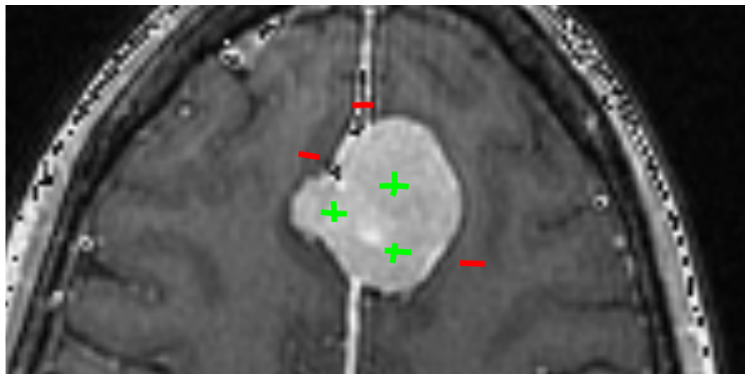
This appendix includes a design for proposed annotation methods for effectivity comparison.

This version corresponds to proposing interactions without an interactive feedback loop. The user completes an annotation, which is then used by AI. The user is a researcher with an IT or AI background. The output data of an annotation varies by type of annotation. The final goal by the AI is to create a binary volume mask aligned for original data voxels. Annotation of one object is considered.

### B.5.1 Common Interaction Variations



**Figure B.2.** Additional paper sketch including GUI for annotation class editing and annotation tool buttons



**Figure B.1.** Positive and negative point annotations

### Positive and Negative Annotations

For each interaction, there is the possibility to work with positive clicks only, or both positive and negative. Positive and negative clicks on an image are demonstrated on figure B.1 .

An alternative to having negative clicks might be always annotating two or more regions at once (eg. tumor, healthy tissue, ...), which would be touching each other and fill the whole space.

### ROI

There is the possibility to also always specify a 3D region of interest and include it in the resulting data.

## ■ B.5.2 Interactions

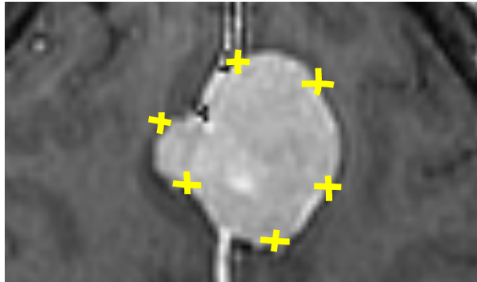


### Point Clicks

The user clicks to select several points while browsing individual 2D slices to mark the desired volume. The output is a list of 3D locations. Demonstration can be seen in figure B.1 .

### Clicking on the edge

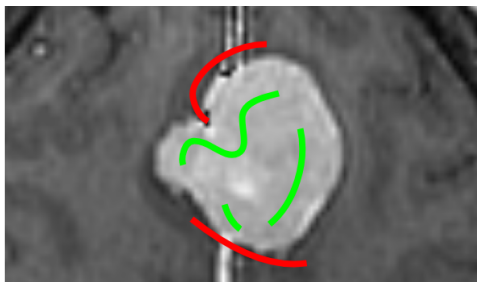
An alternative to making positive and negative clicks is clicking on the border of the desired area B.2 .



**Figure B.2.** Edge point annotations

### Inside / Outside Scribbles

The user draws curves on several 2D slices to mark the desired volume. The output is a list of appropriate representations of 2D curves in a 3D space B.3 .



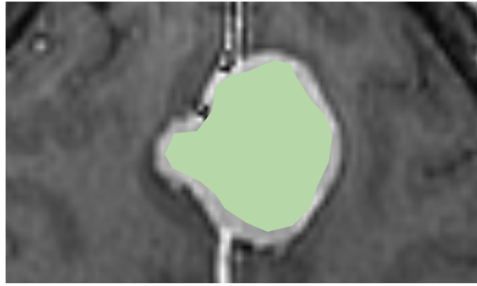
**Figure B.3.** Inside and outside scribbles annotation

### Lasso / Specify Edge

The user draws a curve along the edge of the desired area on selected 2D slices. The output in this case can be the whole area on selected slices (in case the lasso always closes) or eg. just the line(s) specifying the edges of the desired areas B.4 .



**Figure B.4.** Lasso annotation



**Figure B.5.** Brush annotation

### Brush Painting

The user paints on several 2D slices to mark the desired volume. The output is a volume mask (eg. painted voxels) B.5 .

### Brush size

The user can adjust the brush size to draw with.

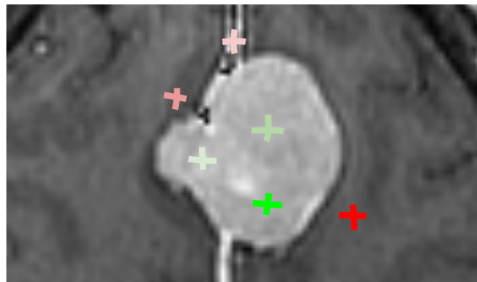
### 2D or 3D

There is the possibility to allow 3D brushes with a 3D head shape that paints multiple 2D slices at once. This could either be a cube, cylinder or a sphere.

## ■ B.5.3 Possible UI/UX features

### Between-layer Visibility

There is the possibility to preview annotations on different layers with reduced opacity. The opacity can be scaled with distance B.6 .



**Figure B.6.** Demonstration of between-layer visibility of point annotation

### Annotation Editing

There is the possibility that the user could drag or delete previously placed points or scribbles, or undo and redo the changes in general. An eraser tool would fit the brush interaction.

## ■ B.6 Axure RP prototype

The Axure RP prototype that I create is located in the external files in a folder Axure RP. I include two versions, while version 7 is not necessarily better than 2. Sadly, I have lost the most functional version, which was somewhere between version 2 and 3. I assume version 2 represents the interactive functionality currently the best.



