# AutoVox

## *Requirements Standard Plan*

Omar Abdelhamid, Daniel Maurer, Andrew Bovbel, Olivia Reich, Khalid Farag

Version 1, 2025-10-10

# Table of Contents

# Academic Integrity Disclaimer

We would like to acknowledge that as a dedicated students of McMaster University, we have thoroughly read and comprehended the Academic Integrity Policy published by the university. We are committed to upholding the principles of academic honesty and integrity in all aspects of our educational journey. We understand the importance of acknowledging the work and ideas of others, and we pledge to ensure that all our academic endeavors are conducted with the utmost originality and compliance with the university's policy.

**We affirm that the content presented in this document is entirely our own, and any external sources used have been appropriately cited and referenced.**

## Omar Abdelhamid

As I submit my work, I, **Omar Abdelhamid**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

## Daniel Maurer

As I submit my work, I, **Daniel Maurer**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

## Andrew Bovbel

As I submit my work, I, **Andrew Bovbel**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

## Olivia Reich

As I submit my work, I, **Olivia Reich**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

## Khalid Farag

As I submit my work, I, **Khalid Farag**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

# Control Information

| **Version** | **Delivery** | | **Feedback** | |
|---|---|---|---|---|
| | *Deadline* | *Delivered* | *Received* | *Integrated* |
| **V1** | October 10, 2025 | October 10, 2025 | | |
| **V2** | | | | |
| **V3** | | | | |

# (G) Goals

## (G.1) Context and Overall Objectives

At McMaster University, the Computing and Software Department working on robotics and soft robotics has faced challenges in efficiently designing and modifying voxel-based 3D-printed components. While 3D printers and traditional CAD software are used, existing tools are time-consuming and inflexible, making voxel-level edits difficult and lengthy. By developing a tailored voxel design tool, Team Five of a Kind aims to provide a cost-effective and user-friendly solution that enables researchers to efficiently edit and voxel properties layer by layer. This tool will streamline the design process, enhance flexibility, and directly support the department's research in multi-material 3D printing.

## (G.2) Current situation

Current workflows for voxel-based 3D printing rely heavily on traditional CAD tools or multiphysics simulators such as COMSOL [3]. While these platforms are powerful, they are not tailored for voxel-level manipulation and are often prohibitively expensive. Designing geometries voxel by voxel can take tens of hours or even days, and once material properties are assigned, they cannot easily be reverted or adjusted. Researchers and students have described frustration with the inflexibility of these tools, as small changes frequently require rebuilding large portions of a model from scratch. This has left many experiments inefficient, time-consuming, and limited in their ability to explore complex material-property distributions.

## (G.3) Expected Benefits

This section summarizes the primary goals of AutoVox, and what is expected it will accomplish. This is described as a whole in Figure 1 below, showing the relationships between primary stakeholders and how each of their goals both interact and contribute to the overall benefit to society.

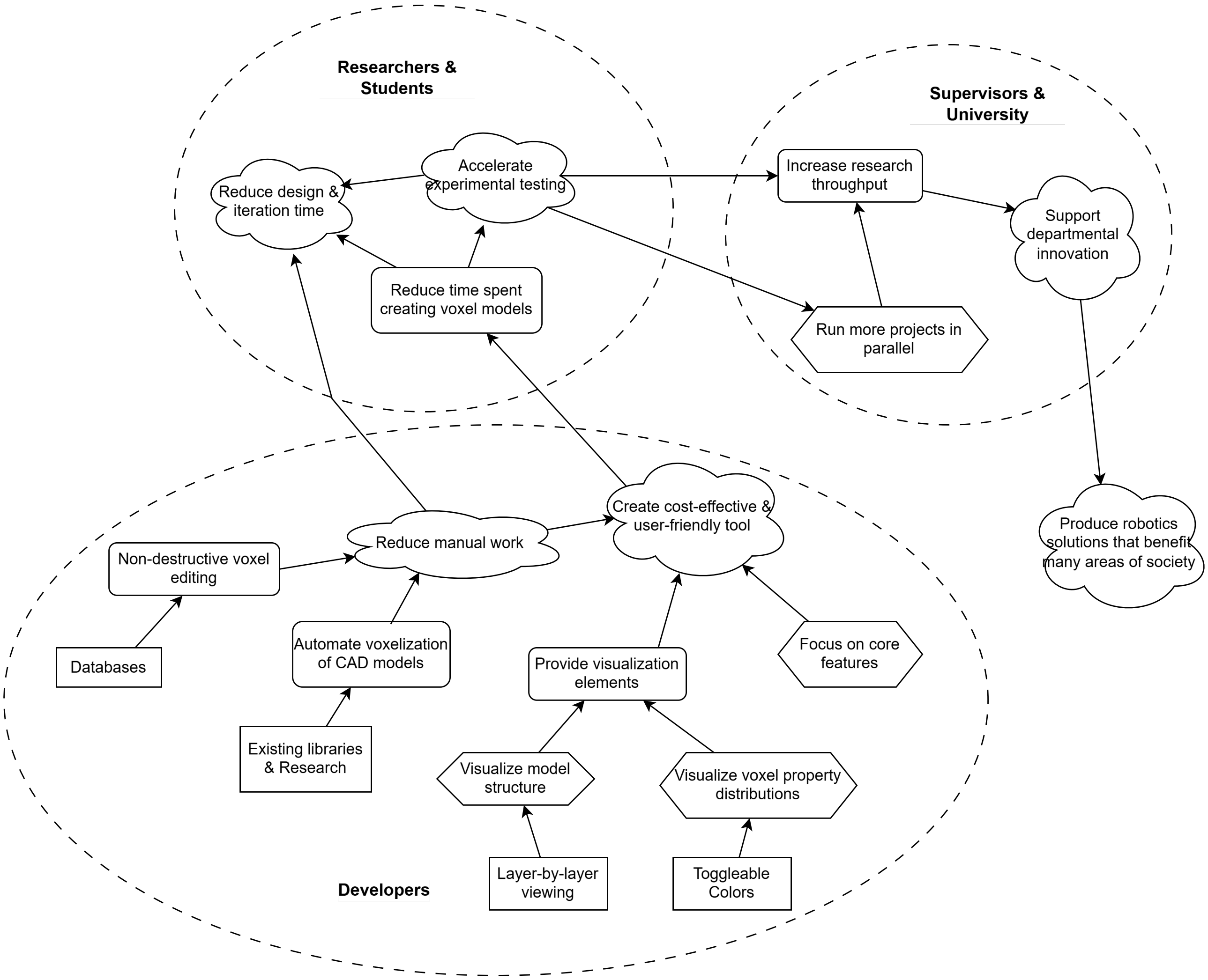


*Figure 1. Goal Model Diagram. Notation is as follows; clouds represent soft goals, rounded boxes represent hard/measurable goals, hexagons represent plans, and unrounded boxes represent resources.*

