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

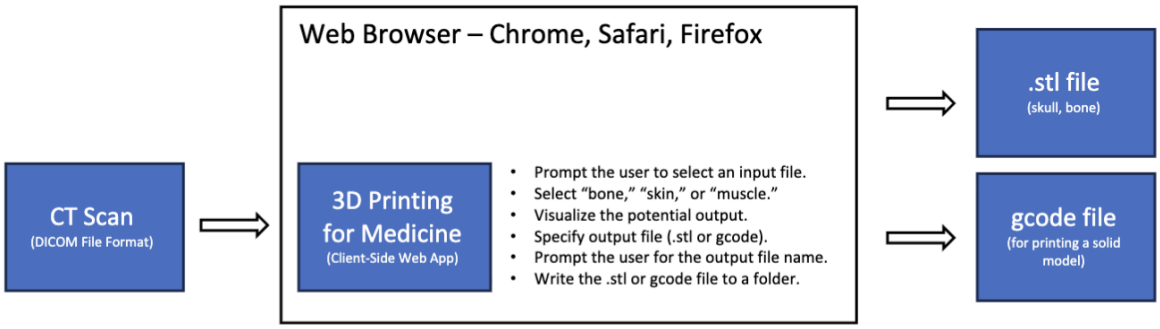
Dr. Terry Yoo, who serves as the Director of the Laboratory for Convergent Science at the University of Maine, has dedicated his research to the fields of medical imaging and 3D printing. His primary focus is on taking complex medical data and turning it into physical models that can be used for both 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

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%