Critical Design Review Document

Enabling 3D Printing of a Medical CT-Scan:

A Web App for Patients and Practitioners

Prepared by The Slice Is Right

Collaborators: Bryan S, Cooper S, Ethan W, Greg M, Israk A

Client: Terry Yoo

COS 397 - Computer Science Capstone 1

December 8th, 2025

Version 1.0.0

CDRD Preliminaries

Abstract

This document outlines the comprehensive design for a secure, web-based medical imaging application, which is intended to convert raw CT scan data into 3D-printable models that are easy to access and use. The application, called 'AnatoPrint' was developed in response to the ongoing need for both strong patient data privacy and improved clinical accessibility. To address these needs, the system introduces a new architecture that processes all sensitive information locally, rather than relying on cloud storage or external servers. This Critical Design Review presents the finalized system architecture, detailed specifications for each component, and a thorough plan for implementation. The goal is to provide a practical, secure, and user-friendly tool that can be used in clinical, educational, and research settings, while always keeping patient data safe on the user's own computer.

Preface

This document is submitted as part of the requirements for the COS 397 Capstone at the University of Maine. It outlines the design and planned implementation of a browser-based medical imaging tool used to convert CT DICOM data into 3D-printable formats. The work is being carried out in collaboration with Terry Yoo from the Laboratory for Convergent Science, who has expressed the need for locally processed and privacy-preserving imaging tools for research and education.

Table of Contents

List of Tables

Symbols (Nomenclature)

- CT - Computed Tomography’
- DICOM - Digital Imaging and Communications in Medicine
- STL - Stereolithography (3D surface file format)
- G-code - Machine instruction format for 3D printers
- VTK.js - Visualization Toolkit for JavaScript
- ITK.wasm - Insight Toolkit JavaScript library for imaging
- Marching Cubes - Surface extraction formula

Summary

The project is a web-based tool that processes CT DICOM data entirely on the client side. Users upload medical scans, view and interact with volumetric renderings, select tissue thresholds, and export anatomical structures as STL or G-code. The design leverages modern JavaScript and WebGL-based visualization to ensure fast, secure, and scalable performance without requiring external computation. The system architecture uses React and TypeScript for interface management, VTK.js for 3D rendering and reconstruction, and ITK.wasm for DICOM I/O and image processing, with custom modules handling segmentation and export. This Critical Design Review details the technical approach, key client-side design decisions, and major challenges such as performance, privacy, and usability. It also describes the planned test strategy: covering functional behavior, performance, compatibility, security, robustness, and accuracy, and outlines the sequence of deliverables, including the SRS, SDD, UIDD, alpha and beta prototypes, and final system and documentation that will guide development through the remainder of the capstone.

Introduction

This project focuses on creating a fully client-side medical imaging application that converts CT DICOM data into printable 3D formats. The motivation comes from the need to support clinical, educational, and research workflows at the Laboratory for Convergent Science while ensuring user privacy and security.

Purpose

The main goal is to address the client's need for research software that can convert CT data into 3D-printable models of bone, skin, and muscle, while also making sure that all medical data stays on the user's local computer for privacy and security reasons. The client is also looking for a way to make it easier to select tissues, preview models, and export files for use in clinical, educational, and research settings.

Method

The tool will make use of open-source components, such as DICOM readers, Marching Cubes surface extraction, STL writers, and custom algorithms for generating G-code; adapting or extending them as needed to support medical imaging research in the Laboratory for Convergent Science.

Design

This system operates entirely on the client side as a web application, meaning that all DICOM data is processed locally on the user's device. By keeping everything local, the system ensures that sensitive medical information remains private and secure, never leaving the user's computer. The architecture is built in a modular way, with each part of the process, loading data, selecting tissues, generating meshes, and exporting files, handled by its own component. This separation not only organizes the workflow but also helps users move smoothly from working with a CT scan to creating a 3D model ready for printing. The interface itself is designed to be straightforward to use, allowing users to adjust thresholds, see anatomical structures update in real time via WebGL, and export their models in STL or G-code formats. The primary focus of the design is to make the system as usable and accurate as possible, while also giving users full control over their medical data at every step.

Equipment

Component	Technology	Description/Justification
Language	TypeScript	A programming language that is known for its Type safety and complex 3D data structures
Frontend	React + Vite	A JavaScript library and build tool that is primarily for building user interfaces. Fast development and component reusability
3D Engine	VTK.js	A JavaScript visualization toolkit used for 3D graphics, image processing, and scientific visualization.
DICOM Parser	ITK.wasm	A JavaScript library that allows for medical image processing and analysis in web browsers. It reads and parses medical imaging formats.
Testing	Playwright	An automation library that allows for cross-browser testing

Table 2.1: Component Technology Stack

Test

Test Conditions:

Testing will be conducted on a desktop test bench with ≥8 CPU cores @ ≥2.8 GHz, 16 GB RAM, an SSD, and a GPU supporting WebGL2/WebGPU. The tests will run on Windows 11, macOS (current), and Ubuntu LTS. The following variables will be controlled: the same DICOM datasets (small: ≤500MB, large: >500MB-1GB), the same browser version (latest stable Chrome, Firefox, Edge, Safari). Variables that will vary: browser type, operating system, and the specific DICOM dataset characteristics (e.g., slice count, spacing).

Test Procedures:

Functional Testing: Execute use-case-driven test scripts using Playwright to automate user flows (Upload DICOM, Select Tissue, Preview, Export) across all supported browser/OS combinations.

Performance Testing: Use in-app timers and browser performance APIs (PerformanceObserver) to measure processing times for DICOM-to-STL/G-code conversion against targets (NFR-01) and UI responsiveness (FPS, Long Tasks) with large datasets. Compatibility Testing: Use Playwright for visual regression testing (pixel diff threshold ≤1 px) and export file hash comparison to ensure byte-identical outputs across browsers.

Security/Privacy Testing: Run the application offline and monitor network traffic with browser DevTools and a local proxy (e.g., mitmproxy) to verify zero network requests during processing.

Robustness Testing: Feed a suite of ≥20 malformed, corrupted, or unsupported DICOM files into the application and verify it fails gracefully with specific error messages without crashing (NFR-10).

Accuracy Testing: Compare the geometry of generated STL meshes back to the source CT voxel data using mesh validation tools to verify deviation is within ±2% (NFR-07).

Test Results:

TEST ID	REQS	TEST CONDITION	RESULT (PASS/FAIL)	NOTES / METRICS
PERF-01	NFR-01	450MB DICOM -> STL, Chrome, Firefox, Edge, Safari	Pending	Target: ≤ 120s
COMP-01	NFR-04	STL Export, Chrome, Firefox, Edge, Safari	Pending	Do files match across browsers
SEC-01	NFR-05	Full workflow offline	Pending	0 network requests
ACC-01	NFR-07	Bone segmentation STL	Pending	Mesh deviation < 2%

Table 2.2: Test Results

Deliverables, Milestones, and Timeline

Deliverable	**Description**	**Due Date**
SRS	Software Requirements Specification	10/29/2025
SDD	Software Design Document	11/17/2025
UIDD	User Interface Design Document	12/03/2025
CDRD	Critical Design Review Document	12/19/2025
UI Prototype	Interactive mockup with DICOM viewer	12/19/2015
Alpha Version	Core functionality (upload, visualize, segment)	12/19/2025
Beta Version	Full exporting capability	03/01/2025
CIR	Code Inspection Report	3/15/2025
UG	User Guide	4/01/2025
FPR	Final Project Report	4/15/2025
Source Code	Github repository and documentation	5/01/2025

Table 2.3: Deliverable and Milestone Timeline

Conclusions

This Critical Design Review Document synthesizes the requirements, architecture, and interface design for "Enabling 3D Printing of a Medical CT-Scan." The design fulfills the core need for a secure, client-side web application that converts DICOM CT data into 3D-printable STL and G-code files. By leveraging VTK.js and a modern web stack, the system provides interactive visualization and local processing, ensuring data privacy. The detailed plans for equipment, testing, deliverables, and timeline establish a clear and feasible path for successful implementation in the subsequent capstone semester.

Recommendations

Based on the analysis completed during this Critical Design Review, the following recommendations are proposed to ensure the system's successful implementation, long-term maintainability, and usability in clinical and research workflows.

1. Prioritize Performance Optimization Early

Client-side processing of large DICOM volumes (≥500 MB) places heavy demands on memory and CPU. We recommend:

- Profiling segmentation and Marching Cubes performance during initial development.
- Implementing Web Workers early to avoid UI blocking.
- Evaluating WebGPU acceleration as an optional enhancement for capable devices.

2. Implement a Modular Segmentation Pipeline

Segmentation will be a core feature, and likely the most complex. Structuring it modularly will allow:

- Swappable thresholding strategies (basic HU ranges, region growing, machine-learning extensions).
- Easier testing and debugging of each segmentation stage.
- Future collaboration with research groups who may contribute algorithms.

3. Build With a "Clinical Usability First" Approach

Focus on user experience, such as:

- Simple, predictable threshold controls
- Real-time previews
- Clear anatomical presets (bone, soft tissue, skin)
- Accessible UI layout for practitioners unfamiliar with 3D software

This will help the tool be adopted beyond technical audiences.

4. Develop a Robust File Validation and Error Handling Layer

Because malformed DICOMs are common in research settings, the system should:

- Detect unsupported/missing metadata early
- Provide actionable user-facing error messages
- Fail gracefully without freezing or corruption

This directly supports requirements related to robustness and privacy.

5. Establish a Repeatable Testing Framework

To ensure consistent results across browsers and OS versions:

- Automate functional tests with Playwright
- Use hash comparison testing for STL/G-code
- Maintain a standard set of DICOM datasets for regression testing

This will reduce bugs and ensure scientific reliability.

6. Plan for Future Extensibility

The architecture should allow future enhancements, including:

- Multi-structure segmentation (e.g., isolating multiple tissue types at once)
- GPU-accelerated reconstruction
- Support for additional export formats (OBJ, PLY)
- Optional cloud-backed collaboration mode (only if privacy constraints change)

Designing for flexibility now will ensure the project remains valuable long-term.

Back Matter

References

(APA 7th edition formatting)

React Documentation. (n.d.). *React: A JavaScript library for building user interfaces.* https://react.dev/

VTK.js User Guide. (n.d.). *Visualization Toolkit for JavaScript.* https://kitware.github.io/vtk-js/

Insight Toolkit (ITK) WebAssembly Documentation. (n.d.). *itk-wasm library documentation.* https://wasm.itk.org/

Playwright Testing Framework. (n.d.). *Playwright Documentation.* https://playwright.dev/

Lorensen, W. E., & Cline, H. E. (1987). *Marching Cubes: A high resolution 3D surface construction algorithm.* ACM SIGGRAPH Computer Graphics, 21(4), 163–169.

DICOM Standards Committee. (n.d.). *Digital Imaging and Communications in Medicine (DICOM) Standard.* https://www.dicomstandard.org/

Other Sources:

- **The Slice Is Right.** (2025). *Software Requirements Specification (SRS).*
- **The Slice Is Right.** (2025). *Software Design Document (SDD).*
- **The Slice Is Right.** (2025). *User Interface Design Document (UIDD).*

Bibliography

A broader list of materials used throughout the project, including background research, software manuals, and technical papers:

- React Developer Guide
- VTK.js Tutorials & Example Repository
- ITK-Wasm Documentation
- Playwright Testing Recipes
- WebGL2 Specification
- Marching Cubes algorithm
- Marching Cubes Paper
- Smoothing
- Marching Cubes algorithm
- Project Github Repo

Acknowledgement

We would like to express our appreciation to:

- **Terry Yoo**, our client, for his continued guidance, domain expertise, and enthusiasm for advancing imaging tools in the Laboratory for Convergent Science.

- **CollaborAITE**, our sister team, for offering feedback and aiding in cross-team architecture discussions.

- **Conall Gouveia**, for his contributions to the G-code generation logic and general support in computational imaging topics.

- **The COS 397 Instructional Staff**, for their support in the capstone process.

Appendix

Appendix A – Team Review Sign-off

This document confirms that all undersigned members of the The Slice Is Right project team have thoroughly reviewed the entirety of this Critical Design Review for the Enabling 3D Printing of a Medical CT-Scan system. By signing below, each team member acknowledges their understanding of the requirements and formally agrees that the content, scope, and structure of this document are accurate and complete, providing a suitable foundation for the subsequent phases of the project.

The comment section provided for each team member is to be used for noting any minor suggestions, editorial feedback, or non-substantive points of clarification. It is recognized that for this sign-off to be granted, there are no major points of contention regarding the technical or functional requirements outlined within this SRS.

Name	Signature	Date	Comments
Cooper Stepankiw		12/8/25	
Bryan Sturdivant		12/8/25	
Israk Arafat	Israk Arafat	12/8/25	
Greg Michaud		12/8/25	

Ethan Wyman	EW	12/8/25	
Terry Yoo			

Appendix B – Document Contributions

This appendix details the contributions of each team member to the creation of this Critical Design Review. All members participated in the collaborative writing, review, and diagramming process to ensure a comprehensive and unified document. The percentage contributions are estimates that reflect the primary authorship and development effort for the various sections.