## ■ B.7 Design Specification 3

This section covers a functional specification made after returning to Figma development. Two most important pages are shown here - namely functional specification and annotation flow diagram. The file `Functional specification.drawio` (or `Functional specification.pdf`) then contains the remaining notes and draft.

### ■ B.7.1 Functional Specification

The functional specification is shown on B.1 .

## Main use cases

### 1) View single file

Open a single file from a file browser and just view the data.

### 2) Annotate single file

Open a single file from a file browser, choose annotation method, choose annotation class, annotate, save.

### 3) Create a dataset for annotation

Create a list of file names to be annotated. This list has a given order - it can be set parametrically by name, date, etc. Or reordered manually using GUI. The representation should ideally be a simple text file, so that it can be easily generated externally.

### 4) Annotate dataset

Open files from a list. See which are annotated and which not. From there, the user can start annotating from selected file, from start/last a.

## App starting points

1. Launch app, then choose what next
  1. single file open
  2. single file drop
  3. folder open
2. Doubleclick file
3. Right click inside folder

## Sampling from dataset

## Work assignment

Idea? The assignator sends a link to download file to the annotator. He downloads it, and should be

Research question: Is there a need for multiple annotation files per image?

Research hypothesis: Do not use the word annotation, but rather use segmentation. Answer: use both, depending on context

## Functional specification

+ brainstorming

## Goal

Tool for development of AI applications in radiology.

## Main use

Training AI models for segmentation in radiology.

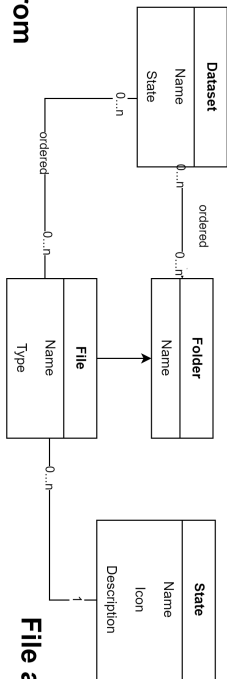
## Annotation files

An additional annotation file is created once an annotation is created or a file's state is changed. It has the same name as the original file but different

## State

Represents a state of a file's annotation. There are default states as well as optional creatable ones. In the table below, all default states as well as a few demonstrative optional are listed.

State	Default	Description
Unannotated	Y	A default file state for files that have not yet been annotated. Such files do not have any additional files yet.
In progress	Y	A file whose annotation was started but not marked as completed yet.
Annotated	Y	A file whose annotation was marked as complete.
Marked for review		
Reviewed		
Rejected		
Marked for annotation		



## File actions

- view
- annotate
- check

## Folder actions

- Make dataset from folder
- creates a dataset file in that folder
- check

## Other

- sort files
- change tabs
- Files
  - Recent
  - Dataset
- create dataset
- refresh files
- browse
  - folder
  - dataset
- file -> opens folder with file selected

## File info

- on file
  - name
  - type
  - status
  - date
  - details
  - full path
  - size
  - number of slices
  - other info

## Folder info

- 

## Other

- show number of selected files/folders

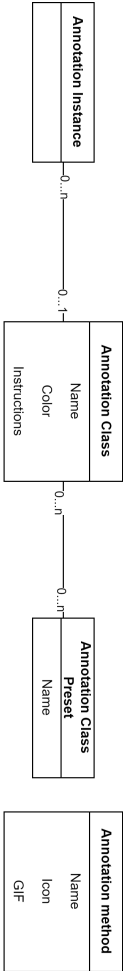
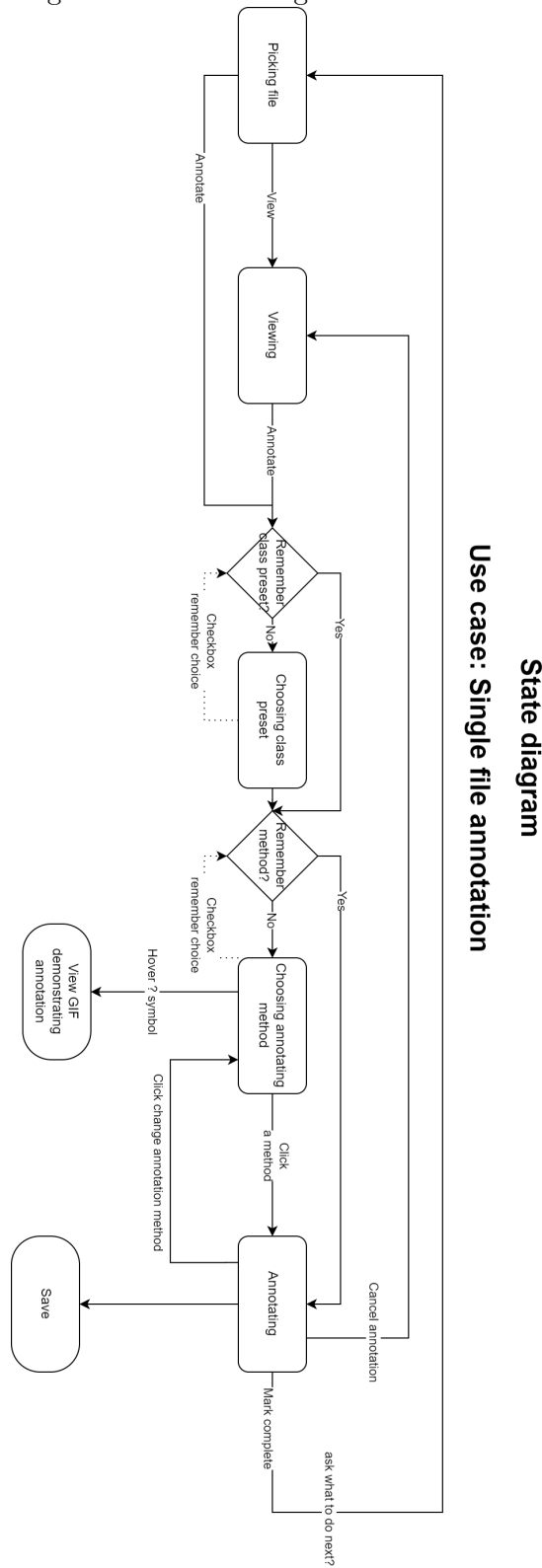


Figure B.1. Functional specification diagram

### B.7.2 Annotation Flow Diagram

The annotation flow diagram is shown on figure B.2 .