The primary benefit of this project is to provide a cost-effective and user-friendly voxel-based design tool that converts CAD models into voxel grids, enabling flexible editing and material property assignment. To accomplish this, supporting goals such as reducing manual work, automating voxelization, providing visualization elements, and non-destructive editing are considered. This solution reduces time spent creating voxel models and supports the department's research in multi-material 3D printing, aligning with the benefits expected by the following primary stakeholders and contributing to their goals.

For researchers and students, the tool will streamline workflows by allowing faster iteration and enabling edits to voxel properties without rebuilding models from scratch. Experimental testing is also accelerated and as a result, more projects can be run in parallel. Research throughput is then increased - a goal of supervisors and the university. These goals ultimately support department innovation, contributing to the overall benefit to society. With research running more quickly, innovative robotics solutions that benefit many areas of society can be developed and iterated on at a faster pace.

# (G.4) Functionality overview

The high-level functional requirements for the voxel-based design platform based on client input can be defined as follows, in no particular order:

- **Support automatic voxel model creation. (G4-HF-1)** The most time-consuming aspect of creating voxel models using current tools is creating them manually; a solution to this involves importing CAD files and converting them into a representative collection of voxels to support further editing.
- **Offer voxel-based editing. (G4-HF-2)** A key functionality that is currently tedious is voxel property assignment; while edits to individual voxels is necessary, a solution such as batch-editing would greatly improve the user experience. To address this, the system should offer an interface allowing the assignment of material and magnetization properties both at the voxel-level and to groups of voxels.
- **Provide visualization tools. (G4-HF-3)** The ability to quickly examine the model structure and property distribution from various angles to verify model correctness is essential. Keeping in mind that visual examination is often most efficient, the system should provide visualization of material distribution and magnetization vectors across the voxel grid, assisting researchers in verification before fabrication.
- **Ensure export compatibility. (G4-HF-5)** Current solutions require work-around-eqsue conversion from the files exported to what the custom 3D printer requires. In order to be both compatible with current printing solutions and relevant in the future, the system should export a file that contains the necessary voxel-level data in an widely-recongized, standard format, such as CSV.

Of the functional requirements listed above, the two most critical are **automatic voxel model creation** and **voxel-based editing**, as these form the foundation of the system's functional uses and research value.

Some high-level non-functional requirements for the system can be defined as follows, in no particular order:

- **Interface usability. (G4-HNF-1)** The system should have an interface that is suitable for both researchers and students, minimizing training overhead. Functions involving edits, changes to model perspective, and model import/export should be accomplished in as few interactions as possible, by a high percentage of users.
- **System performance. (G4-HNF-2)** The system should support model conversion at a minimum rate, and changes to visual perspectives or edits with a maximum delay, even for projects containing a maximum number of voxels. These rates, delays, and maximums will be set based on user needs.
- **File compliance. (G4-HNF-3)** The systme should support a standard CAD file format and integrate with the existing custom 3D printing software.

# (G.5) High-level usage scenarios

This section outlines the fundamental usage paths and primary interactions users will have with the system. These high-level use cases present the main scenarios that the system is designed to address. For more detailed, special scenarios, refer to S.4.