Member Name	Primary Responsibilities	Estimated Percentage
Israk Arafat	Appendix, Bibliography, Summary, Table of contents	20%
Gregory Michaud	Preliminaries, Design, Formatting	20%
Cooper Stepankiw		20%
Bryan Sturdivant	Symbols, Intro, Purpose, Method, Equipment, Deliverables	20%
Ethan Wyman	Recommendations Back-Matter Appendix	20%
Total		100%

Appendix C – Software Requirements Specification (SRS)

Software Requirement Specification

Enabling 3D Printing of a Medical CT-Scan:

A Web App for Patients and Practitioners

Prepared by The Slice Is Right

COS 397 - Computer Science Capstone 1

October 29th, 2025

Version 1.0.0

System Requirements Specification

Table of Contents

1. Introduction

This capstone project, completed as part of the requirements for the Bachelor of Science in Computer Science at the University of Maine, focuses on developing a web application called *“Enabling 3D Printing of a Medical CT-Scan: A Web App for Patients and Practitioners.”* The goal of this project is to create a browser-based tool that gives users the ability to upload medical CT scans in the DICOM format, produce a viewable 3D model, and turn them into files that can be used for 3D printing. The application is designed to produce both polygonal (.stl) files, which can be used to visualize different anatomical structures like bone, skin, or muscle, and G-code files, which are needed for 3D printers to accurately reproduce tissue characteristics. One of the most important aspects of this project is that all processing is done locally on the user's computer, which helps protect sensitive medical data and makes the tool more accessible for both medical research and patient education. By working on this project, we are contributing to the University of Maine’s ongoing efforts to advance medical imaging and 3D printing, especially through the Laboratory for Convergent Science.

1.1 Purpose of This Document

This document is meant to lay out the scope, design goals, and reasoning behind the proposed capstone project. It acts as a starting point for planning and requirements, giving the development team, faculty supervisors, and project sponsor a clear idea of what is expected. Here, the system’s purpose, background, and context are explained so that everyone involved understands what the final product should accomplish and how it will be judged. The intended audience includes Dr. Terry Yoo, the University of Maine Computer Science faculty, and any future students or researchers who might build on this work. The document goes on to provide background information and references, sets out the main objectives for the system, and explains where the system ends and outside elements like users, input files, and output devices begin.

1.2. References

Primary Project Source

Yoo, Terry. *Enabling 3D Printing of a Medical CT-Scan: A Web App for Patients and Practitioners.* University of Maine, Laboratory for Convergent Science, 2025.

Web and Technical References

- "DICOM Standard." National Electrical Manufacturers Association (NEMA). Available at: https://www.dicomstandard.org/
- "Marching Cubes Algorithm for 3D Surface Construction." Wikipedia. Available at: https://en.wikipedia.org/wiki/Marching_cubes
- "STL (File Format) Specification." 3D Systems. Available at: https://www.3dsystems.com/support
- "G-code Reference." RepRap Wiki. Available at: https://reprap.org/wiki/G-code websites).

1.3. Purpose of the Product

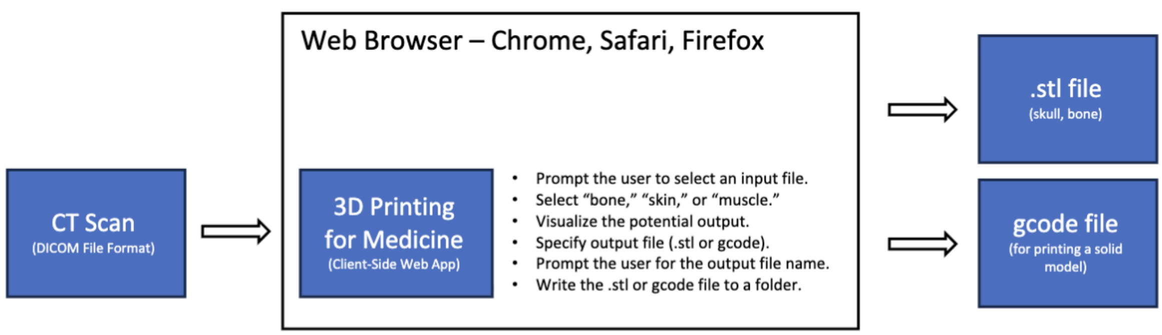
Dr. Terry Yoo, who serves as the Director of the Laboratory for Convergent Science at the University of Maine, has dedicated his research to the fields of medical imaging and 3D printing. His primary focus is on taking complex medical data and turning it into physical models that can be used for both educational and research purposes. The main motivation for this project comes from a clear need within the medical community: there is currently no secure, browser-based tool that enables patients and practitioners to easily visualize and 3D print anatomical structures directly from CT scans. Most existing solutions require third-party software or cloud-based processing, which can introduce privacy concerns and complicate the workflow. To address this, the goal is to develop a functional, open-source web application that can convert DICOM files into both STL and G-code formats. This would make it much easier to prepare CT scan data for 3D printing, while also making these tools more accessible to researchers and educators working in medical imaging. Unlike cloud-based services, which require uploading data to remote servers owned by a third party, this approach eliminates the risks associated with data transmission, storage on external servers, and potential third-party data mining or breaches. Users retain full control and custody of their data from start to finish, which is a security benefit and also makes sure the data never enters a shared or potentially non-compliant environment.

1.4. Product Scope

This project focuses on creating a web application that gives users the ability to turn CT scan data into files ready for 3D printing, all from within their own browser. The main goal is to make the process straightforward and accessible, so that users can handle everything themselves without needing extra software or sending their data anywhere else. The system will allow users to:

- Upload a medical CT scan in the DICOM format.
- Choose an anatomical layer (bone, skin, or muscle).
- Visualize the generated 3D model.
- Select the desired output format (.stl or G-code).
- Download the resulting file locally for use in 3D printing.

Every part of the process happens right in the user's web browser, whether that is Chrome, Safari, or Firefox. This means that all medical data stays on the user's own device at all times. There is no use of cloud storage, outside databases, or any network-based processing. By keeping everything on the client side, the system protects user privacy and keeps things as simple as possible.



2. Functional Requirements

Requirements are ranked 1-5. 1 Being the least important and 5 being the most important.

1. DICOM File Import
    a. The System shall allow the user to select and upload a DICOM file (or folder of DICOM files) from their local machine
    b. The system shall validate the DICOM format before processing
    c. The system shall read and parse the input DICOM file to extract 3D voxel intensity data and relevant metadata such as slice thickness, pixel spacing, image orientation, and study dimensions, which will be used for 3D model generation.
2. Anatomical Tissue Selection
    a. The system shall allow the user to select an anatomical target for visualization and output - bone, skin, muscle
    b. The system shall filter or threshold CT data based on the selected tissue type
    c. The system shall update any visualization or preview accordingly when a different region is selected - going from bone to muscle updates automatically
3. 3D Visualization
    a. The system shall generate and display a 3d preview of the reconstructed anatomy from the CT data within the web browser
    b. The visualization shall allow the user to rotate, zoom, and inspect the 3D model interactively maintaining 30fps while inspecting the visualization
    c. The visualization shall update dynamically based on user selection (tissue type or output format)
4. File Output
    a. The system shall prompt the user to enter or confirm an output file name
    b. The system shall ask the user for an .stl or .gcode file
    c. The system shall save the resulting .stl or .gcode to a local folder chosen by the user
    d. The system shall notify the user when the file generation is complete or if there is an error
5. Polygonal Model Generation(STL conversion)
    a. The system shall apply a Marching Cubes algorithm to extract a polygonal surface representation of the selected anatomy
    b. The system shall generate a 3D mesh from the processed CT scan data
    c. The system shall export the resulting 3D model as a .STL file suitable for 3D printing or slicing
6. STL Generation
    a. The system shall allow the user to select an .stl output type
7. G-Code Generation
    a. The system shall allow the user to select .gcode output type
    b. The system shall generate G-code output that reproduces the X-ray attenuation properties of the tissue - mimicking density or opacity
    c. The system shall write the generated G-code to a user-specified output location on the local machine

8. User Interface
    a. The system shall provide an intuitive and responsive graphical user interface that allows users to:
        i. Select input files(DICOM)
        ii. Choose anatomy type(bone, skin, muscle)
        iii. Choose output type(.stl or .gcode)
        iv. Visualize results in 3D view
        v. Choose output file name and location
9. Client-Side Execution
    a. The system shall run entirely within the client's browser with no external data transfer
10. Cross-Browser and Cross-Platform Compatibility
    a. The system shall pass all system tests on Chrome, Safari, and Firefox
    b. The system shall pass all system tests on across MacOS, Windows, and Linux operating systems
11. Automated Testing and Verification
    a. The system shall include an automated test suite to verify correct operation across supported browsers and OS platforms
    b. The system shall use a web testing framework (e.g., Selenium) to validate core functionality

Use Case Specifications:

Use Case -1: Import DICOM

Number	1
Name	Import DICOM File
Summary	The user selects and uploads a DICOM file or folder of DICOM files containing CT scan data for processing
Priority	5
Preconditions	1. The web app is loaded in the user's browser 2. The user has valid DICOM data stored locally

Postconditions	1. The DICOM file is validated and parsed 2. CT image data and metadata are ready for processing	
Primary Actor	User	
Secondary Actors	Web browser file Input/Output API	
Trigger	User selects “Upload DICOM” or equivalent option in the interface	
Main Scenario	**Step**	**Action**
	1	The user opens the web app
	2	The system prompts the user to select a DICOM file or folder
	3	The user selects one or more DICOM(.dcm) files from their local machine
	4	The system validates the selected files to make sure they’re valid DICOM and in the correct format
	5	The system parses the DICOM data and extracts image slices and metadata
	6	The system notifies the user that the import was successful or reports an error if validation fails
Extensions	**Step**	**Branching Action**
	4a	If the selected file is not a valid DICOM format **:** The system displays an error and prompts the user to try again

	5a	If the file is corrupted: The system alerts the user and terminates the import process
Open Issues	None	
Verification Tests	1. Load valid DICOM: Try with different sizes and number of files. 2. Invalid file: error shown	

Use Case -2: Select Anatomical Region

Number	2
Name	Select Anatomical Region
Summary	The user chooses which anatomical region(bone, skin, or muscle) to visualize and output
Priority	4
Preconditions	● User case 1 complete ● DICOM data has been converted into a 3D volume representation ● The visualization interface is active and displaying the base 3D model
Postconditions	● The selected anatomical region's segmentation parameters are applied to the 3D volume ● The visualization updates to show only the chosen tissue type ● The system stores the active preset for potential STL/G-code export

Primary Actor	End user	
Secondary Actors	• DICOM Processing Module: Applies segmentation thresholds and updates voxel classifications • Browser Interface: Displays the updated 3D visualization and captures user input	
Trigger	User selects a tissue type (bone, skin, or muscle) from the segmentation menu in the interface	
Main Scenario	**Step**	**Action**
	1	The user selects the tissue type (“bone”, “skin”, or “muscle”) from the visualization interface
	2	The system retrieves predefined segmentation thresholds for the selected tissue type
	3	The DICOM Processing Module filters voxel data according to the thresholds and generates a segmented 3D volume
	4	The Browser Interface updates the visualization to show only the segmented anatomical structure
	5	The system enables the export options (STL or G-code) for the active region
Extensions	**Step**	**Branching Action**
	1a	If the user selects an invalid or unsupported tissue type, the system displays an error message and prompts for a valid selection
	3a	If segmentation parameters fail to apply (e.g., corrupted data), the system reverts to the previous view and logs the error

Open Issues	● Threshold values for each tissue type (bone, skin, muscle) must be confirmed and tested for accuracy across multiple DICOM datasets ● Decide whether users can define custom tissue thresholds
Verification Tests	1. Selecting “Bone” applies bone segmentation thresholds and updates visualization 2. Selecting “Skin” applies skin segmentation thresholds and updates visualization 3. Selecting “Muscle” applies muscle segmentation thresholds and updates visualization 4. Verify that switching between presets updates the visualization without reloading the DICOM file 5. Confirm STL export matches the visible region

Use Case -3: Preview Output

Number	3
Name	Visualize and Inspect 3D Anatomy
Summary	The system generates and displays a 3D preview of the selected anatomical region (e.g., bone, skin, or muscle) as a surface or volume rendering. The user can inspect the model interactively by rotating, zooming, and adjusting threshold levels to refine the visualization before export.
Priority	4
Preconditions	● Use Case 1 (DICOM File Upload and Parse) completed successfully ● Use Case 2 (Select Anatomical Region) completed successfully ● Segmented voxel data is available for rendering

Postconditions	• A 3D preview of the selected anatomy is rendered on screen • The user can adjust visualization parameters (e.g., threshold level, surface type). • The system stores the user's chosen settings for output generation	
Primary Actor	End user	
Secondary Actors	• Rendering Engine: WebGL/WebGPU - generates real-time surface or volume previews • Segmentation Engine: Supplies voxel intensity and region data for rendering. • UI Controller - Manages user interactions such as zoom, rotation, and threshold toggles.	
Trigger	The user selects "Preview Output" in the application interface.	
Main Scenario	**Step**	**Action**
	1	The system requests voxel and segmentation data from the Segmentation Engine
	2	The Rendering Engine generates an isosurface or volume preview of the selected anatomy
	3	The user inspects the 3D preview by rotating, panning, and zooming
	4	The user adjusts the intensity threshold or rendering mode(surface/volume).
	5	The preview updates dynamically based on the user's adjustments