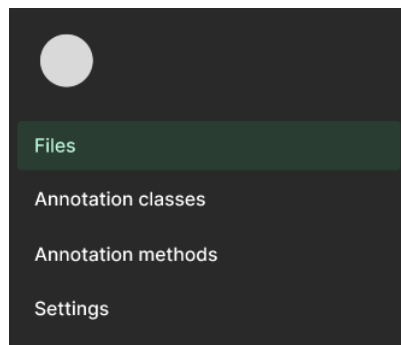


**Figure B.2.** Single file annotation state flow diagram

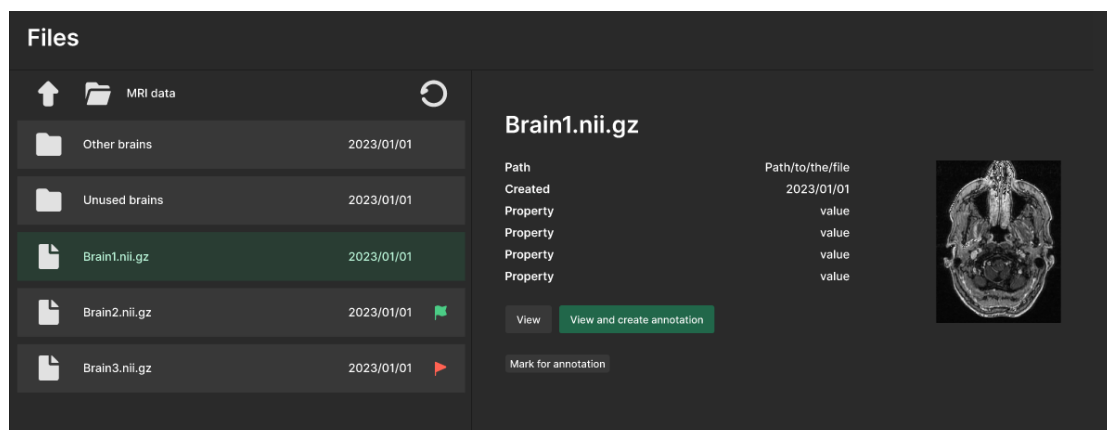
# Appendix C

## Prototype Before Testing

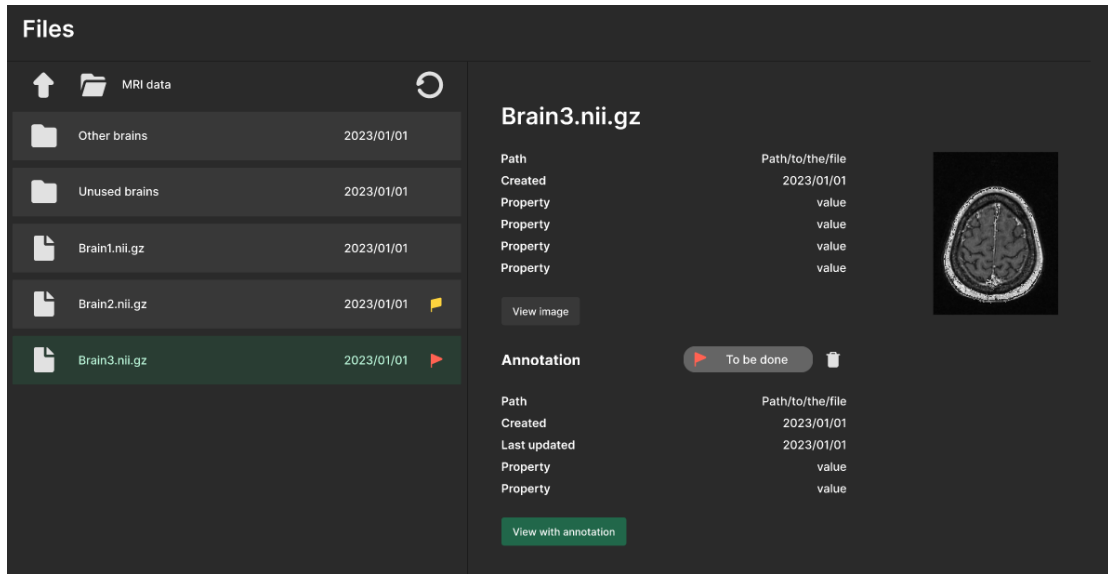
Following is a set of images from the prototype before it was tested through usability tests on figures C.3 to C.12 .