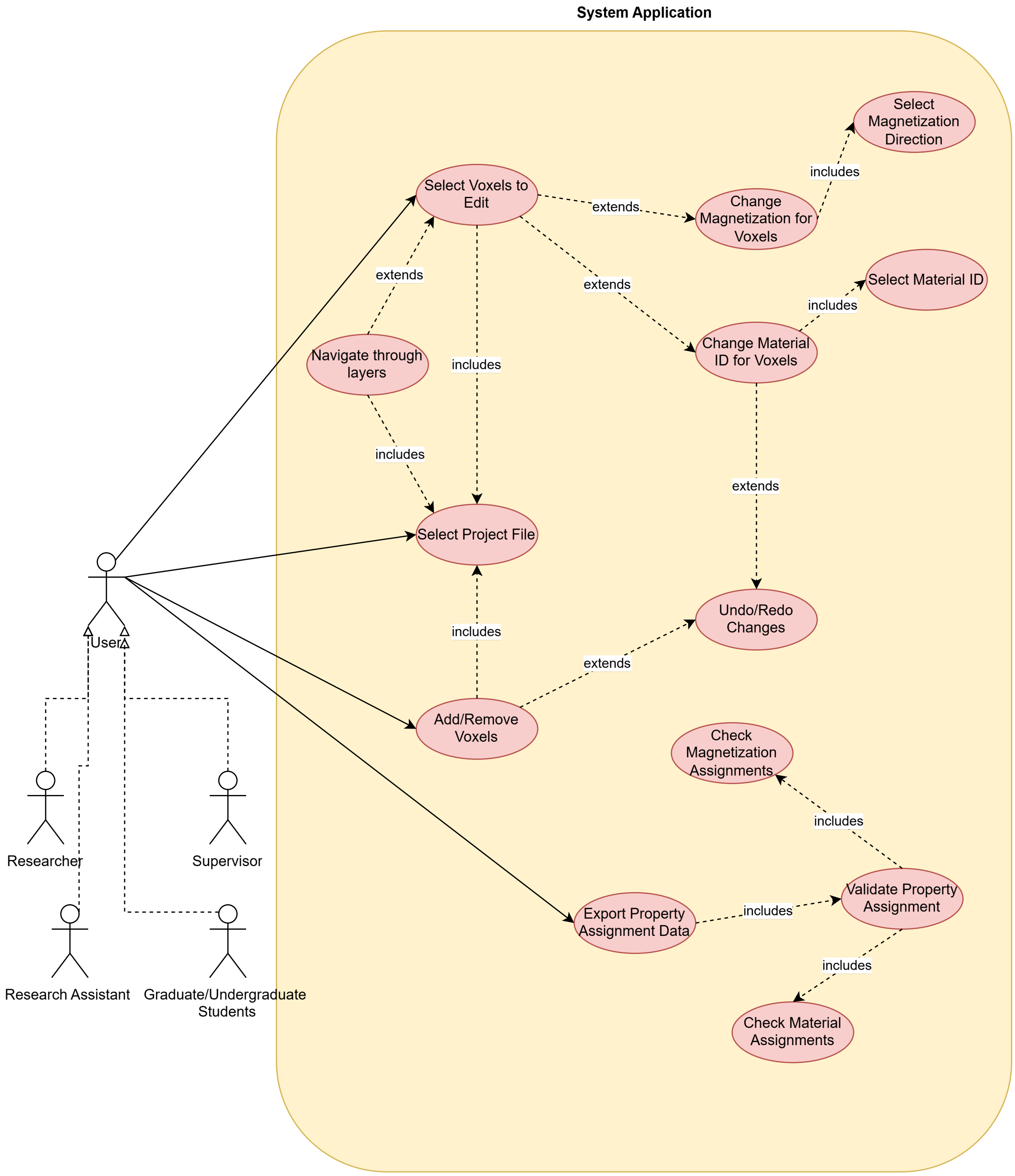


*Figure 2. High-level Use Case Diagram*

Figure 2 above highlights how we anticipate our direct stakeholders will interact with the voxel property assignment system for the principal use cases. More details for these usages are as follows:

- **UC1: Import CAD File.**
  1. User obtains a finished CAD file from CAD software (e.g., AutoCAD).
  2. User opens the system and selects the option to import the CAD file.
  3. System converts the file to a set of voxels and creates a new project file.
  4. System displays the voxelized model ready for property assignment.
- **UC2: Browse and Select Voxels.**

1. User opens the system and selects an existing project file.
2. System loads and displays the selected file.
3. User navigates layers to view the voxel structure and selects a set of voxels.
4. System highlights selected voxels for property assignment.
5. User can modify selections across different layers.

- **UC3: Assign Properties to Voxels.**
  1. User opens the system and selects an existing project file, and selects voxels.
  2. User specifies a magnetization direction.
  3. System provides visual feedback indicating the voxels have been magnetized.
  4. User specifies a material ID.
  5. System provides visual feedback indicating the voxels have been assigned a material.
- **UC4: Add and Delete Voxels.**
  1. User opens the system and selects an existing project file.
  2. User selects a grid space where they wish to create a new voxel.
  3. System adds a new blank voxel to the model at the selected space.
  4. User selects a voxel they wish to delete.
  5. System removes the voxel from the model.
- **UC5: Export Property assignment Data.**
  1. User completes property assignment for the entire model and initiates export process.
  2. System validates that all voxels have both a material and a magnetization vector assigned.
  3. System generates a file containing per-voxel data, including voxel location, magnetization vector, and material ID.
  4. User receives the exported file for use with a multi-material 3D printer.

# (G.6) Limitations and Exclusions

Below is a list of limitations that the system will not do:

- System will not act as a replacement for full multiphysics simulation software such as COMSOL. While it provides voxel-level editing and visualization, it will not perform advanced physics simulations (e.g., fluid dynamics, stress-strain analysis, or electromagnetic field simulations).
- System will not provide automatic optimization of material distributions. Users are responsible for deciding how to assign materials and magnetization properties; the tool only facilitates editing and visualization.
- System will not guarantee the accuracy of exported designs beyond voxel-level conversion. Verification of compatibility with all 3D printers and fabrication processes is outside the project scope. (May be expanded on later)
- System will not provide real-time error correction or printer control. The platform’s role concludes once a voxelized model is exported; any fabrication issues must be handled by the 3D printer’s native software.

# (G.7) Stakeholders and requirements sources

This section identifies the groups of people who can affect or be affected by the voxel-based design platform, along with the main documents and sources that informed its requirements.

**Direct stakeholders**

- **Researchers and Graduate Students** Researchers are the primary users of the system. They design voxel-based geometries, assign material and magnetization properties, and export models for 3D printing.

*Alex (Graduate Researcher)* – Alex is a Master's student working on magnetically responsive soft robots. They spend long hours editing COMSOL files voxel by voxel and often need to redo entire models after small mistakes. They want a faster, more flexible way to edit and re-magnetize voxels.

- **Research Assistants and Undergraduate Users** Research assistants and undergraduate students support active projects by preparing CAD models and testing the voxel design workflow. They rely on the tool to visualize and verify structures before printing.

*Jordan (Research Assistant)* – Jordan assists in multiple robotics experiments and often handles file preparation for printing. They need an intuitive interface that allows them to quickly confirm magnetization directions and export models accurately.

- **Supervisors and Research Leads** Supervisors oversee the progress of research projects and ensure that new tools contribute to the lab's long-term goals of faster prototyping and higher-quality results.

*The Supervisor* – The research supervisor manages several students and prototypes simultaneously. They value a system that reduces design time, prevents repetitive work, and scales efficiently for larger CAD models.

**Indirect stakeholders**

- **University IT and Technical Support** – Responsible for installing and maintaining the software on lab systems, ensuring compatibility with institutional infrastructure, and troubleshooting technical issues as needed.
- **3D Printer Technicians and Operators** – Use the exported voxel models during fabrication. While they do not interact with the system directly, their feedback informs file format standards and ensures successful printing.
- **Future Developers and Maintainers** – Extend and update the voxel design platform to support new research needs. They rely on clear documentation, modular architecture, and consistent version-control practices.
- **University Administration and Research Management** – Ensure that the software complies with institutional policies regarding data management, security, and accessibility. They indirectly influence system requirements through governance and approval processes.

**Requirements sources**

- Official project brief and supervision notes from McMaster University's Engineering Research Department.
- Observations and feedback from researchers and assistants using CAD and COMSOL in robotics and soft robotics labs.