	6	The user accepts the preview to confirm settings for export
Extensions	**Step**	**Branching Action**
	1a	If the input file is large or preview rendering exceeds 60 seconds, the system generates a lower-resolution preview first and notifies the user
	2a	If the rendering engine fails to initialize, the system displays an error message and provides troubleshooting guidance
	4a	If the user selects an invalid threshold value, the system resets to the previous valid configuration
Open Issues	● Define a target rendering time (e.g., preview should appear within 60 seconds for a 512x512 and/or 1024x1024 dataset ● Determine default threshold presets for each tissue type	
Verification Tests	1. Preview appears within 10 seconds on each supported browser 2. Rotating, zooming, and threshold changes update the 3D model in real time 3. Switching between surface and volume modes updates the visualization correctly 4. Accepting the preview stores current visualization parameters for output generation	

Use Case -4: Generate STL

Number	4
Name	Create Polygonal Mesh (.stl export)
Summary	The system converts the segmented anatomical region into a polygonal surface mesh using the Marching Cubles algorithm. The

	resulting mesh is optionally smoothed or repaired and then exported as an .stl file suitable for 3D printing or further modeling.	
Priority	5	
Preconditions	• Use Case 1 (DICOM Upload and Parse) and Use Case 2 (Select Anatomical Region) are complete • Segmentation thresholds and voxel intensity data are available • The system's Rendering and Marching Cubes modules are initialized	
Postconditions	• A polygonal mesh is generated in memory • Mesh geometry is optionally post-processed (smoothing, hole-filling, normal correction) • An .stl file is generated and downloaded to the user's system • Mesh statistics (vertex count, bounding box, and surface area) are displayed	
Primary Actor	End User	
Secondary Actors	• Marching Cubes Algorithm Module: Extracts the isosurface based on active intensity thresholds • Mesh Post-Processing Engine: Smooths and repairs the mesh to ensure printability. • File I/O System: Encodes the mesh data as an STL file and manages download • UI Controller: Handles user actions and displays generation progress	
Trigger	User clicks "Generate STL" in the interface	
Main Scenario	**Step**	**Action**
	1	The system retrieves segmented voxel data from memory

	2	The Marching Cubes Algorithm Module computes the polygonal isosurface for the selected anatomy
	3	The Mesh Post-Processing Engine smooths and repairs geometry
	4	The system calculates mesh statistics (vertex/triangle count, bounding box dimensions)
	5	The File I/O System encodes the mesh as an .stl file and initiates download
	6	The user verifies the exported file or proceeds to G-code generation
Extensions	**Step**	**Branching Action**
	2a	If voxel data is missing or corrupted, the system displays an error and aborts mesh generation
	3a	If post-processing fails, the system exports the raw mesh and warns the user
		If download fails due to browser security restrictions, the system offers a manual file-save option
Open Issues	Determine default smoothing parameters (number of iterations, smoothing, strength)	
Verification Tests	1. Output .stl is readable by standard slicers 2. Mesh bounds match CT bounds within tolerance	

Use Case -5: Generate G-code

Number	5
Name	Generate 3D Print Instructions(.gcode export)
Summary	The system converts the segmented anatomical model into printable G-code instructions using parameters provided by the G-code Generation Module. The G-code encodes extrusion paths and material parameters that replicate the density or opacity patterns of the original CT tissue data
Priority	4
Preconditions	● Use Case 1 (DICOM Upload and Parse) and Use Case 2 (Select Anatomical Region) are complete ● STL mesh data or voxel intensity data is available in memory ● The client-provided G-code Generation Module is integrated with the system
Postconditions	● G-code file is generated in memory and available for download ● Output metadata (e.g., estimated print time, layer count, and file size) is displayed to the user ● The system validates G-code syntax for standard 3D printer compatibility
Primary Actor	End User
Secondary Actors	● G-code Generation Module (Client-Provided): Converts model geometry and density information into printer-specific extrusion paths ● Hardware Profile Manager: Stores printer configuration presets (e.g., nozzle size, filament material) ● UI Controller: Managers download actions and user prompts

Trigger	User clicks “Generate G-code” in the application interface	
Main Scenario	**Step**	**Action**
	1	The user selects desired printer settings (e.g., nozzle diameter, material type) or accepts defaults
	2	The system sends the segmented mesh or voxel data to the G-code Generation Module
	3	The G-code Generation Module processes the data using tissue-mimicking density parameters
	4	The system validates the generated G-code format and syntax
	5	Estimated print metrics (e.g., layers, duration, material volume) are displayed
	6	The user confirms and downloads the .gcode file
Extensions	**Step**	**Branching Action**
	1a	If printer profile templates (e.g., nozzle size, layer height, filament type) are unavailable, the system applies default print parameters
	3a	If the algorithm fails the generate valid G-code, the system displays an error message and logs diagnostic data
	5a	If print metrics cannot be computed, the system allows manual download but warns the user

Open Issues	• Specific algorithm implementation details are pending from client • Define standard printer compatibility targets(e.g., PrusaSlicer, Cura) • Determine how tissue density maps to print settings (e.g., infill %, extrusion rate, material type)
Verification Tests	1. Generated .gcode file opens successfully in common slicers 2. File metadata matches expected layer count and print dimensions 3. G-code output is identical across supported browsers for the same input data and settings 4. Test prints produce expected density or opacity variations when printed with the appropriate filament

3. Non-Functional Requirements

Test Bench (for NFR-01/04/08/09): Desktop with ≥8 CPU cores @ ≥2.8 GHz, 16 GB RAM, SSD, and a GPU supporting WebGL2/WebGPU; OS: Windows 11, macOS (current), Ubuntu LTS; Browsers: latest stable Chrome, Firefox, Edge, and Safari. All verification below is performed on this bench unless stated otherwise.

NFR-01 (Performance) Priority-5

Description - The system shall process and convert a DICOM dataset to an STL or G-code file within (a) ≤ 120s for DICOM datasets ≤500 MB and (b) ≤240s for >500 MB to 1 GB.

Verification - Time 10 runs each on three benchmark studies per size bracket; the 95th-percentile duration must meet the target. Instrument start/stop via in-app timers and confirm with browser performance traces.

NFR-02 (Usability) Priority-4

Description - First-time users shall complete the task Import → Preview → Export STL unaided in ≤2 minutes with ≤10 interactions from the main screen; System Usability Scale ≥80. UI must meet WCAG 2.1 AA for keyboard navigation and contrast on all primary flows.

Verification - Moderated study with ≥10 representative users; collect completion time, interaction count, errors, and SUS; run automated WCAG checks and confirm 0 critical AA violations.

NFR-03 (Reliability) Priority-5

Description - For CT DICOM series with supported transfer syntaxes, the system shall complete processing without crash, freeze, or corrupted output in ≥99% of runs across a 200-case suite.

Verification - Execute the suite; record crash-free rate; validate exported STL loads successfully in a mesh validator, and pass manifold checks.

NFR-04 (Compatibility) Priority-4

Description - Functional behavior and outputs shall be equivalent on Chrome, Firefox, Edge, and Safari. Visual layout differences allowed ≤1 px; exported files must be byte-identical except for metadata timestamps.

Verification - Playwright-based cross-browser E2E runs; visual diff threshold ≤1 px and compare export hashes.

NFR-05 (Security & Privacy) Priority-5

Description - After initial static asset load, the app shall perform all computation locally and issue no network requests; no protected health information is persisted beyond the browser session unless the user explicitly exports a file; exported files must not contain embedded protected health information.

Verification - Run offline and confirm full functionality; capture traffic with DevTools and a local proxy like mitmproxy and check 0 requests during processing; inspect exports for metadata.

NFR-06 (Maintainability) Priority-3

Description - Codebase shall achieve Maintainability Index ≥70/100 and cyclomatic complexity ≤10 for ≥90% of functions; public APIs documented with generated docs like TypeDoc covering 100% of exported symbols.

Verification - Static analysis with ESLint, MI report, and docs generation; architecture review confirms clear module boundaries for parser, segmentation, renderer, exporter.

NFR-07 (Accuracy) Priority-5

Description - The generated STL and G-code outputs shall represent anatomical geometry within a ± 2% deviation of the source CT data.

Verification - Compare polygonal model geometry to CT voxel data using mesh validation tools.

NFR-08 (Scalability) Priority-2

Description - When handling datasets up to 1.0 GB, the UI shall remain responsive with main-thread Long Tasks <200 ms (95th-percentile) and interaction FPS ≥ 20 during orbit/zoom and slice scrubbing.

Verification - Stress tests with large real and synthetic studies; collect Long Task and FPS metrics via PerformanceObserver.

NFR-09 (Resource Efficiency) Priority-4

Description - For a 500 MB dataset, peak resident memory shall be ≤3× input size and ≤6 GB, average CPU utilization ≤80% across all cores, and any single main-thread task ≤200 ms during conversion and preview.

Verification - Profile with Chrome Performance and OS monitors, then record max RSS, CPU, and Long Task durations.

NFR-10 (Robustness) Priority-5

Description - For malformed or unsupported inputs like missing slices, inconsistent spacing, or unsupported transfer syntax, the system shall detect the issue and display a specific error ID and message within ≤1s of detection, without crashing or losing user data.

Verification - Feed ≥20 intentionally corrupted or incomplete DICOM files; verify 0 crashes, correct error IDs/messages, and that no partial exports are produced

4. User Interface

See “User Interface Design Document for Enabling 3D Printing of a Medical CT-Scan.” here.

[Will be linked when document is created]

5. Deliverables

Provide a list of all deliverable items (that is, all artifacts that you will deliver to the customer). This list will include items such as the product itself (What format? Source code? Executable code? Object code?), documentation, and training resources (if any). Specify when (date) and in what format (e.g., hard copy, zip file (how delivered, Git?)) each will be delivered. A tabular format works well for this section. We will assume that the deliverable items are as follows:

Hard copies of each of the following: ***[Will be linked when document is created]***

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- User Manual
- Administrator Manual
- Copies of all Biweekly Status Reports

An electronic file containing the following:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- User Manual
- Administrator Manual
- All source code
- The executable program
- Any other software required for installation and execution of the delivered program.

6. Open Issues

Issues that have been raised and do not yet have a conclusion. These issues will be addressed later in the development process.

Appendix A – Agreement Between Customer and Contractor

This Software Requirements Specification document constitutes a formal agreement between the Customer, Terry Yoo, and the Contractor, The Slice Is Right, regarding the functional and non-functional requirements for the Enabling 3D Printing of a Medical CT-Scan system. By signing below, both parties acknowledge that this document accurately and completely captures the mutual understanding of the system to be developed. The Customer agrees that this SRS provides a sufficient basis for the Contractor to proceed with the system design and implementation, and the Contractor agrees to develop a system that conforms to the requirements described above.

Any future changes, additions, or modifications to the requirements specified in this document must be managed through a formal change control process. A request for change must be submitted in writing by either party and will be evaluated for its impact on project scope, schedule, and feasibility. An amended version of this SRS, or a formal change order referencing this document, must be mutually agreed upon and signed by authorized representatives of both the Customer and the Contractor before any changes are implemented in the project.

Name	Signature	Date	Comments
Cooper Stepankiw			
Bryan Sturdivant			
Israk Arafat			
Greg Michaud			
Ethan Wyman			
Terry Yoo			

Appendix B – Team Review Sign-off

This document confirms that all undersigned members of the The Slice Is Right project team have thoroughly reviewed the entirety of this Software Requirements Specification for the Enabling 3D Printing of a Medical CT-Scan system. By signing below, each team member acknowledges their understanding of the requirements and formally agrees that the content, scope, and structure of this document are accurate and complete, providing a suitable foundation for the subsequent phases of the project.

The comment section provided for each team member is to be used for noting any minor suggestions, editorial feedback, or non-substantive points of clarification. It is recognized that for this sign-off to be granted, there are no major points of contention regarding the technical or functional requirements outlined within this SRS.

Name	**Signature**	**Date**	**Comments**
Cooper Stepankiw			
Bryan Sturdivant			
Israk Arafat			
Greg Michaud			
Ethan Wyman			
Terry Yoo			

Appendix C – Document Contributions

This appendix details the contributions of each team member to the creation of this Software Requirements Specification. All members participated in the collaborative writing, review, and diagramming process to ensure a comprehensive and unified document. The percentage contributions are estimates that reflect the primary authorship and development effort for the various sections.

Member Name	Primary Responsibilities	Estimated Percentage
Israk Arafat	Functional Requirements	20%
Gregory Michaud	Introduction Section	20%
Cooper Stepankiw	Title Cover Team Logo Section 4,5 Appendix A,B,C	20%
Bryan Sturdivant	Functional Requirements	20%
Ethan Wyman	Section 3; Non-Functional Requirements Table of Contents Formatting	20%
Total		100%

Appendix D – System Design Document (SDD)

System Design Document

Enabling 3D Printing of a Medical CT-Scan:

A Web App for Patients and Practitioners

Prepared by The Slice Is Right

COS 397 - Computer Science Capstone 1

November 10th, 2025

Version 1.0.0

Table of Contents

1 Introduction

This capstone project, completed as part of the requirements for the Bachelor of Science in Computer Science at the University of Maine, focuses on developing a web application called *"Enabling 3D Printing of a Medical CT-Scan: A Web App for Patients and Practitioners."* The goal of this project is to create a browser-based tool that gives users the ability to upload medical CT scans in the DICOM format, produce a viewable 3D model, and turn them into files that can be used for 3D printing. The application is designed to produce both polygonal (.stl) files, which can be used to visualize different anatomical structures like bone, skin, or muscle, and G-code files, which are needed for 3D printers to accurately reproduce tissue characteristics. One of the most important aspects of this project is that all processing is done locally on the user's computer, which helps protect sensitive medical data and makes the tool more accessible for both medical research and patient education. By working on this project, we are contributing to the University of Maine's ongoing efforts to advance medical imaging and 3D printing, especially through the Laboratory for Convergent Science.