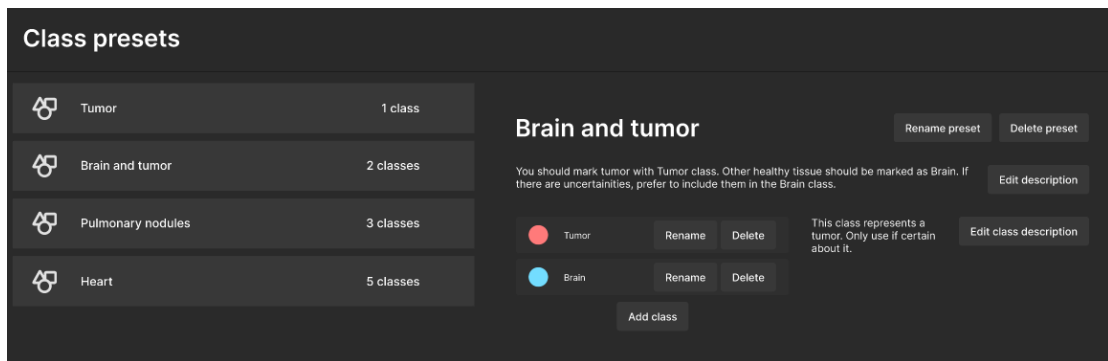
**Figure C.3.** Menu



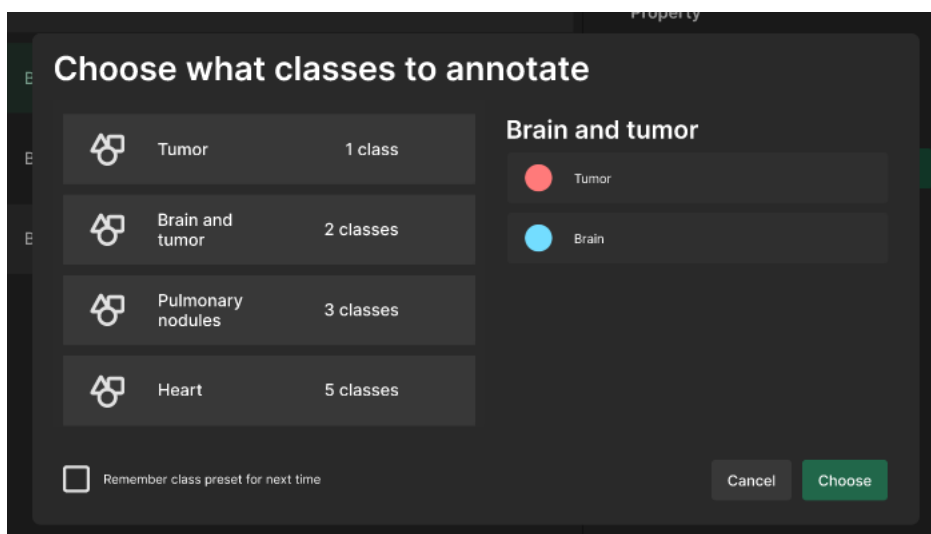
**Figure C.4.** File tab with detail on file without annotation



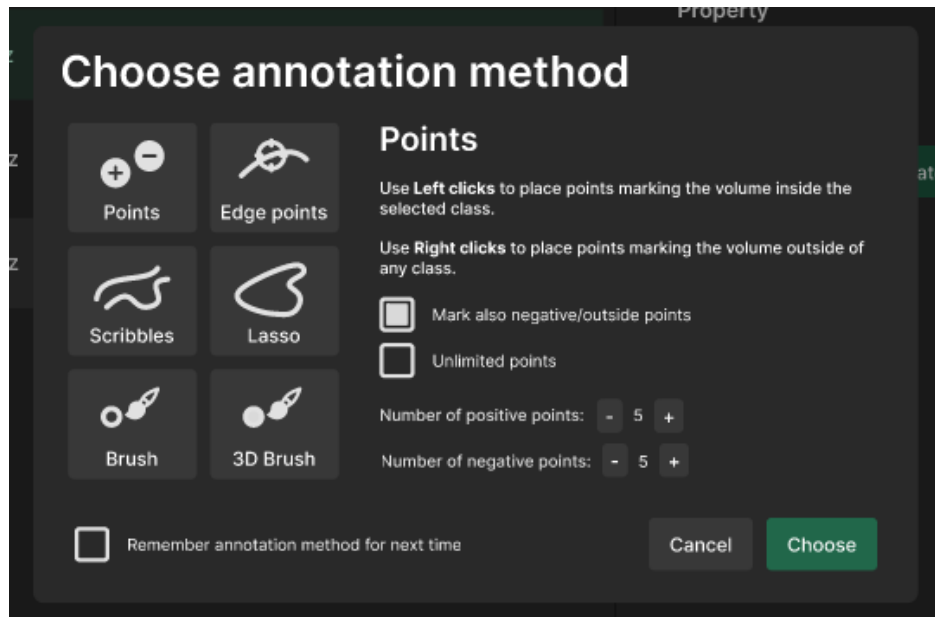
**Figure C.5.** Files tab with detail on file with annotation