- Technical documentation for CAD file formats (e.g., STL, STEP) and 3D-printer integration standards.
- Institutional policies on software deployment, accessibility, and research data management.
- Informal discussions and requirement clarification meetings with lab members and the supervising research team.

# (E) Environment

## (E.1) Glossary

This section details all technical terms used throughout this project's documentation, as well as project-specific vocabulary (eg. words used in different contexts here) and any non-obvious acronyms.

**3D**

Three-dimensional; in the context of this project, 3D implies the image produced evokes depth. In the case of voxels or vectors, 3D implies three spatial dimensions.

**3D Mesh**

Alternative way of referring to the 3D model contained within a CAD file; these files often store these models as collections of vertices, colloquially referred to as a mesh.

**3D Model**

A collection of vertices in 3D space, connected via edges and faces. A given CAD file contains these models in some fashion.

**CAD File**

A file containing a 3D model, generated by CAD (Computer Aided Design) software. In the context of this project, these are files the system can take as input in order to output a project file, containing a voxel model equivalent of the original 3D model.

**CAD Software**

Software used to create and modify 3D models and designs, stored as CAD files.

**Custom 3D Printer Software**

The software used by Researchers/Lab Operators that controls the custom 3D printer. This software takes the files the system exports (printer-ready files) as input.

**Display Section**

A portion of the voxel model that is displayed at a time. For large models, the model is broken down into parts; only one is displayed at a time. This is done to save computer resources.

**Export File / Printer-ready File / Metadata File**

A file generated from a project file, containing all data related to each voxel (coordinates, material assignment, magnetization).

**Grid**

The workspace a voxel model exists within (eg. a coordinate system). Voxels align to a cartesian coordinate system, voxels can be created within it.

**Magnetization vector**

A 3D polar coordinate the user enters to specify the magnitude and direction of the magnetic field within an individual voxel. Also referred to as a magnetization. See more about magnetic field theory here [4].

**Material ID**

A non-negative whole number that represents the material of an individual voxel. Users can link a color and name to these IDs. Also referred to as a material assignment.

**Model Slicing / Voxelization**

The process by which an input CAD file is converted to a voxel model / project file. The name results from the analogy of "cutting" a 3D model into cubes.

**Project File**

A file that contains the data related to a voxel model, as well as other saved data the system stores with a project (eg. display sections).

**Voxel**

Traditionally, a voxel is the equivalent of a 3D pixel - values aligning to a regular 3D grid that takes on the appearance of a cube. Within our project, it is possible for the height of these voxels to change.

**Voxel model**

A collection of voxels that the user can manipulate, assigning each voxel properties or adding/removing voxels.

# (E.2) Components

This section details any components external to the system that either affect or are affected by the system. As our project is made to be stand-alone and does not require the interfaces of any external hardware or applications, the components listed here are software that will be used in conjunction with the resulting system, and thus indirectly interact with it.

There are two primary components that the system interacts with:

- **External CAD Software.** This component is any external software responsible for generating files users will import and use the system to convert. In this way, the system indirectly interacts with this software; if this software changes how it exports files, the system would be affected.
- **Custom 3D Printer Software.** This component is the pre-existing software that operates on the files the system will export. Similar to the above, should this software change the format of the file needed as input, the system would be affected.

# (E.3) Constraints

This section details constraints imposed on the project. To summarize, key constraints include specifically which files and operating systems the system must support.

- **Operating System Support (E3.1).** The system must support the Windows and Linux operating systems. This constraint is necessary to ensure the application can be installed and run on the lab computers utilised by the application's users.
- **STL File Support (E3.2).** To support the importing and converting of user-created external CAD files, the system must parse STL files. Information relating to the specifications of STL files can be found here [5].
- **Export File Support (E3.3).** In order to export user-created voxel project files in a format usable by pre-existing 3D printing software, the system must support the creation of export files containing all necessary metadata, organized in such a way that minimal changes need to be made to the existing software.
- **Minimize resource usage (E3.4).** The system must be implemented in such a way to minimise the usage of computer resources as much as is feasible. Some level of usage is unavoidable; a main feature of the system is the rendering of a 3D graphical user interface. However, it is important to balance needed functionality and device resource usage. Desktop resources are limited, and may be utilised for other tasks simultaneously.

# (E.4) Assumptions

This section details any assumptions made that are not imposed upon the project. As follows are assumptions made to simplify project scope.

- **Operating System Support (E4.1):** We assume that only recent Windows releases and Linux distributions that are made use of by primary users will need to be supported. For the first development iteration, this is sufficient to satisfy all potential users.
- **Processing Power (E4.2):** We assume that the lab computers possess at least the resources required to handle rudimentary 3D graphics.
- **Browser Support (E4.3):** We assume that the system will only support Chromium browsers. This will capture a majority of the most popular browsers in use.
- **Language Support (E4.4):** We assume that the only language the system supports will be English. This captures the needs of the primary users.

# (E.5) Effects

This section details the most important effects of the system; we focus on the benefits usage of the system provides to stakeholders.

- **Increased productivity of users.** A key pain point expressed by primary users is the extensive time spent tediously creating models voxel-by-voxel from paper sketches. With our system in place automating conversion from CAD files, users will be able to spend this time on other projects, alleviating hours of menial work from their schedule. In addition, the dedicated, lightweight property

assignment interface will speed up time to assign material IDs and magnetic vectors to voxel models versus the general-purpose physics simulation software currently employed by primary users.
- **Increased lab output.** With primary users more productive, model ideas can be actualised and tested more quickly. Users will no longer need to budget time to convert models to voxels, and can instead develop new ideas. With multiple instances of the system, many models can be worked on in parallel, further increasing output. Finally, with the aid of the system, the production of more complicated models becomes more feasible, freed from time constraints.

# (E.6) Invariants

This section details assumptions related to system operations that we expect to hold throughout the system's usage. Included are both assumptions related to file usage and user characteristics.

- The system assumes that imported models are printable (with reference to the custom 3D printer); it is assumed that users are knowledgeable to be able to recognize possible errors that may occur. Checking model geometry is considered out of project scope.
- The system assumes that voxel height will not change after converting a CAD file; the system will not provide methods to alter the height of voxels after initial project creation.
- The system assumes that users are aware of any associated physics with respect to model simulation; simulation of models is considered out of project scope.
- The system assumes that users will not attempt to edit voxel models with software other than the system itself.