1.1 Purpose of This Document

The main goal of this System Design Document (SDD) is to explain, in technical terms, how the system described in the Software Requirements Specification (SRS) will actually be built. In other words, this document takes the requirements and turns them into a clear plan for development, covering things like the system's architecture, how data will be organized, and how the different parts of the project will be broken down. This SDD is written for Dr. Terry Yoo, the University of Maine Computer Science faculty, and everyone on *The Slice Is Right* development team. It is meant to be a guide for how the web application, "Enabling 3D Printing of a Medical CT-Scan: A Web App for Patients and Practitioners," will be put together, including the main structure, how the different components will work together, how data will move through the system, and how files will be organized. By laying all of this out, the SDD helps make sure that everyone involved has the same understanding of how the design meets the requirements in the SRS, and that there is enough technical detail for building, testing, and maintaining the system in the future.

1.2 References

Primary Project Source

Yoo, Terry. *Enabling 3D Printing of a Medical CT-Scan: A Web App for Patients and Practitioners.* University of Maine, Laboratory for Convergent Science, 2025.

Web and Technical References

- "DICOM Standard." National Electrical Manufacturers Association (NEMA). Available at: https://www.dicomstandard.org/
- "Polygonizing a Scalar Field(Marching Cubes)." Paul Bourke. Available at: https://paulbourke.net/geometry/polygonise/
- "STL (File Format) Specification." 3D Systems. Available at: https://www.3dsystems.com/support
- "G-code Reference." RepRap Wiki. Available at: https://reprap.org/wiki/G-code websites).
- "JavaScript library for scientific visualization"(vtk.js). Kitware. Available at: https://kitware.github.io/vtk-js
- "Vtk-s Github Repository." Kitware. Available at: https://github.com/Kitware/vtk-js/tree/master
- "Selenium Automates Browsers." Selenium. Available at: https://www.selenium.dev/

2 System Architecture

This section defines how the system is organized and why. In 2.1, Architectural Design, we present the overall runtime approach and key quality attributes, along with the logical and technology views. In 2.2, Decomposition Description, we break the system into concrete parts with clear responsibilities and interfaces.

2.1 Architectural Design

The system uses a client-only, browser-resident architecture to keep CT data private and simplify deployment.

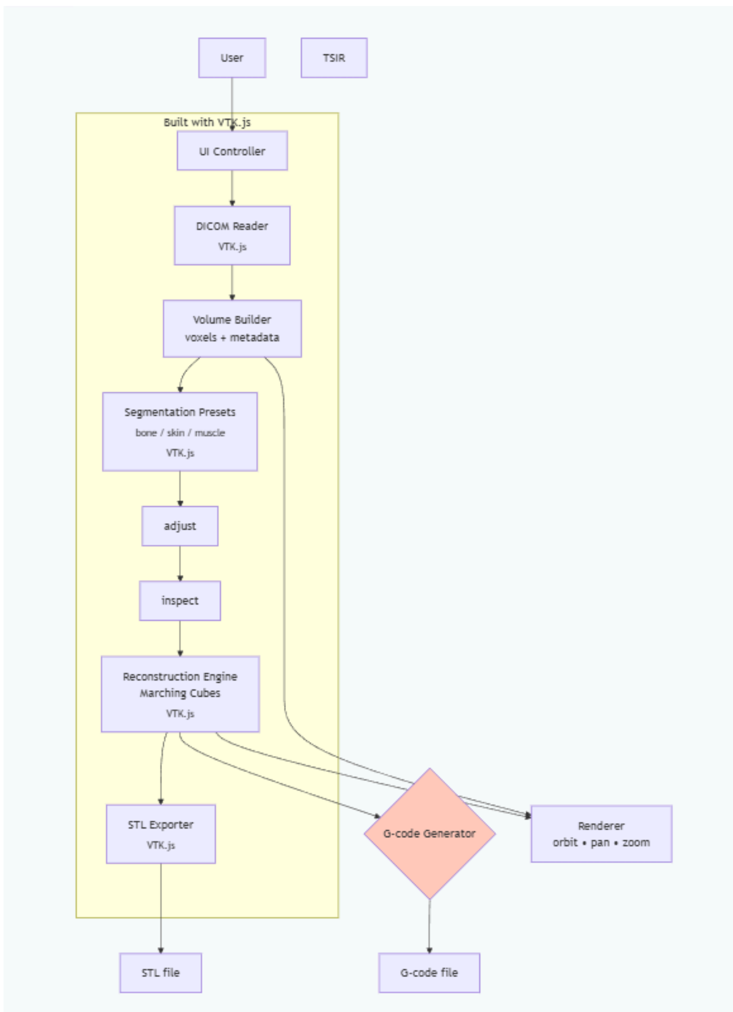


Figure 2.1 - Logical architecture diagram

As shown in the diagram, the logical flow of The Slice Is Right runs entirely on the user's device, with its core components built upon the open-source VTK.js library. The process begins with the VTK.js DICOM Reader, which ingests and validates data to produce a voxel volume.

VTK.js Segmentation Presets then apply Hounsfield Unit (HU) thresholds to this volume to create a tissue mask. The VTK.js Reconstruction Engine, utilizing the Marching Cubes algorithm, extracts a triangle mesh for visual preview within the Renderer. From this point, the user can choose to export the model. The VTK.js STL Exporter saves the mesh as an STL file directly. Alternatively, the generated mesh is passed to an external G-code Generator, a component over which The Slice Is Right has no control. The UI mediates each step, allowing users to adjust thresholds, re-run reconstructions, and visually confirm results before saving. No runtime network traffic occurs, and the only artifacts written are the output files the user explicitly chooses to save.

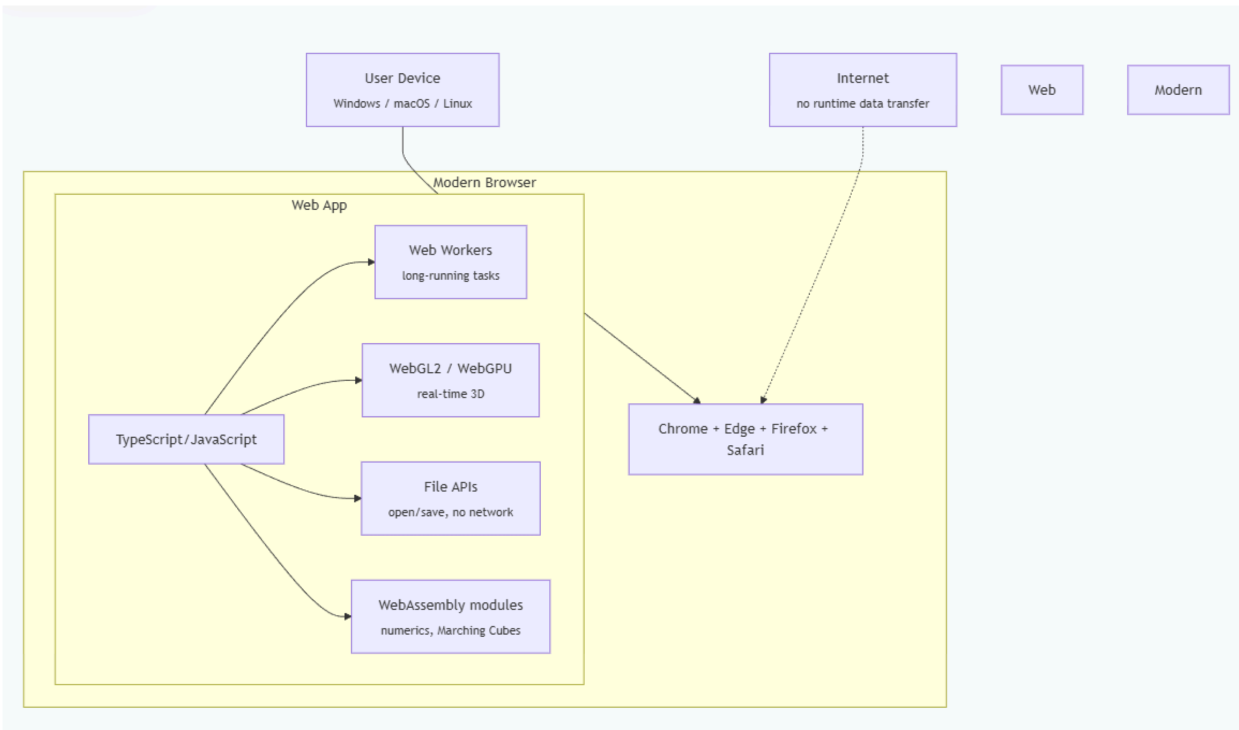


Figure 2.2 - Technology architecture diagram

Figure 2.2 Illustrates the technology stack inside a modern browser. Application logic is written in TypeScript, while long-running computations are executed in Web Workers to maintain a responsive interface. Vtk.js handles performance-critical tasks such as surface extraction via the marching cubes algorithm and 3D rendering, leveraging WebGL2 for GPU acceleration and optionally using WebGPU when available. Browser file APIs manage local file open and save to enable offline operation. This design targets the latest versions of Chrome, Edge, Firefox, and Safari on Windows, macOS, and Linux, balancing privacy and portability while providing sufficient performance for typical study data sizes.

Pending decisions: default WebGL2 vs. opportunistic WebGPU; whether G-code profiles are fixed templates or user-saved.

2.2 Decomposition Description

This section breaks down the system into its main parts. The entire application runs in a web browser on the user's own computer. This design keeps data private and allows for offline use. The main parts work together to manage the steps from loading a medical scan to exporting a 3D model. These include the UI Controller that manages the different steps, the processing units that handle data and complex calculations, and the visual display that shows the 3D model. The following sections explain what each of these parts does and how they interact.

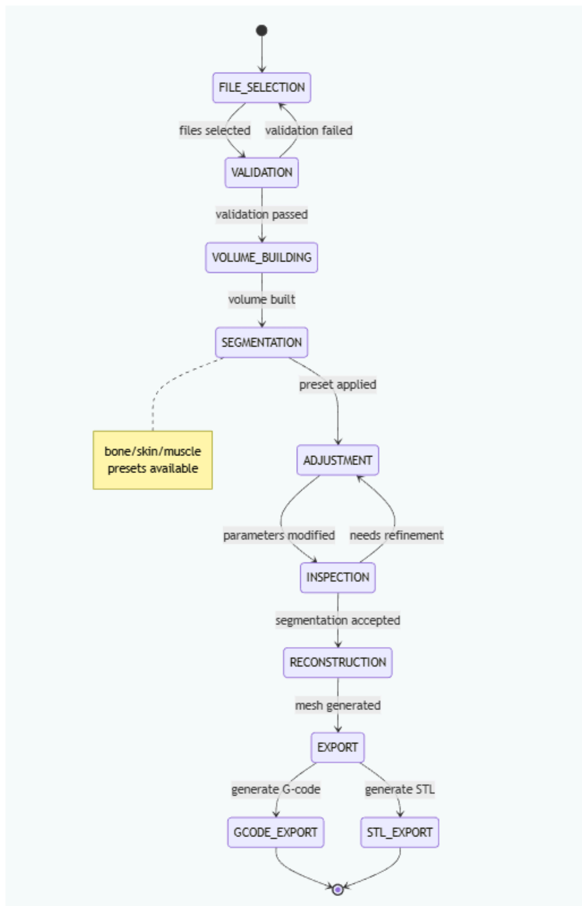


Figure 2.3

The UI Controller manages application flow through distinct states that correspond to different processing stages. It begins in FILE_SELECTION where users provide input data. Upon file selection, it transitions to VALIDATION, ensuring DICOM compatibility before proceeding to VOLUME_BUILDING where 3D voxel data is constructed. The controller then enters the SEGMENTATION phase, offering preset tissue types (bone/skin/muscle) for initial processing. This leads to an adjustable loop between ADJUSTMENT and INSPECTION states, allowing users to refine parameters and preview results iteratively until satisfied. Once segmentation is approved, the RECONSTRUCTION state generates the 3D mesh using the Marching Cubes algorithm. Finally, the controller transitions to EXPORT options where users can choose between GCODE_EXPORT for 3D printing or STL_EXPORT for model sharing. Each state maintains clear boundaries with defined entry/exit conditions, ensuring smooth workflow progression while allowing backward transitions for refinement when needed.