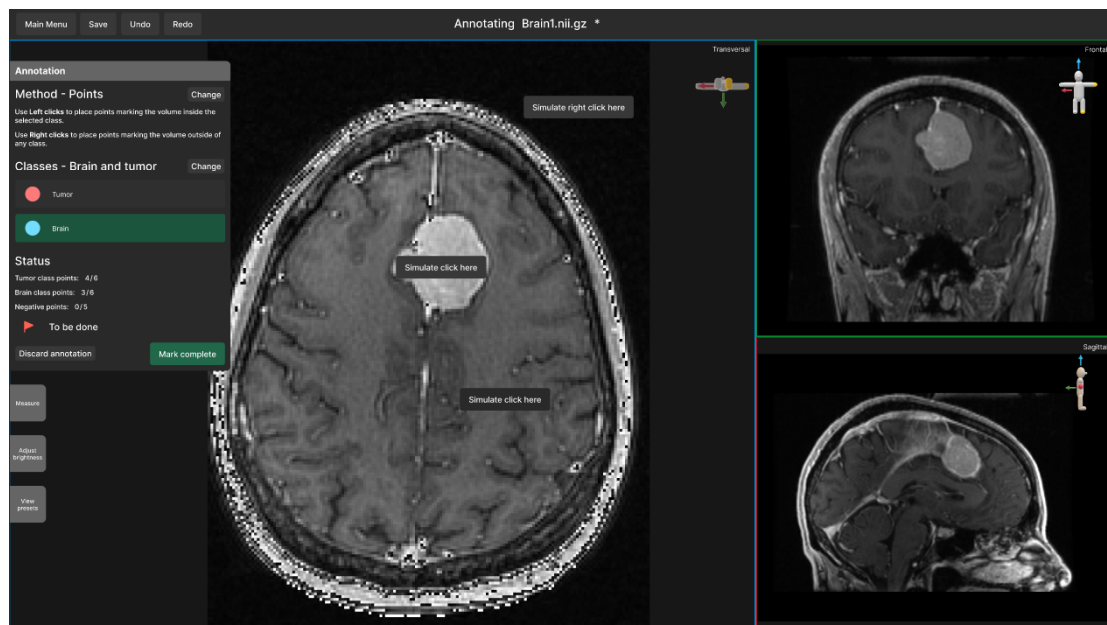
**Figure C.6.** Class presets tab



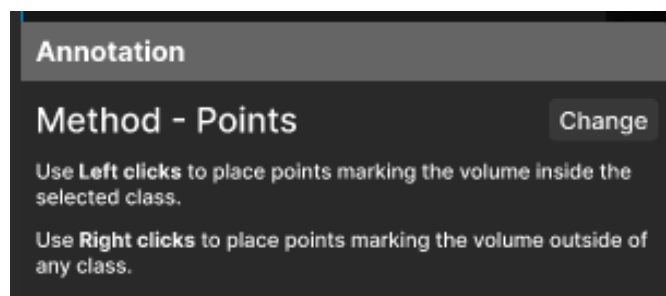
**Figure C.7.** Choose what classes to annotate popup with Brain and tumor preset selected



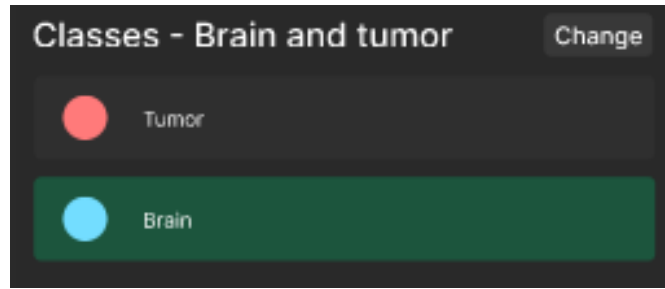
**Figure C.8.** Choose annotation method popup with Points method selected and negative points turned on



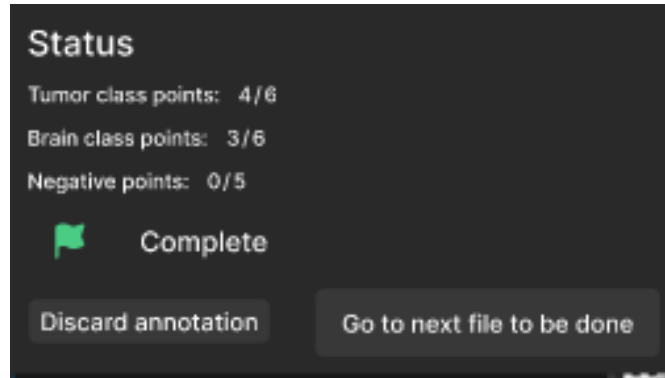
**Figure C.9.** Viewer screen with annotation tool pane open; A single brain slice is shown; Buttons that simulate point annotation are located on the annotated image



**Figure C.10.** The top of annotation tool pane with selected Point annotation method



**Figure C.11.** The middle of annotation tool pane with Class selection section; Brain class is currently being annotated



**Figure C.12.** The bottom of annotation tool pane with current Status and number of annotated points



# Appendix D

## Usability Test v1.0

This section contains the plan and scenario for the first conducted usability test. The test design has been refined through discussion with my advisor before put into practice.

### D.0.1 Test Properties

- Moderated
- In-person
- Assessment

### D.0.2 Goals

- Check whether a basic PC user will be able to follow instructions and complete basic tasks regarding annotation, aka whether the user will be able to
- easily learn how to use the tool
- work fast
- intuitively understand the GUI
- Determine and pinpoint weak and problematic spots in the process for improvement of the above

### D.0.3 Testing Environment

- Provided laptop (incl. screen and keyboard) and mouse
- Figma prototype
- Quiet place with chair and table
- OBS screen recording software
- Speaking aloud - nahrvat

### D.0.4 Metrics

- The success rate of individual tasks
  - Number of errors
  - The time each task requires
- Users' subjective satisfaction
  - Possibly measure mood before and after the test

### D.0.5 Notes

This version of the prototype is not supposed to test the final workflow of the program. Rather, it is meant to test the core aspects of an early version, where an interactive human-AI loop has not yet been introduced.

## ■ D.0.6 Test Scenario

### Introduction Speech

Note: includes notes in Czech

- what is the topic
  - tyka se lecarskych obrazu napriklad z CT a MRI, kde obrazek je 3D / trojrozmerny sken nejake tkane
- what am i doing
  - tvorim prototyp nastroje na praci s lecarskymi daty, ve kterem nam jde jednak o zobrazeni takovych obrazu, tak i o vytvoreni popisku a znaceni jednotlivych typu tkani v danem obrazu
- what are we doing here today
  - otestovat zakladni navigaci v programu, ten proces prace s tim
- what do I want from the participant
  - snazim se zmapovat, jak clovek ktery to nezna s tim bude pracovat
  - je v poradku, i kdyz neco nepujde, nebo bude z neceho zmateny
  - proto bych byl rad, kdyby v prubehu testu premyslel nahlas - je pro me dulezite poznat, jak nad tim premyslis, a proc na co klikas, i kdyby to mel byt jen pokus omyl
- how will we do that
  - budu zadavat ukoly, a pokladat pripadne otazky
  - to co uvidis, neni vysledny program, ale teprve brzky prototyp - ne vse tedy bude fungovat a nepujde na vsechno kliknout co bude ve finalni verzi. i tak budu rad, kdyz budes komentovat, pokud bys treba byval chtel na neco kliknout, nebo ocekaval, ze se neco stane, nebo naopak nestane.