# (S) System

## (S.1) Components

- **Import Manager**: This component has the responsibility of interpreting and manipulating a CAD mesh contained within a compatible file. It centralizes the business logic for model initialization through defined display segments as well as voxel slicing, which further partitions the display segments into voxels. Upon completing the slicing process, it interacts with the Visualization Manager by indicating all provided information is present and prompting it to begin the process of recreating the voxel model.
- **Visualization Manager:** This component is in charge of handling all tasks associated with recreating a 3D rendition of the voxel model in our software and subsequent user interaction during individual voxel access. It interacts with the Import Manager to gather information regarding image specification. It exposes a user interface that permits responsive 3D image interaction, allowing users to manipulate and examine it from all perspectives. It also centralizes the business logic related to intuitive visualization that enables extensive user understanding of the current model state.
- **Editing Manager:** This component takes on the responsibility of facilitating and capturing all data from user modifications on a voxel level. This includes the assignment of desired properties on an individual or multi-voxel scale. It centralizes the business logic of voxel alterations and metadata annotations. It exposes a user interface that allows easy navigation and modification of each voxel's metadata.
- **Exportation Manager:** This component acts as an integration gateway that facilitates seamless integration with pre-existing software developed for printing magnetized, multi-material models. It handles the business logic that dictates file exportation with complete metadata annotations, proper printing compatibility and capability for future regeneration within the software.

This system will not directly interact with any external components. However, as outlined by the Exportation Manager, our system must output a file that is compatible with external printing software to support the natural workflow of the printing process.

The application of the overall system functionality will be handled by the AutoVox, which is a desktop application. Users will be required to directly install the software on their laptop, as the AutoVox will be a locally deployed software system.

Figure 3 illustrates how the components detailed above are connected and interact. Figure 4 provides a boundary diagram for the system, detailing high-level interactions between the user and system (use cases shown here are detailed in section S.4).