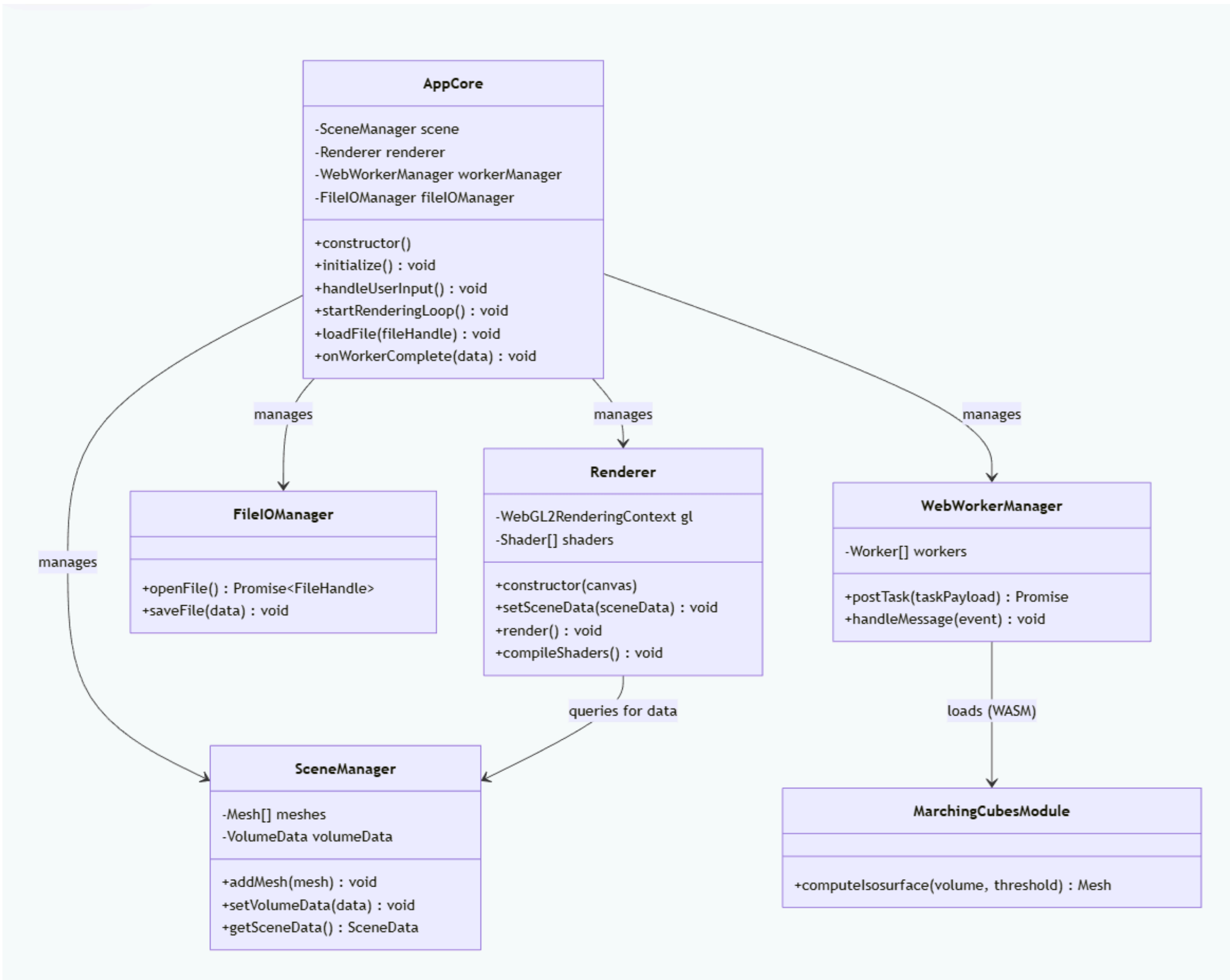


Figure 2.4

Based on the technology architecture described in the Architectural Design, the class diagram elaborates on the implementation within the Main App and Renderer. The AppCore class acts as the central Controller, initializing the system and managing the flow of data. It depends on the SceneManager (the Model) to maintain the state of the 3D scene and volume data. The Renderer (the View) is responsible for taking the scene graph from the SceneManager and drawing it to the screen. The WebWorkerManager abstracts the complexity of dealing with concurrent threads and is responsible for loading and executing the WebAssembly module for the performance-critical MarchingCubesModule. The FileIOManager encapsulates all interactions with the browser's file system. This hybrid structure effectively balances performance, organization, and the constraints of the modern web environment.

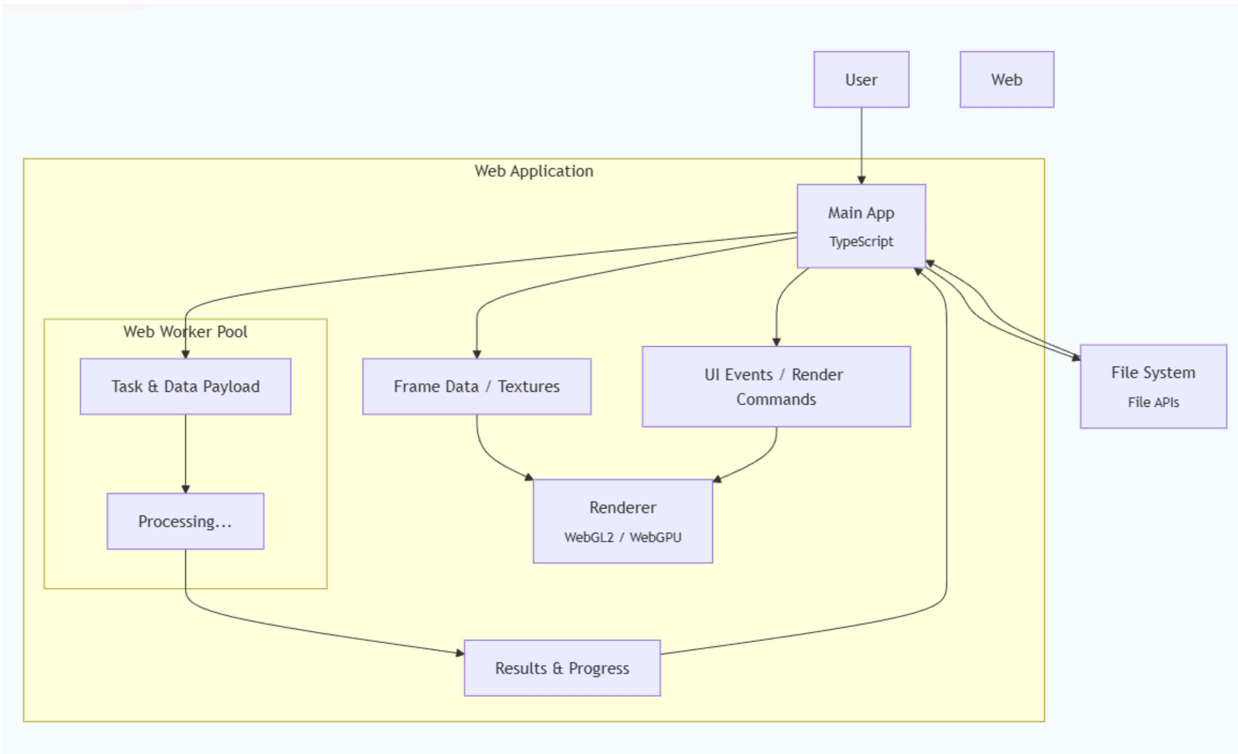


Figure 2.5

Based on the technology architecture described in the Architectural Design, the structural diagram communicates the system's runtime architecture, highlighting concurrent execution and data flow. The Main App handles the user interface and central coordination. It dispatches computationally intensive tasks, like running the Marching Cubes algorithm, to the Web Worker Pool to maintain UI responsiveness. The worker sends results back to the main thread, which forwards scene data to the Renderer. The Renderer, utilizing WebGL2 or WebGPU, processes this data to generate real-time 3D visuals. All data persistence is handled locally via the File System APIs, completing the offline operation capability.

3 Persistent Data Design

3.1 Client-Side History Storage

To support session persistence and user history without requiring server-side storage, the application leverages browser cookies. Cookies are used to store metadata such as recently loaded DICOM files, user-selected anatomical regions and settings, or application preferences. This allows the system to restore the user's state between sessions while maintaining privacy, since no data is sent over the network.

Key points:

- **Scope**: Cookies are limited to storing small amounts of non-sensitive information (e.g., file names, timestamps, user preferences).
- **Implementation**: The application reads and writes cookies via standard browser APIs when files are loaded, saved, or modified.
- **Retention**: Cookies are configured with an appropriate expiration period to maintain history while avoiding indefinite storage.

3.2 File Descriptions

The system processes and generates three primary file types: **DICOM**, **STL**, and **G-code**. Each file type is handled entirely on the client side using browser APIs, with no network or database storage. The following subsections describe each file's structure, purpose, and data fields relevant to implementation.

DICOM Input Files

File Type: .dcm
Source: User-supplied medical CT scan (DICOM standard compliant)
Purpose: Represents the raw input data from which anatomical reconstructions are generated.

Each DICOM file contains one image "slice" with embedded metadata that defines pixel intensity values (Hounsfield Units), orientation, spacing, and acquisition information. When multiple slices are provided, the system reconstructs them into a volumetric dataset (3D voxel array).

Structure Overview

Field Name	Data Type	Description
PatientID	string	Unique identifier for the study (removed before processing to ensure privacy).
SeriesInstanceUID	string	Unique series identifier used to group slices into a 3D volume.
PixelData	uint16[][]	2D array of voxel intensity values representing one CT slice.
Rows, Columns	int	Dimensions of the pixel grid for each slice.
SliceThickness	float	Distance (mm) between consecutive slices.
PixelSpacing	float[2]	Physical spacing (mm) between adjacent pixels within a slice.
RescaleIntercept, RescaleSlope	float	Parameters for converting raw pixel data to Hounsfield Units.
ImageOrientationPatient, ImagePositionPatient	float[6], float[3]	Defines spatial orientation and position for reconstructing the 3D volume.

STL Output Files

File Type: .stl (ASCII or Binary)

Purpose: Encodes the 3D polygonal surface mesh derived from segmented CT data.

The STL file describes a 3D geometry using a collection of triangular facets, each defined by three vertex coordinates and a surface normal vector. Binary STL is used by default for compactness.

Structure Overview (Binary STL)

Field Name	Data Type	Description
Header	char[80]	Optional text header describing export parameters.
NumTriangles	uint32	Number of triangular facets in the mesh.
NormalVector	float[3]	Unit normal vector for each triangle.
Vertex1, Vertex2, Vertex3	float[3] each	Coordinates of the triangle's vertices in model space.
AttributeByteCount	uint16	Optional per-triangle attribute (set to zero).
PixelSpacing	float[2]	Physical spacing (mm) between adjacent pixels within a slice.
RescaleIntercept, RescaleSlope	float	Parameters for converting raw pixel data to Hounsfield Units.
ImageOrientationPatient, ImagePositionPatient	float[6], float[3]	Defines spatial orientation and position for reconstructing the 3D volume.

G-code Output Files

File Type: .gcode
Purpose: Encodes printer instructions for fabricating physical models that replicate the geometry and X-ray attenuation properties of tissue.

Each G-code file contains a sequence of printer movement and extrusion commands derived from the segmented mesh or voxel data. These files can be opened directly in common slicers or sent to standard 3D printers.

Structure Overview

Field Name	Data Type	Description
Header	string	Printer configuration and initialization commands.
LayerCommands	list	Sequence of movement and extrusion commands for each print layer.
Footer	string	Shutdown commands to finalize the print.
Comments	string	Optional notes describing density mapping or export parameters.
AttributeByteCount	uint16	Optional per-triangle attribute (set to zero).
PixelSpacing	float[2]	Physical spacing (mm) between adjacent pixels within a slice.
RescaleIntercept, RescaleSlope	float	Parameters for converting raw pixel data to Hounsfield Units.
ImageOrientationPatient, ImagePositionPatient	float[6], float[3]	Defines spatial orientation and position for reconstructing the 3D volume.

4 Requirements Matrix

The requirements matrix provides a structured overview of the system's functional and non-functional requirements, linking each requirement to the modules and components responsible for its implementation. This matrix serves as a reference to ensure that all specified behaviors and constraints are addressed in the design and development process.

Each entry in the matrix includes:

- **Requirement ID**: A unique identifier for traceability
- **Requirement Name**: A unique name
- **Primary Implementation Modules**: The system components responsible for fulfilling the requirement

To provide both high-level and detailed perspectives, the requirements matrix is presented in two formats:

- Concise Requirements Matrix
    - This table offers a high level-mapping of each functional requirement to the primary modules responsible for its implementation. It is intended for quick reference and traceability.
- Detailed Requirements Matrix
    - This table expands on the concise version by breaking down each requirement into specific steps, associated UI elements, and the responsibilities of each module during those steps. It provides a stepwise view of system behavior for design, implementation, and testing purposes.