### Tasks

1. View an image called brain1.nii.gz, once you see a medical image, tell once you are done
  1. Q: Can you explain, what do you see?
2. Now mark five points inside the tumor on this image, and finish by marking it as complete
3. Now go to image brain2.nii.gz. Change the method so that you can also mark points that lie outside the tumor for more precise annotation. Add as many points as you feel and click mark complete at the end.
4. Now delete the annotation for this image and mark this image as to be done, prepare it for annotation using 6 edge points.
5. Then go and annotate Brain3.nii.gz using 5 points for brain, 5 points for tumor and 3 points outside, mark complete
6. Tell, how would you add another class called Vascular system to Brain and Tumor? Show by navigating the program.

### Follow-up questions

1. What was most confusing for you during the test?
2. What did you like the most, what felt good to do?

### Optional questions

1. How did you feel about changing between Brain and Tumor class?
2. How would you change the color of the Tumor class?
3. How do you interpret the flags there?
4. What do you think Mark for annotation is used for?
5. How would you browse to a different folder on the computer?
6. Did you look the text near the file name at the top of editor?

### ■ D.0.7 Test notes and findings

Notes from the tests with interpreted findings can be found in the following files:

- Usability test 1.0-F.1-1.md
- Usability test 1.0-F.2-2.md
- Usability test 1.0-F.2-3.md

### ■ D.0.8 Usability Test v1.0 - Findings Summary

This appendix presents a summary of interpreted findings from the first usability test. It is represented by a list of actionable insights, roughly sorted by severity and number of affected users. A table with other noted insights follow.

#### Actionable Insights

Actionable insights are contained in the following table.

Actionable insight	Users	Severity	Implication for Prototype Revision
The simulated interaction for placing points using buttons in the prototype is not satisfying - leads to confusion since its a different type of interaction and also lacks appropriate feedback. Note: clicks are expected to place points with the color of the annotated class.	3 (1,2,3)	xxx	Keep current design trend and implement in next prototype
<b>View image</b> and <b>View with annotation</b> buttons are confusing for users and add cognitive load for deciding. They do a similar thing but are with different style and sometimes on different places.	2 (2,3)	xxx	Include only one button to open the file called <b>Open image</b> . Annotation can be started only from the viewer.
Including <b>the outside</b> or <b>negative points</b> in the annotation is not properly understood - is confused for annotation with another class (eg. outside a tumor is brain)	2 (2,3)	xxx	Think whether to include such option at all, if keeping, rename and redesign clearly.
The left menu feels like a central point from where everything is done. It was noted it tries to do too many things. A user tried to find changing the brain file here. The selector of current class for annotation feels a bit clustered within the annotations pane.	2 (1,3)	xx	Think about logically close information should be near each other (eg in a section), but sections should be clearly distinguished. Compare the use of left pane vs top menu for certain tasks.
Mark for annotation button is weirdly sized and placed. It is expected to work for multiple files at once. Annotation method popup is unexpected on click.	2 (1,3)	xx	Make it a regular button. Do not open the popup but only mark with todo status. Preselecting method is a next optional step. Think about multiple file marking.
Annotation methods are expected to be adjusted per class but there is no option to adjust the number of points for a different class (if more than 1). Clarification on this topic is lacking.	2 (1,2)	xx	Clarify and/or restructure annotation method settings with multiple classes in mind.
There are two buttons in editor that change the status of file. The status of the file is also indicated/manipulated at multiple places and yet not clear to some users (top name, annotation menu status, annotation menu Mark complete, main menu -> save). Mark complete can be also understood per class, not per file and leads to confusion.	2 (1,3)	x	Think about a proper way to clearly indicate the status/progress of a file at one place and allow changing the status and saving from that place. Clearly distinguish between what information belongs to a class and what to a file.

## Actionable Insights pt. 2

Further Actionable insights are contained in the following table.

Intuitively, the users would circle out the tumor (like lasso). The lasso tool was also recognized and would be preferred.	1 (2)	x	Prioritize testing lasso method
Create annotation popup in main menu is surprising	1 (1)	x	Solved in
Presets and classes are not that clearly distinguished	1 (1)	x	
Question mark help is actually used	2 (1,3)		Further evolve the use of such help
It is expected that the brain could be rotated eg. by clicking and dragging the 3D figure	2 (1,2)		
Some text (eg. the point counter on the left) is too small / hard to notice	1 (1)		Make point counter and other small text more clear
After clicking Mark complete, it changes to a non-primary button with a different type of action and therefore also removes a visual clue for the user (what to do next, what is important).	1 (3)		Do not change Mark complete button to a different type of button/action.
Having 3 views lead to slight confusion about which can and should be used for annotation, and whether the points are counted per view.	1 (3)		Make it clear which view(s) can be used for annotation and that the number of points is total
The icon of point annotation with plus and minus signs feels little confusing	1 (3)		
Some text (tumor class points, names of axes) is harder to read	1 (2)		
The annotation method settings might be little overwhelming at first	1 (2)		
When annotating with multiple classes, it is harder to notice which class are we currently annotating	1 (2)		Adding a cursor with color should help, but selected element should be clearer
Users would like to use right click on a file (eg. to mark for annotation)	1 (1)		Add right click on file
Adding new class is expected to be possible from Change classes menu	1 (1)		Add option to change classes from there
Modifying the classes is expected to be possible per file bases (without changing the globals)	1 (1)		
Changing settings of annotation is expected in File -> Project or something similar	1 (1)		

### Actionable Insights pt. 3

Further Actionable insights are contained in the table below.