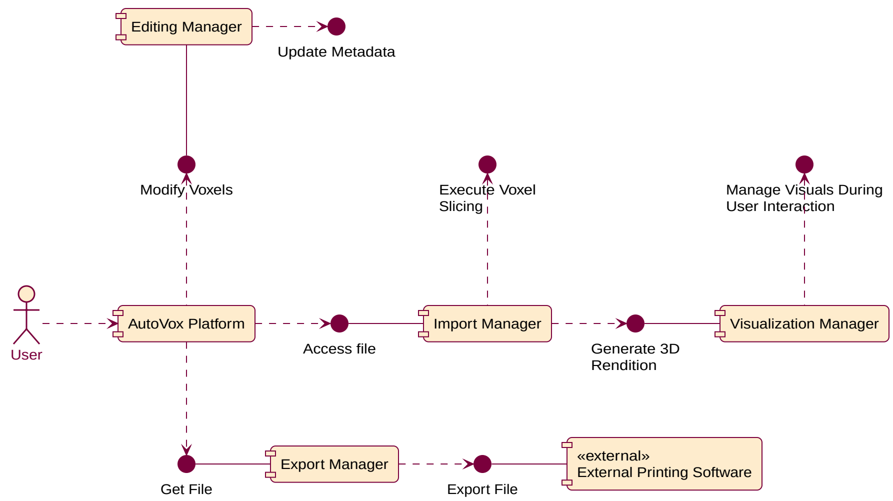


*Figure 3. Components involved in AutoVox platform; system interaction diagram.*

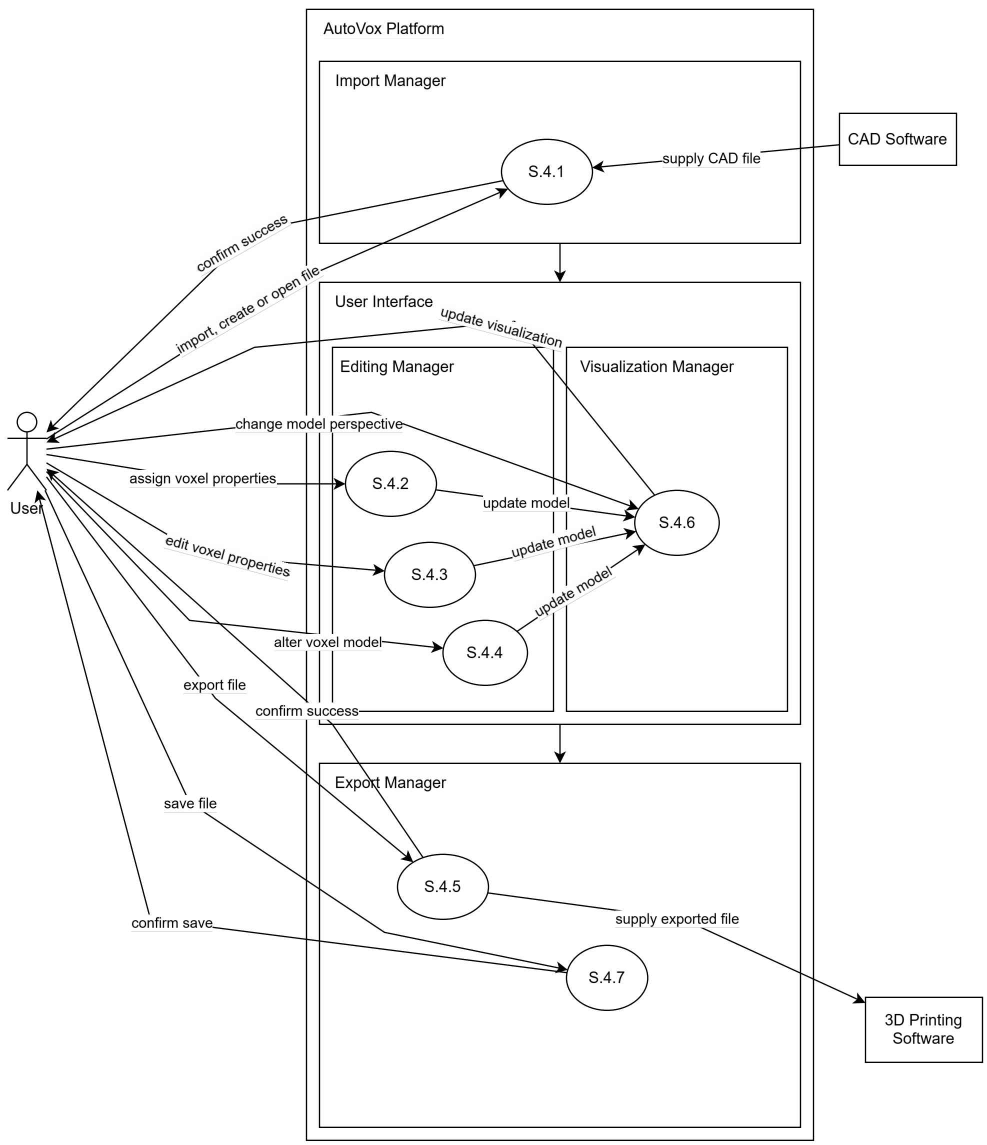


*Figure 4. AutoVox boundary diagram.*

# (S.2) Functionality & Prioritization

The following section outlines the specific functional and non-functional requirements, where each requirement has been defined within the scope of a specified component except for a few requirements that pertain to the entire system. Each requirement contains a section underneath that outlines the priority, likelihood of change, and justification to convey its relative importance and potential variability.

For certain requirements, variables have been defined. A description of each variable as well as its assigned value has been included below:

- **OPTIMAL_VOXEL_SIZE** = 5.5nm This was derived using the optimal voxel size specified by the stakeholder.
- **MAX_VOXELS** = 13,996,800,000 voxels This considers the maximum number of voxels contained within the entire model.
- **MAX_LAYER_DISPLAY** = 518,400 voxels This was derived assuming 960 voxels x 540 voxels per layer. It considers the maximum number of voxels contained within a single layer for a single display window
- **MAX_DISPLAY** = 103,680,000 voxels This was derived assuming 960 voxels x 540 voxels per layer. It considers the maximum number of voxels that will be displayed at a given time.
- **MIN_RATE** = 500,000 voxels per second This considers how fast voxel data will be generated during file import and export.
- **MAX_EDIT_LATENCY** = 1 second This considers how quickly the system should respond to model modifications performed by the user.
- **MAX_VIEW_LATENCY** = 500 ms This considers how quickly the system should respond to changes in model view orientation by the user.
- **MAX_INTERACTIONS** = 5 This considers the maximum number of steps (eg. clicks, enter details into a text box) a user should take to complete a part of a task.
- **USABILITY_PERCENTAGE** = 90% Relates to the percentage of users that should be able to succeed at different tasks within the system.

## (2.2.1) Import Manager

### Functional Requirements

1. **New project creation (F211):** Import Manager shall allow a user to create a new project file, with the option to import an existing CAD file from their personal device that will be sliced into voxels.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It's required for users to be able to start new projects without having to build a model voxel-by-voxel. It contains critical functionality, which means there's limited flexibility for change.

2. **Past project access (F212):** Import Manager shall allow a user to open an existing project file in order to reopen a project with all magnetization and material properties preserved.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low

*Justification:* It will improve user experience by avoiding full re-magnetization for minor adjustments to the model. There's minimal chance of change as it is the simplest method to streamline future edits to existing models.

3. **Voxel size (F213):** Import Manager shall allow the user to configure the voxel dimensions into which the imported CAD model will be sliced.

   *Priority:* Low (could-have)
   *Likelihood of change:* Medium
   *Justification:* This is not required by the user for current usage, but the added functionality may simplify more complicated processes in the future. Change may occur depending on how it impacts the core functionality of voxel slicing.

4. **Geometric model size (F214):** Import Manager shall allow users to modify the dimensions of their model in order to scale the model to the desired printing size before being sliced into voxels.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Medium
   *Justification:* It offers the flexibility to modify geometric model size to enhance overall user experience. Change may occur depending on how it impacts the core functionality of voxel slicing.

5. **Model adjustment (F215):** Import Manager shall resolve partial voxels and interpret them using a method that preserves the integrity and accuracy of the provided model.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It ensures that the model is preserved, minimizing user effort to achieve voxel-level accuracy. It contains critical functionality, which means there's limited flexibility for change.

6. **Model division (F216):** Import Manager shall partition the model into a grid of display sections that do not surpass the MAX_DISPLAY threshold and retain a latency between user input and display update of MAX_VIEW_LATENCY.

   *Priority:* Medium (should-have)
   *Likelihood of change:* High
   *Justification:* It can improve user experience by creating easy-to-handle partitions and preventing lag that may arise from high-demand visuals. The MAX_DISPLAY threshold is subject to alteration to ensure optimal performance, adhering to MAX_VIEW_LATENCY.

### Non-Functional Requirements

1. **Scalable projects (NF211):** Import Manager shall support the generation of project files that contain up to MAX_VOXELS without loss of functionality.

   *Priority:* Medium (should-have)
   *Likelihood of change:* High
   *Justification:* It will ensure the software can sufficiently handle large-scale projects. The MAX_VOXELS threshold is subject to change depending on software constraints and realistic current needs of a

typical user.

2. **Slicing process performance (NF212):** Import Manager shall generate the project file at a minimum rate of MIN_RATE.

   *Priority:* Low (could-have)
   *Likelihood of change:* High
   *Justification:* It reduces wait time to improve user experience but is not critical. Changes to the MIN_RATE threshold may be necessary in order to sufficiently support the execution of a high-demand operation.

3. **Ease of use (NF213):** Import Manager shall ensure projects can be created or opened in less than MAX_INTERACTIONS by USABILITY_PERCENTAGE of users.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low
   *Justification:* The process of initial project creation from either a CAD file or from scratch, or opening an existing project file should be as simple as possible to encourage system intuivity. This requirement is unlikely to change given this importance, but the definition of maximum interactions may need to be adjusted given possible complexity in project creation.