Concise Requirements Matrix:

UC ID	**Functional Requirement Name**	**Primary Implementation Modules**
UC-01	Import DICOM File	DICOM Parser Module
UC-02	Select Anatomical Region	Segmentation Module, DICOM Parser Module, Rendering Engine
UC-03	Visualize and Inspect 3D Anatomy	Segmentation Module, Rendering Engine, DICOM Parser Module
UC-04	Create Polygonal Mesh	Marching Cubes Module, Segmentation Module,

		DICOM Parser Module
UC-05	Generate 3D Print Instructions	G-Code Generator Module, Segmentation Module, DICOM Parser Module

Detailed Requirements Matrix:

Use Case/ Functional Requirement	**UI/Frontend**	**Dicom Parser Module**	**Segmentation Module**	**Rendering Engine**	**Marching Cubes Module**	**G-Code Generator Module**
UC-1: Import Dicom File						
Allow user to select and upload a DICOM file or folder	File upload button, selection and drag-and-drop interface					
Validate DICOM format before processing	Display progress and errors	Validates DICOM files				
Parse DICOM data to extract voxel intensity and metadata		Reads slices, extracts voxel and metadata				
Notify user of success or failure	Pop-up status messages	Returns validation result				
UC-2: Select Anatomical Region						

Allow user to select tissue type (bone, skin, muscle)	Dropdown for tissue selection	Provides metadata for thresholds	Applies segmentation thresholds	Updates 3D preview dynamically		
Filter CT data based on selected tissue		Supplies voxel intensity data	Threshold filtering logic	Visualizes filtered data		
Update visualization automatically when region changes	Handles selection from user		Applies new mask	Rerenders visualization		
UC-3: Visualize 3D Anatomy						
Generate and display 3D preview	“Preview” button and controls	Provides volume data	Provides segmentation data	Renders model		
Allow user to rotate, zoom, and inspect model interactively	Camera and interactions controls			Real-time rendering		
Update visualization dynamically based on user input	Threshold sliders	Provides data	Recomputes segmentation	Updates rendered output		
UC-4: Generate STL File						
Allow user to select STL (.stl) output type	Dropdown file type selection					
Apply Marching Cubes to extract polygonal surface		Supplies Voxel Grid	Provides segmented mask		Surface Extraction	
Generate and export 3D mesh (.STL)	“Export STL” Button		Supplies final segmentation		Encodes STL Mesh	
Notify user when file generation is complete or failed	Pop-up status message				Reports Success / Error	

UC-5: Generate G-Code File						
Allow user to select G-Code (.gcode) output type	Dropdown file type selection					Initializes Module
Generate G-Code reproducing tissue density/opacity		Supplies Voxel/ Mesh data	Provides tissue density mapping			Converts to printer-ready G-code
Validate and save G-Code to user location	“Save G-Code” button					Performs Syntax Validation
Notify user when file generation is complete or failed	Pop-up status message					Performs validation

Appendix A – Agreement Between Customer and Contractor

This System Design Document constitutes a formal agreement between the Customer, Terry Yoo, and the Contractor, The Slice Is Right, regarding the architectural and detailed design for the Enabling 3D Printing of a Medical CT-Scan system. By signing below, both parties acknowledge that this document accurately translates the requirements specified in the Software Requirements Specification into a technical blueprint for system construction. The Customer agrees that this design provides a sufficient and appropriate basis for the system's implementation, and the Contractor agrees to develop a system that faithfully adheres to the design described herein.

Any future changes, additions, or modifications to the design specified in this document must be managed through a formal change control process. A request for change must be submitted in writing by either party and will be evaluated for its impact on the system's architecture, project scope, schedule, and feasibility. An amended version of this SDD, or a formal change order referencing this document, must be mutually agreed upon and signed by authorized representatives of both the Customer and the Contractor before any design changes are implemented in the project.

Name	**Signature**	**Date**	**Comments**
Cooper Stepankiw			
Bryan Sturdivant			
Israk Arafat			
Greg Michaud			
Ethan Wyman			
Terry Yoo			

Appendix B – Team Review Sign-off

This document confirms that all undersigned members of The Slice Is Right project team have thoroughly reviewed the entirety of this System Design Document for the Enabling 3D Printing of a Medical CT-Scan system. By signing below, each team member acknowledges their understanding of the proposed system architecture and detailed design and formally agrees that the content, scope, and technical direction outlined in this document are accurate, feasible, and provide a suitable foundation for the implementation phase of the project.

The comment section provided for each team member is to be used for noting any minor suggestions, editorial feedback, or non-substantive points of clarification. It is recognized that for this sign-off to be granted, there are no major points of contention regarding the architectural or detailed design decisions outlined within this SDD.

Name	Signature	Date	Comments
Cooper Stepankiw			
Bryan Sturdivant			
Israk Arafat			
Greg Michaud			
Ethan Wyman			
Terry Yoo			

Appendix C – Document Contributions

This appendix details the contributions of each team member to the creation of this System Design Document. All members participated in the collaborative design process, including architectural planning, component specification, diagramming, writing, and review to ensure a comprehensive and technically sound document. The percentage contributions are estimates that reflect the primary authorship and development effort for the various sections and diagrams.

Member Name	Primary Responsibilities	Estimated Percentage
Israk Arafat	Architectural Design	20
Gregory Michaud	Introduction Sections	20
Cooper Stepankiw	Section 2.2 Appendix A,B,C	20
Bryan Sturdivant	Requirements Matrix	20
Ethan Wyman	Requirements Matrix, Formatting	20
Total		100%

Appendix E– User Interface Design Document (UIDD)

User Interface Design Document

Enabling 3D Printing of a Medical CT-Scan:

A Web App for Patients and Practitioners

Prepared by The Slice Is Right

Collaborators: Bryan S, Cooper S, Ethan W, Greg M, Israk A

Client: Terry Yoo

COS 397 - Computer Science Capstone 1

December 3rd, 2025

Version 1.0.0

Table of Contents

1. Introduction

The purpose of this project is to create a browser-based, client-side web application that allows users to upload medical CT scans in DICOM format, view the volumetric data, and generate anatomical models that can be printed in both STL and G-code formats. One of the main goals is to address the client's need for research software that can convert CT data into 3D-printable models of bone, skin, and muscle, while also making sure that all medical data stays on the user's local computer for privacy and security reasons. To achieve this, the tool will make use of open-source components, such as DICOM readers, Marching Cubes surface extraction, STL writers, and custom algorithms for generating G-code, adapting or extending them as needed to support medical imaging research in the Laboratory for Convergent Science. The client is looking for a way to make it easier to select tissues, preview models, and export files for use in clinical, educational, and research settings. This document will describe the user interface that will help accomplish these tasks, and it is being submitted as part of the capstone requirement for the Bachelor of Science in Computer Science at the University of Maine.

1.1 Purpose of This Document

This User Interface Design Document serves to outline and explain the visual and interaction design choices made for the *Enabling 3D Printing of a Medical CT-Scan* system. It is intended for a range of readers, including the customer, course instructors, and the development team known as The Slice Is Right. The document lays out the standards for the interface, including how screens are organized, how users will navigate through the system, and the rules for entering data. These elements are all designed to support the workflow that was described in the project proposal, which involves selecting CT data, choosing specific tissue types, previewing the results, and finally exporting either STL or G-code files. By establishing these specifications, the document aims to guide the implementation process and to make sure that the interface remains consistent, easy to use, and maintainable across all browsers and operating systems that are supported.

1.2 References

Primary Project Source

Yoo, Terry. *Enabling 3D Printing of a Medical CT-Scan: A Web App for Patients and Practitioners.* University of Maine, Laboratory for Convergent Science, 2025.

Web and Technical References

- “DICOM Standard.” National Electrical Manufacturers Association (NEMA). Available at: https://www.dicomstandard.org/
- Lorensen, William E., and Harvey E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm.” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, 1 Aug. 1987, pp. 163–169, https://doi.org/10.1145/37402.37422.
- “STL (File Format) Specification.” 3D Systems. Available at: https://www.3dsystems.com/support
- “G-code Reference.” RepRap Wiki. Available at: https://reprap.org/wiki/G-code websites).
- “JavaScript library for scientific visualization”(vtk.js). Kitware. Available at: https://kitware.github.io/vtk-js
- “Vtk-s Github Repository.” Kitware. Available at: https://github.com/Kitware/vtk-js/tree/master
- “Selenium Automates Browsers.” Selenium. Available at: https://www.selenium.dev/
- “ITK-Wasm.” *ITK-Wasm*, 2024, docs.itk.org/projects/wasm/en/latest/. Accessed 4 Dec. 2025.

2. User Interface Standards

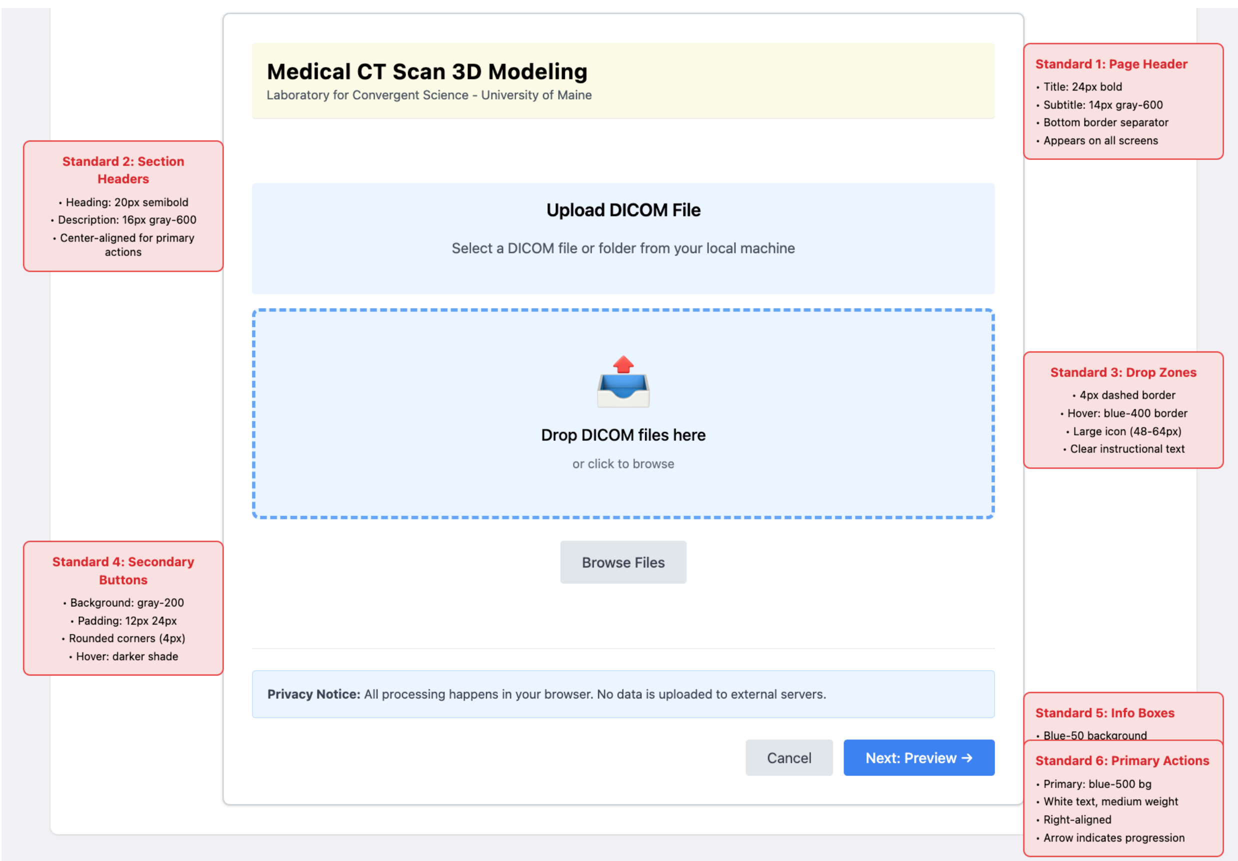
The user interface standards for the Slice Is Right system are outlined in detail. The primary purpose of setting these standards is to create a sense of consistency, predictability, and overall usability throughout every part of the web-based application. By defining the general layout, navigation structure, commonly used components, and the patterns for handling errors, these standards serve as the foundation for the entire system. Establishing these conventions is important because it allows users to have a unified and reliable experience as they move through tasks such as uploading CT scans, previewing anatomical structures, and generating printable output files. It is important to note that all of the screen designs, walkthroughs, and workflows that will be discussed in later sections are based on the principles and guidelines that are introduced here.

2.1 General Screen Layout

All screens in the system will follow a shared structural template consisting of:

1. Header Bar: Displays the application title, current file status(e.g., “No DICOM Loaded”), and quick-access icons.
2. Navigation Panel: Provides access to primary actions including Upload, Preview, Export, and Settings.
3. Main Content Area: Dynamically changes depending on the selected module (upload form, viewer canvas, Export selections, Settings).
4. Message/Alert Popup: Used for success notifications, warnings, and general errors.

Figure 2.1: General Screen Layout



Early conception of the General Screen Layout Including a DICOM file uploader with descriptions on actions and future existing items

2.2 Common Interface Components

- The following reusable components will be used throughout the system:
- Primary Action Buttons(e.g., “Upload”, “Preview”, “Export”):
    - Located in the top navigation bar. These represent the main actions users take within each module and follow a consistent color and style hierarchy.
- Secondary Buttons (e.g., “Reset”, “Cancel”, “Clear”)
    - Found within forms, upload areas, and settings panels. These perform reversible or less critical actions and are styled distinctly from Primary Buttons.
- Drop-down Selectors
    - Selectors follow consistent interaction patterns and validation rules. Used for choosing configurable options such as:
        - Slice thickness
        - Export resolution
        - Algorithm selection
        - Reconstruction or meshing presets
- File Input Component
    - A standardized uploader for importing .dcm files or zipped DICOM sets. This component includes file validation, drag-and-drop support, and visible progress reporting.
- 3D Viewer Canvas (VTK.js)
    - The interactive visualization window where CT slices, meshes, and thresholded anatomy are displayed. This component supports rotation, zooming, HU filtering, and slice navigation.
- Progress Indicators
    - Displayed during operations such as file upload, DICOM parsing, mesh generation, and STL export. These may appear as inline spinners, a progress bar, or step indicators.
- Hounsfield Unit (HU) Adjustment Slider
    - A real-time slider component that allows users to set HU window/level thresholds. Adjusting the slider updates the viewer immediately and is used for isolating tissue types (bone, skin, muscle).
- Viewer Interaction Controls
    - Standardized interaction methods available in all viewer-related modules:
        - Mouse wheel: zoom in/out
        - Click + drag: 3D rotation
        - Arrow keys: fine-grain rotation adjustments
        - Slice sliders: navigate through slices
    - These controls maintain consistent behavior across tools so users do not need to relearn interactions.

- Modal Dialogs
    - Used for critical confirmations (e.g., overwriting an export), platform warnings, and operations that require explicit user acknowledgement.