The <b>Annotating ... and Viewing...</b> text at the top did not help distinguishing, whether annotation is in progress	1 (1)		Show information about annotation status only at one place
The ability to create annotation both from the main menu and from the viewing screen is little confusing.	1 (1)		See related insight
Changing color of class is expected on click on the circle	1 (1)		Keep current design trend and implementation
Checkboxes and the ability to select multiple files is hard to find	1 (1)		Further develop multi-file actions (eg Mark for annotation)

### Other Insights

Other insights are contained in the table below

Actionable insight	Users	Implication
Basic layout seems clear	3 (1,2,3)	Keep
Discarding annotation seems clear	3 (1,2,3)	Keep
Flag meanings were interpreted relatively clearly: no flag - nobody touched it; red - todo (suggested alternatively tumor found); yellow - in progress (not mentioned by participants); green - complete (suggested alternatively no tumor found)	3 (1,2,3)	Keep as progress indicators
Mouse scroll and click and drag are expected to move the image	1 (2)	
Mark complete seems intuitive and easy to do	1 (2)	
Double click on file is expected to open it	1 (2,3)	Add support
Navigation to main menu seems clear	2 (1,2)	Keep the button at that place
User pointed out the possible need for making a text note on the annotation	1 (2)	Consider for future
User pointed out the list of predefined classes would probably be bigger in reality and include medical terms	1 (2)	Keep that in mind when designing
User pointed out how he decided based on whether buttons are primary or not.	1 (3)	Keep that in mind when designing
<b>Go to next file to do</b> feels nice to have, is expected to go to any next file without complete status	1 (3)	Keep, but rename for better clarity. Think about which files to go to.
The counter of points is understood after noticing it	2 (1,3)	
The annotation / viewing screen does not have a proper name yet		Screen naming: Main menu, Viewer ?
The name <b>class preset</b> and <b>class</b> might not be the right vocabulary		

# Appendix E

## Prototype After Testing

This appendix presents the final version of the prototype. Screenshots can be seen on figures E.13 to E.17 , while the original Figma prototype can be viewed on the following link:

- <https://www.figma.com/design/gMJBMoArskC79AcPjnMr6>

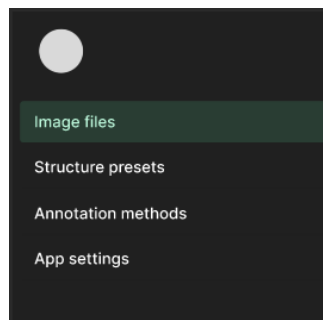


Figure E.13. Updated menu tabs

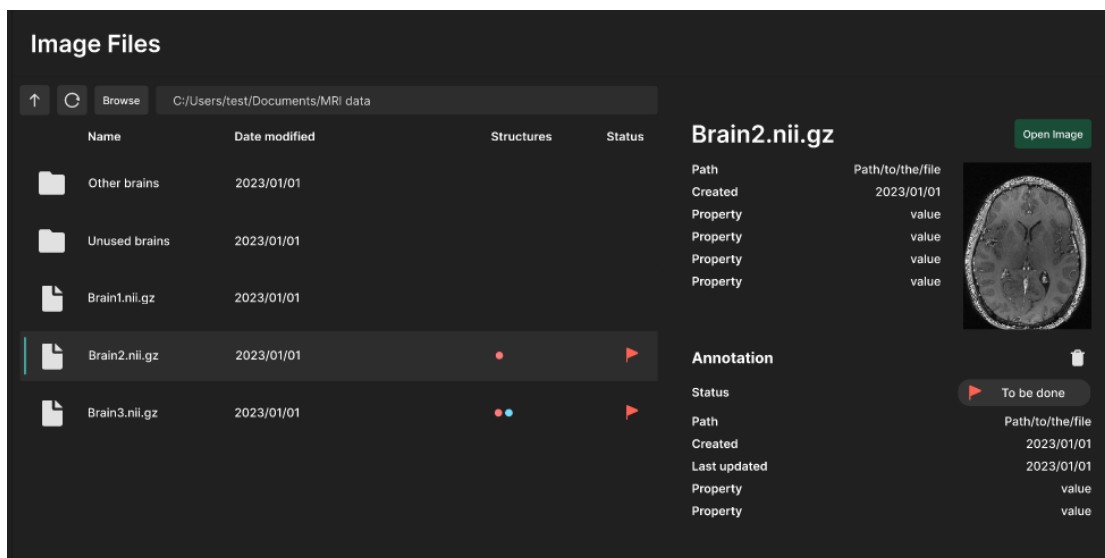
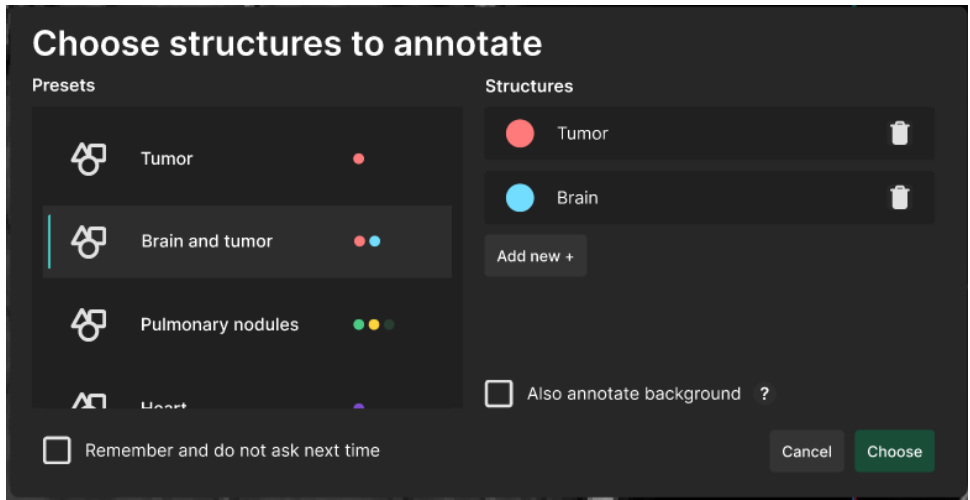
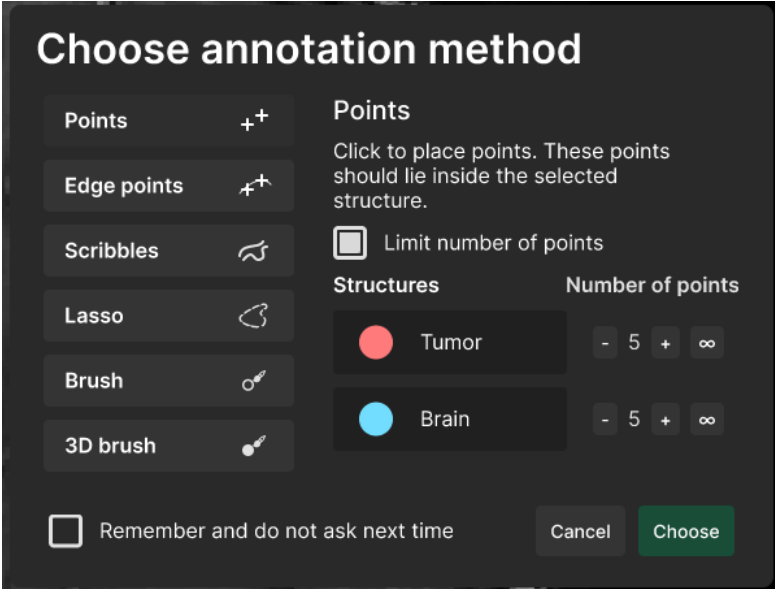