## (2.2.2) Visualization Manager

### Functional Requirements

1. **3D image rendition (F221):** Visualization Manager shall display voxel models using a method that ensures each outer voxel is visually separate from surrounding voxels.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It provides vital visualization of the model, which is a critical guide for property assignment. Users need to be able to tell where each voxel starts and ends in order to properly examine models. Due to its critical nature, there's limited flexibility for change.

2. **Partition division (F222):** Visualization Manager shall provide users with one-interaction navigation across display sections to ensure access to all model partitions.

   *Priority:* Medium (should-have)
   *Likelihood of change:* High
   *Justification:* Its priority is linked to the Model division requirement. Once the Model division requirement is implemented, its priority is elevated to ensure a smooth and positive user experience once the model is divided into partitions. Modifications will likely be tightly coupled with any changes in the Model division requirement.

3. **Image interaction (F223):** Visualization Manager shall provide users with an interface that permits continuous navigation across perspectives of the voxel model, following real-world 3D movement conventions.

*Priority:* High (must-have)
*Likelihood of change:* Medium
*Justification:* It provides users with a quality experience by ensuring complete and extensive visualization. Following real-world conventions such as left movements to turn an object left ensures the interface is intuitive. Variations may occur depending on the perspectives required to satisfy the user's visualization needs.

4. **Layer focus (F224):** Visualization Manager shall allow users to isolate and display a specific layer to facilitate property assignments, rendering all other voxels irrelevant while that layer is in focus.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It facilitates accessible interaction with voxels during property assignment. There's minimal chance of change because it is the simplest method to access voxels inside a 3D model.

5. **Highlight voxel selection (F225):** Visualization Manager shall visually indicate which voxels the user has currently selected within a specific layer.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It provides active feedback based on current user interaction to improve usability. It contains critical visualization functionality which means there's limited flexibility for change.

6. **Material assignment tracker (F226):** Visualization Manager shall integrate tracking of voxels that have been assigned material IDs by adjusting the colour of the voxel to indicate assignment completeness.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Medium
   *Justification:* It provides users with easy visualization of remaining material assignment work, which improves usability in non-critical areas. There's a possibility of change regarding how it is displayed to users.

7. **Magnetization assignment tracker (F227):** Visualization Manager shall integrate tracking of voxels that have been assigned magnetization vectors by providing the option to toggle the colour of all magnetized voxels.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Medium
   *Justification:* It provides users with easy visualization of remaining voxels that require magnetization, which improves usability in non-critical areas. Similarly to the Material tracker, there's a possibility of change regarding what constitutes complete magnetization and how it is displayed to users.

### Non-Functional Requirements

1. **Image updates (NF221):** Visualization Manager shall update any changes to the perspective of the 3D model with a latency of less than MAX_VIEW_LATENCY.

*Priority:* Medium (should-have)
*Likelihood of change:* High
*Justification:* It allows a seamless user experience with smooth and responsive interaction. The latency threshold is likely to change in relation to technical feasibility as long as there is no significant lag that may interfere with user interaction.

2. **Visual scalability (NF222):** Visualization Manager shall support the visual display of voxel models that contain up to MAX_VOXELS without exceeding MAX_VIEW_LATENCY or MAX_EDIT_LATENCY, or any loss of functionality.

   *Priority:* Medium (should-have)
   *Likelihood of change:* High
   *Justification:* It will ensure software can sufficiently handle large-scale visuals within a project. The MAX_DISPLAY threshold is subject to change to ensure optimal visuals where there is minimal risk of undesirable lag.

3. **Accessible colours (NF223):** Visualization Manager shall ensure all colours utilized for all interfaces (e.g. voxel highlight colours, text colours, background colours) have sufficient contrast in accordance with WCAG 2.0 guidelines and can be distinctively recognized under standard lighting conditions. See more about WCAG 2.0 here [6].

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low
   *Justification:* It contains critical standards for easy visualization which means there's limited flexibility for change. If improper colours are used, the system may be rendered unusable.

## (2.2.3) Editing Manager

### Functional Requirements

1. **Magnetization assignment (F231):** Editing Manager shall allow users to select an individual or group of voxels and set a specific magnetization vector for them.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It provides users with the ability to magnetize voxels, fulfilling one of the key stakeholder needs. Its critical nature means there's limited capacity for change.

2. **Favourite bar (F232):** Editing Manager shall allow users to define, maintain, and access in one interaction a list of 'favourite' magnetization vectors.

   *Priority:* Low (could-have)
   *Likelihood of change:* Low
   *Justification:* It improves user satisfaction by allowing quicker and reusable magnetizations, though not critical to core functionality. There's minimal chance of change because it is the simplest method to streamline the assignment process of frequently applied magnetization values.

3. **Material assignment (F233):** Editing Manager shall allow users to select an individual or group of

voxels and assign a specific material ID to them.

*Priority:* High (must-have)
*Likelihood of change:* Low
*Justification:* It provides users with the ability to assign specific materials to each voxel, fulfilling one of the key stakeholder needs. Its critical nature means there's limited capacity for change.

4. **Material labels (F234):** Editing Manager shall allow users to assign a label to each material ID.

   *Priority:* Low (could-have)
   *Likelihood of change:* Low
   *Justification:* It supports a smoother user experience by ensuring clarity in how material numbers correspond to specific materials within individual projects. There's minimal chance of change since labels are the most direct method to convey this information.

5. **Property replication (F235):** Editing Manager shall allow users to select a group of voxels and replicate the defined properties of those voxels to other layers.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low
   *Justification:* It streamlines repetitive assignment processes to improve workflow efficiency. The core functionality is not expected to experience significant change even if there are multiple ways to integrate the requirement.

6. **Auto save progress (F236):** Editing Manager shall periodically save any changes made to set property configurations without requiring manual action to preserve data.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low
   *Justification:* It strengthens dependability and user confidence within critical operations. There's minimal chance of change because it is the simplest method to ensure data preservation during user interaction.

7. **Edit history (F237):** Editing Manager shall allow users to access edit history and revert the model to a previous version.

   *Priority:* Medium (should-have)
   *Likelihood of change:* High
   *Justification:* It facilitates positive user interaction when inevitable mistakes are made during assignment and editing. There will likely be change regarding what constitutes a previous version (i.e., how far back within modification history a user can revert).