Each module reuses these components to ensure predictable behavior and consistent visual language throughout the application. .

2.3 Navigation Standards

Navigation is intentionally simple and task-oriented:

- A persistent top navigation bar that allows switching between Upload, Preview, Export, and Settings at any time
- Navigation should not interrupt long-running tasks such as DICOM parsing or mesh generation. Users should be able to view progress without losing work.
- Cancel and back options must be available in all modules where user actions can be undone.
- Viewer-specific controls (rotation, zoom, HU slider, slice sliders) remain active when navigating between preview-related submodules to ensure continuity.

2.4 Error Handling Standards

Errors will be communicated in a consistent manner across all screens:

- Inline Error Messages
    - Displayed under specific fields when invalid input is detected (e.g., invalid slice thickness, unsupported file type).
- Alert Bar Messages: Appear at the top of the content area for warnings, such as:
    - “Failed to parse DICOM header”
    - “Upload failed”
    - “STL conversion failed”
- Toast Notifications:
    - Used for positive or informational events (e.g., “STL exported successfully”).

Errors will always explain the issue and provide remediation guidance when possible.

2.5 Data Persistence Standards

The system uses lightweight browser-based persistence (cookies or local storage) to retain non-sensitive user preferences without requiring a login. This supports a smooth workflow by preserving:

- Last used HU slider values
- Preferred viewer orientation or slice position
- Recently used export settings (resolution, file type)
- Recently uploaded DICOM file

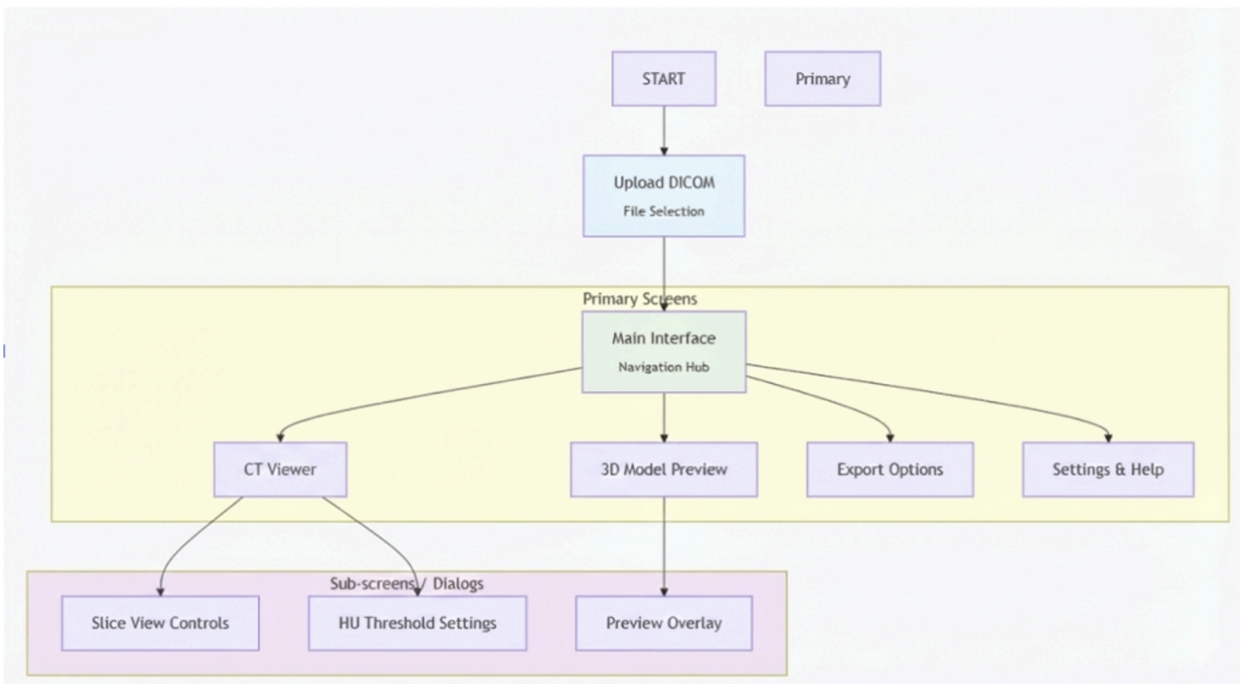
Persistent data never includes personal information; only settings that improve usability are stored.

3. User Interface Walkthrough

This section provides a comprehensive walkthrough of the primary user interface screens that make up the system. It begins with a navigation diagram illustrating how users move between major system components and continues with detailed screen mockups that demonstrate the layout and function of each interface. The walkthrough explains how users upload DICOM files, select tissue types, preview the resulting 3D models, and export files in STL or G-code format. All screens shown in this section follow the standards defined in Section 2, ensuring a consistent user experience across the application.

3.1 Navigation Flow

Figure 3.1 Navigation Flow



Typical Navigation of a user interacting with the system including where they would start and what would take the user where.

Application Purpose: A medical CT scan 3D modeling application that enables users to convert imaging data into 3D-printable models.

Core Workflow: The user flow begins with uploading DICOM files and progresses through visualization, segmentation, and export.

Key Feature - Data Privacy: All processing is performed client-side, ensuring no medical data is transferred over the network.

Navigation Structure: The app uses a hub-and-spoke model, centered around a "Main Interface / Navigation Hub."

Start Point: The journey begins at the "Upload DICOM / File Selection" screen.
Primary Screens: From the main hub, users can access four core modules:

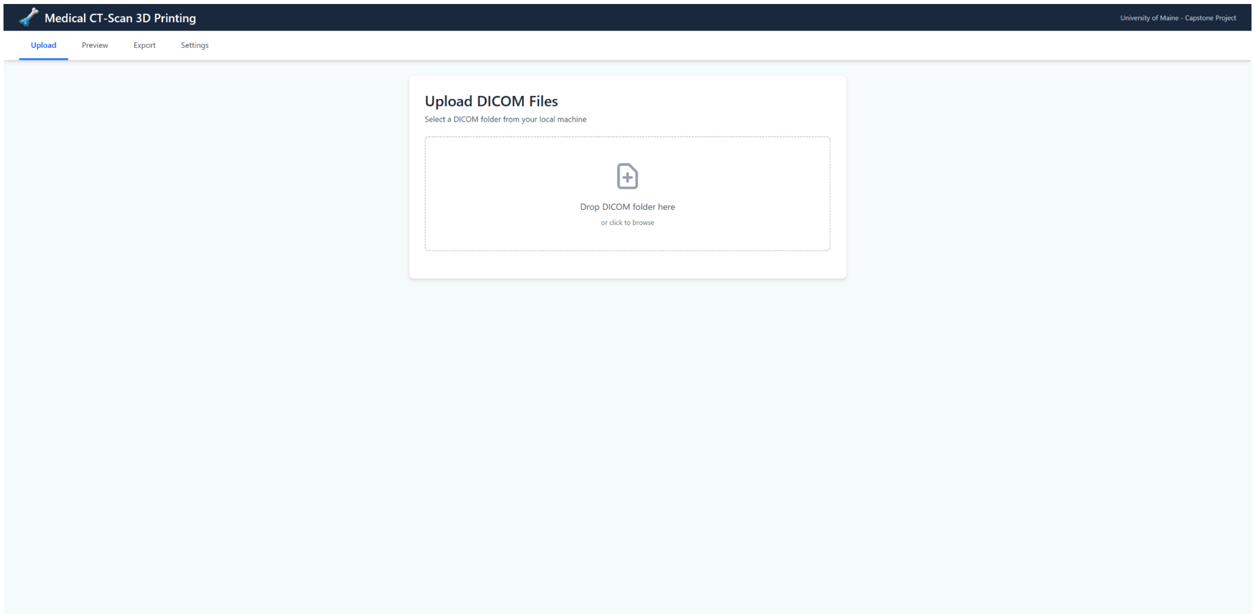
- Upload DCIM
- CT Viewer
  - Tissue Selection (Segmentation)
  - 3D Model Preview
- Export Options
- Settings & Help

Sub-Screens: Primary screens often contain specialized sub-screens for detailed tasks, such as "Slice View Controls" and "HU Threshold Settings" for the CT Viewer

3.2 Screen Walkthrough

Home Page

Figure 3.2 DICOM Uploader



Main Upload Area where the user can upload a DICOM scan into the dropbox area to allow the system to process it

Primary Method: A large, central interactive zone with the text "Drop DICOM files here or click to browse".

Functionality: This area supports the two standard methods for file input:

Drag-and-Drop: Users can drag one or more DICOM files or a folder from their desktop directly into this zone.

Click to Browse: Clicking anywhere in this zone will open the system's native file explorer dialog.

Action Button:

"Browse Files" Button: A secondary, explicit button to trigger the file browser. This provides a familiar, unambiguous alternative to clicking the upload zone.

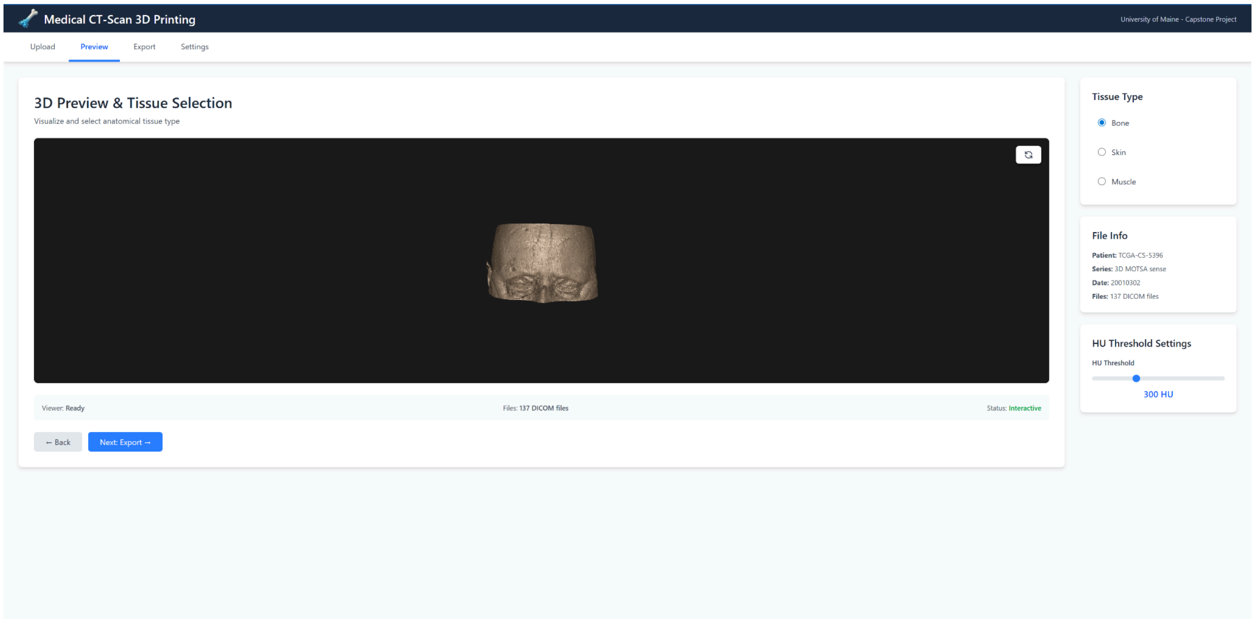
Standardized Navigation & Features:

Browser Back Button: The primary method to return to a previous screen or the application homepage is assumed to be the standard web browser's "Back" button, as no dedicated on-screen "Back" or "Cancel" button is present.

Proceeding to the Next Screen: Navigation to the next part of the application is automatic and triggered after a successful file selection and upload. the application will advance to “3D Preview” upon processing the valid DICOM data.

3.3 3D Preview

Figure 3.3 3D Viewer



After the user has uploaded a DICOM they can view it in the viewer, moving the figure and scaling it to their needs

Control Panels (Left Sidebar):

Tissue Type Selection: A list of anatomical presets (Bone, Skin, Muscle) allowing users to select which tissue to segment and visualize in the 3D preview.