Figure E.14. Updated file selection screen





**Figure E.15.** Updated choose structures to annotate dialog



**Figure E.16.** Updated choose annotation method dialog

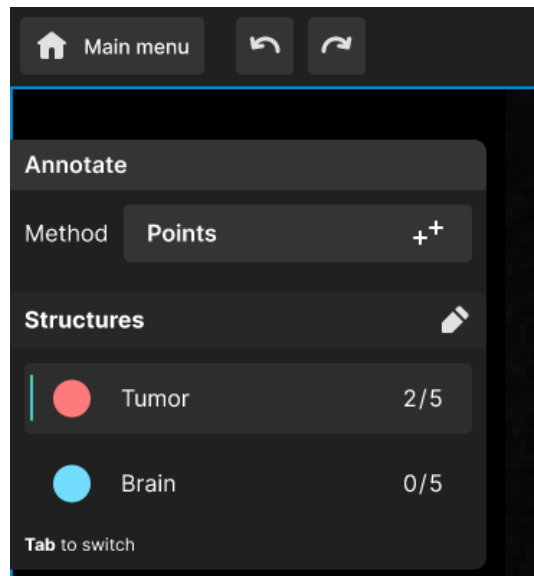


Figure E.17. Updated annotation tool pane

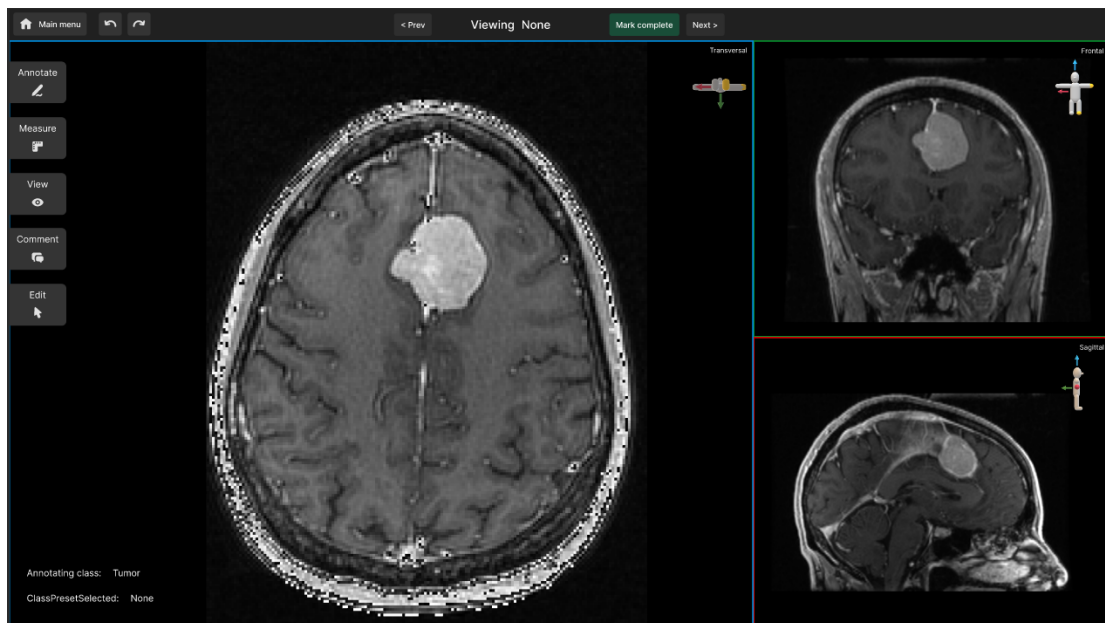


Figure E.18. Updated viewer screen