File Information Panel: Displays crucial metadata from the loaded DICOM file:
- **File:** The source file name
- **Slices:** The number of axial slices in the volume
- **Dimensions:** The voxel grid size
- **Spacing:** The physical size of each voxel, indicating high resolution

HU Threshold Settings: Can control HU Threshold of the 3d model through this slider

Status Bar: Located below the control panels, it provides real-time feedback:
- **Performance:** 30 FPS (Indicates smooth, real-time rendering)
- **Polygons:** 125,430 (Shows the complexity of the currently displayed 3D mesh)
- **Status:** Ready (Confirms the system is idle and awaiting user input)

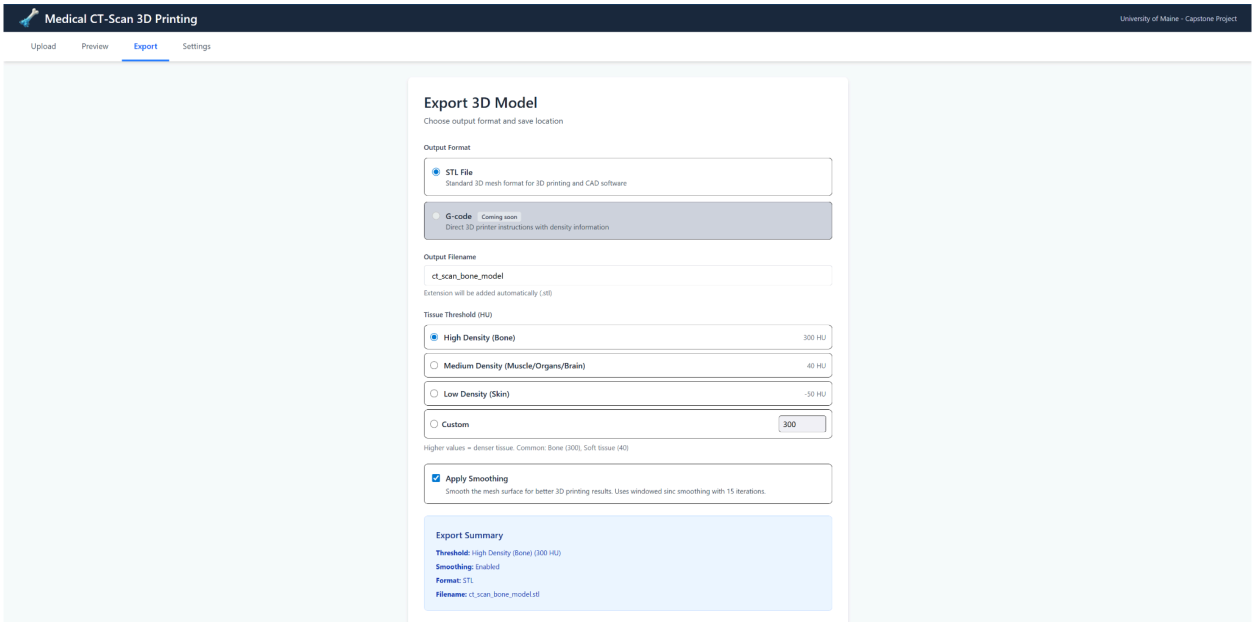
Main Visualization Area:
- **Label:** "3D Model Preview"
- **Primary Function:** An interactive WebGL viewport for the generated 3D model.
- **Interaction Hints:** Labeled "[Rotate - Zoom - Pan]" instructs the user on how to manipulate the view using standard mouse/touch controls.

Navigation:

Proceeding Forward: A "Next: Export →" button or link is present, typically in the bottom-right corner, to advance to the export screen once the user is satisfied with the tissue selection and preview.
Returning Back: As a standardized feature, the user can likely use the browser's "Back" button to return to the previous screen (e.g., the DICOM upload page).

3.4 Export

Figure 3.4 3D Modeler Exporter



3D Modeler Exporter allows the user to customize their 3D model options and make it readily available to ship

Output Format Selection:

STL File Option:

Description: "Standard 3D mesh format for 3D printing and CAD software"
Key Features: Polygonal surface representation, compatibility with slicing software, preserves geometry.
Use Case: Standard 3D printing workflow where the user will use external slicing software.

G-code Option:

Description: "Direct 3D printer instructions with density information"
Key Features: Mimics X-ray attenuation, reproduces tissue density, ready for printing.
Use Case: Advanced users who want to print density-mapped models directly.

File Settings Panel:

Output File Name: An editable text field pre-populated with "ct_scan_bone_model" where users can customize the output file name. The appropriate file extension (.stl or .gcode) will be added automatically.

Save Location: Shows the current destination path "/Downloads" with a "Browse..." button to open a system dialog for selecting a different save folder. Help text "Select where to save the output file" provides additional guidance.

Export Summary Panel: Provides a final confirmation of the export parameters:

Tissue Type: Shows which segmentation preset is being exported)
Format: Confirms the selected output format

Estimated Size: Manages user expectations for file size
Processing Time: Informs the user about potential wait time

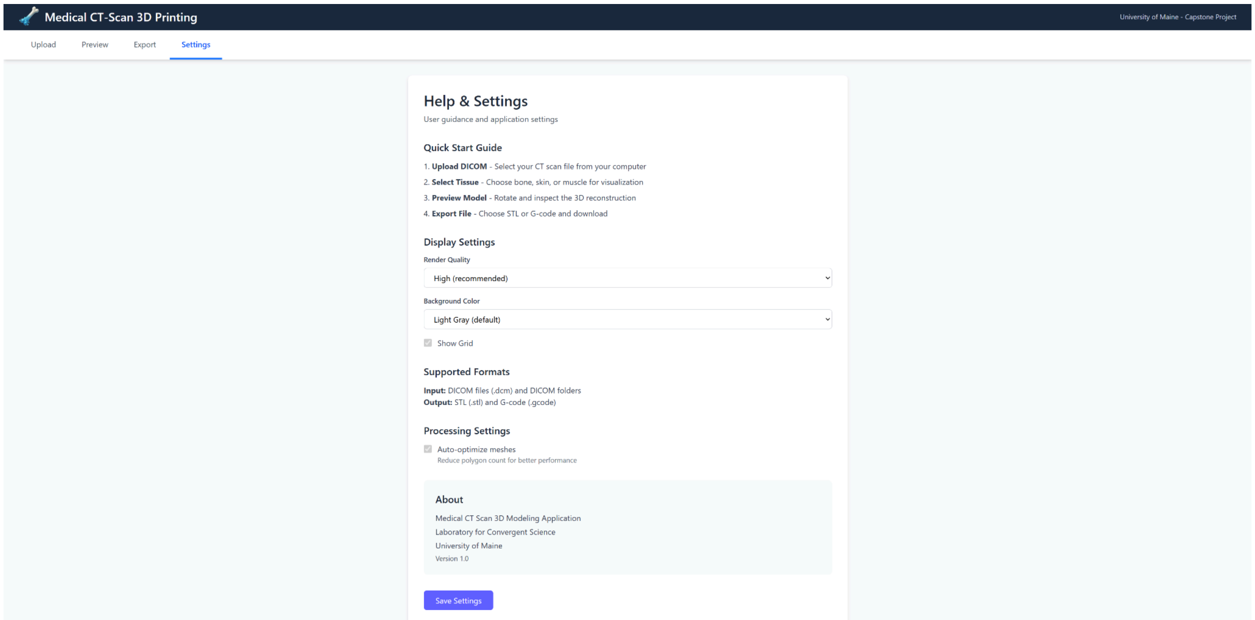
Action Buttons:

Back: Returns the user to the previous screen (3D Preview & Tissue Selection) to make changes.
Cancel: Aborts the export process entirely, likely returning to the main application hub.
Generate & Download: The primary action button that initiates the file generation process and triggers the download to the specified location once processing is complete.

Standardized Navigation: The browser's back button would provide an alternative method to navigate back to the previous screen, consistent with the application's navigation pattern.

3.5 Settings:

Figure 3.5 Help Page and Settings



The user can navigate through the help page if there are any issues with the navigation or understanding and the user can adjust their settings to customize the experience

Quick Start Guide Section: Provides a simple, 4-step workflow overview for new users:

Upload DICOM: Select CT scan files from the computer.
Select Tissue: Choose bone, skin, or muscle for segmentation.
Preview Model: Rotate and inspect the 3D reconstruction.
Export File: Choose STL or G-code format and download.

Display Settings Panel:

Render Quality: A dropdown (set to "High (recommended)") to balance visual fidelity against performance.
Background Color: A dropdown (set to "Light Gray (default)") to change the canvas color around the 3D model for better contrast.
Show Grid: A toggleable option to display a reference grid in the 3D preview.

Supported Formats Panel:

Input: Clearly lists accepted file types: "DICOM files (.dcm) and DICOM folders".
Output: Lists generated file formats: "STL (.stl) and G-code (.gcode)".

Processing Settings Panel:

Auto-optimize meshes: A checkbox option to "Reduce polygon count for better performance" during processing.
Default Export Format: A setting (shown as "STL") to pre-select the user's preferred output format.

Browser Compatibility Panel: Lists supported web browsers, with "Chrome (recommended)" highlighted, followed by Firefox and Safari.

Privacy & Data Panel:

> **Save recent files (cookies):** A toggleable setting for whether the app should remember recent files.
> **Clear History Button:** An action button that immediately wipes all locally saved browser data and cookies related to the application.

About Panel: Displays application information:

> **Application Name:** "Medical CT Scan 3D Modeling Application"
> **Developing Institution:** "Laboratory for Convergent Science, University of Maine"
> **Version Number:** "Version 1.0.0"

Navigation & Action Buttons:

> **Back to Home:** A navigation button to return to the main application interface or previous screen.
>
> **Save Settings:** The primary action button that applies any changes made to the settings. Changes are likely not permanent until this button is clicked.

Standardized Navigation: The browser's back button would provide an alternative method to navigate back without saving changes.

4. Data Validation

This section defines validation rules for all user-entered or user-selected data in the application. For each item we specify the control, data type, formats, limits (including boundary cases), defaults, and the precise validation behavior and error text.

ID	Screen / Control	Data item	Type	Allowed values / format	Limits & boundary cases	Required?	Default	Validation & error message
DV-01	Upload DICOM -> “Choose file(s)”	CT study files	File list	.dcm files or a DICOM folder (case-insensitive extension)	Min 1 file; Max total size ≤ 1.0 GB; must contain a single consistent series (orientation, spacing, dimensions); reject corrupt/non-DICOM; multi-series -> prompt to pick one; empty selection not allowed	Yes	—	On select: validate header + series consistency. Errors: “No DICOM files selected.” / “File is not valid DICOM.” / “Mixed or inconsistent series—select one series to continue.” / “Dataset exceeds 1.0 GB; load a smaller study.”
DV-02	Segmentation panel -> “Tissue”	Anatomical region	Enum	Bone, Skin, Muscle	Only these presets; switching re-applies thresholds	Yes	Bone	On change: re-segment + refresh preview. Error: “Unsupported tissue type.”
DV-03	Segmentation panel -> “HU Threshold”	Lower HU threshold	Integer	Whole number HU	Range −1024 … 3071; step 1; out-of-range clamps to nearest bound	No (overrides preset)	Preset per tissue	On blur: clamp; Error (only if typed outside range): “Enter a value between −1024 and 3071 HU.”
DV-04	Preview -> “Render mode”	Render mode	Enum	Surface, Volume	—	Yes	Surface	Invalid selection error: “Choose Surface or Volume.”
DV-05	STL Export -> “Smoothing iterations”	Mesh smoothing iterations	Integer	Digits only	0 … 10; step 1	No	2	Error: “Smoothing iterations must be 0–10.” (If post-process fails, export raw mesh and warn.)
DV-06	Export -> “Output type”	Export format	Enum	STL, G-code	Must match file extension rule in DV-08	Yes	STL	Error: “Select STL or G-code to continue.”
DV-07	Export -> “File name”	Output filename	String	1–128 chars; letters, numbers, spaces, dash/underscore; no path	Must not contain \/:*?"<>|; auto-append .stl or .gcode to match DV-06; not blank; deduplicate existing names by suffixing (1) etc.	Yes	Suggested from study UID (sanitized)	Errors: “Enter a file name.” / “File name contains invalid characters.” / “Extension doesn’t match selected format—using .stl/.gcode.” (App writes only to user-chosen location; no PHI embedded in exports.)
DV-8	G-code Settings -> “Printer profile”	Printer configuration	Enum	Generic FDM plus any user-saved profiles	If unavailable, fall back to default profile	No	Generic FDM	Warning (not blocking): “Printer profile unavailable—using defaults.”

Appendix A – Agreement Between Customer and Contractor

This User Interface Design Document constitutes a formal agreement between the Customer, Terry Yoo, and the Contractor, The Slice Is Right, regarding the architectural and detailed design for the Enabling 3D Printing of a Medical CT-Scan system. By signing below, both parties acknowledge that this document accurately translates the requirements specified in the Software Requirements Specification into a technical blueprint for system construction. The Customer agrees that this design provides a sufficient and appropriate basis for the system's implementation, and the Contractor agrees to develop a system that faithfully adheres to the design described herein.

Any future changes, additions, or modifications to the design specified in this document must be managed through a formal change control process. A request for change must be submitted in writing by either party and will be evaluated for its impact on the system's architecture, project scope, schedule, and feasibility. An amended version of this UIDD, or a formal change order referencing this document, must be mutually agreed upon and signed by authorized representatives of both the Customer and the Contractor before any design changes are implemented in the project.

Name	**Signature**	**Date**	**Comments**
Cooper Stepankiw			
Bryan Sturdivant			
Israk Arafat			
Greg Michaud			
Ethan Wyman			
Terry Yoo			

Appendix B – Team Review Sign-off

This document confirms that all undersigned members of The Slice Is Right project team have thoroughly reviewed the entirety of this User Interface Design Document for the Enabling 3D Printing of a Medical CT-Scan system. By signing below, each team member acknowledges their understanding of the proposed system architecture and detailed design and formally agrees that the content, scope, and technical direction outlined in this document are accurate, feasible, and provide a suitable foundation for the implementation phase of the project.

The comment section provided for each team member is to be used for noting any minor suggestions, editorial feedback, or non-substantive points of clarification. It is recognized that for this sign-off to be granted, there are no major points of contention regarding the architectural or detailed design decisions outlined within this UIDD.

Name	Signature	Date	Comments
Cooper Stepankiw			
Bryan Sturdivant			
Israk Arafat			
Greg Michaud			
Ethan Wyman			
Terry Yoo			

Appendix C – Document Contributions

Member Name	Primary Responsibilities	Estimated Percentage
Israk Arafat	Section 4	20%
Gregory Michaud	Introductions, 3.1	20%
Cooper Stepankiw	Appendix A,B,C, Section 3	15%
Bryan Sturdivant	Section 2, Section 3 mockups	25%
Ethan Wyman	Section 3	20%
Total		100%