8. **Select layer (F238):** Editing Manager shall allow users to select an entire layer with one interaction to enable assignment of a specified material or magnetization amongst all voxels within a layer.

   *Priority:* Low (could-have)
   *Likelihood of change:* High
   *Justification:* It promotes efficiency and reduces the time required for wide-scale property assignment.

Change may occur if the software already sufficiently supports large-scale assignment and the feature becomes redundant.

9. **Manual voxel alteration (F239):** Editing Manager shall allow users to add and delete voxels of the same defined dimensions present in the rest of the model.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low
   *Justification:* It provides users with greater control over small model adjustments through quick and simple voxel edits. There's minimal chance of change because it is the simplest method to allow voxel-level edits.

10. **Reset Voxels (F2310):** Editing Manager shall allow users to reset all property assignments to their unassigned state for an individual or group of selected voxels.

    *Priority:* Medium (should-have)
    *Likelihood of change:* Low
    *Justification:* It eases the process of user correction for widespread mistakes. There's minimal chance of change since it encapsulates what a user would require to restore a blank canvas for property assignment.

#### Non-Functional Requirements

1. **Ease of use (NF231):** Editing Manager shall ensure that all interactions covered in the above requirements (e.g. assigning properties, layer selection, undo/redo) can be completed in MAX_INTERACTIONS or less by USABILITY_PERCENTAGE of users.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Medium
   *Justification:* It ensures simple functionality to enhance usability in critical operations. The standards for what defines an intuitive interface may be altered to better reflect user familiarity with the property assignment process.

2. **Metadata update (NF232):** Editing Manager shall update the voxel property metadata with a latency of less than MAX_EDIT_LATENCY to allow consistent, synchronized model modification updates.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Medium
   *Justification:* It is crucial for guaranteeing prompt and effective updates to the model. The latency threshold may change in relation to technical feasibility as long as there is no significant lag that jeopardizes update consistency.

### (2.2.4) Export Manager

#### Functional Requirements

1. **Property validation (F241):** Export Manager shall validate that all voxels within the voxel model have been assigned a magnetization vector and material upon receiving a request to export a file for

printing (even if the assigned value is null to indicate no magnetization required).

*Priority:* Medium (should-have)
*Likelihood of change:* Medium
*Justification:* It helps users ensure minimal issues during the printing stage. It may require modification if assigning null magnetization to unmagnetized voxels is determined to be unnecessarily demanding for users.

2. **File export (F242):** Export Manager shall produce a standalone file (e.g. printer-ready file or export file), containing all metadata for each voxel, that the user can name and save locally on their personal device outside of the software.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It is an essential function that enables users to print their model with complete property assignments. Due to its critical nature, there's limited flexibility for change.

3. **Project export (F243):** Export Manager shall allow the user to export their model manually as a project file, that the user can name and save locally on their personal device.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low
   *Justification:* It is an essential function that enables users to create duplicates of their projects to support iteration (the equivalent of "Save as" functionality). With iteration being important to the design process, this is unlikely to change.

4. **Exportation progress tracker (F244):** Export Manager shall provide a progress bar that gives a visual indicator of how far along the software is in the exportation process.

   *Priority:* Low (could-have)
   *Likelihood of change:* High
   *Justification:* It can enhance the user experience by engaging users and reducing uncertainty, though not critical. Given the intensive nature of file export, changes may be required to improve progress tracking reliability.

5. **Model summary (F245):** Export Manager shall allow users the option to export a file that summarizes model statistics upon producing a printer-ready file of the model.

   *Priority:* Low (could-have)
   *Likelihood of change:* High
   *Justification:* It provides helpful information to the user regarding their overall model but is not critical to core functionality. The exported statistics will likely be adjusted to include data most relevant to a typical user.

### Non-Functional Requirements

1. **Fail safe (NF241):** Export Manager shall ensure the voxel data and metadata within a project file remain intact and unaltered in the event of an incomplete export, preserving the integrity of the

project file without loss.

*Priority:* High (must-have)
*Likelihood of change:* Low
*Justification:* It promotes reliability and trust, maintaining a positive user experience. It is crucial that previous work is not lost during a failed export, which means there's limited flexibility for change.

2. **Exportation performance (NF242):** Export Manager shall export metadata files for geometric models at a minimum rate of MIN_RATE.

   *Priority:* Low (could-have)
   *Likelihood of change:* High
   *Justification:* It ensures reduced wait time and improves overall user experience, though it is not critical. Changes to the MIN_RATE threshold may be necessary to sufficiently support high-demand operations.

3. **Ease of use (NF243):** Export Manager shall ensure projects can be exported and/or saved in under MAX_INTERACTIONS.

   *Priority:* Medium (should-have)
   *Likelihood of change:* Low
   *Justification:* Similar to previous Ease of use requirements, these operations should be as streamlined as possible to be intuitive and quick.

## (2.2.5) General Non-Functional Requirements

These requirements do not apply to a specfic component but instead to the system as a whole.

1. **Manual Accessibility (GEN1):** The system shall include a user's manual that details primary use cases, minimizes description to what is necessary, and is considered interpretable by primary users.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* It is important that the included manual exhibits these properties, otherwise the document will not be helpful at its intended purpose - both helping new users learn how to accomplish their tasks with the system and acting as a quick reference to experienced users.

2. **Implementation Maintainability (GEN2):** The system shall be implemented in a modular, maintainable, and extensible fashion.

   *Priority:* High (must-have)
   *Likelihood of change:* Low
   *Justification:* Ensuring the system exhibits these qualities is critical both during development to allow for easy modification and after, in the case that further revisions of the system are made. Given how these qualities are highly desirable, this requirement is inflexible.

# (S.3) Interfaces

Users will use this interface to interact with the generated 3D rendition of the CAD model while also viewing voxel layers to execute the core functionality of assigning magnetization and materials to voxels. It should demonstrate seamless and user-friendly navigation across multiple perspectives of the 3D model through rotation and resizing while allowing easy access to the different voxel layers and an intuitive process for property assignment and model modifications. The provided wireframe mockups showcase the simplest design that captures the essential functionalities in a well-organized interface that users will be able to navigate naturally without confusion.

Figures 4 thru 6 below are some initial, general mockups, communicating these ideas without enforcing a strict design layout.

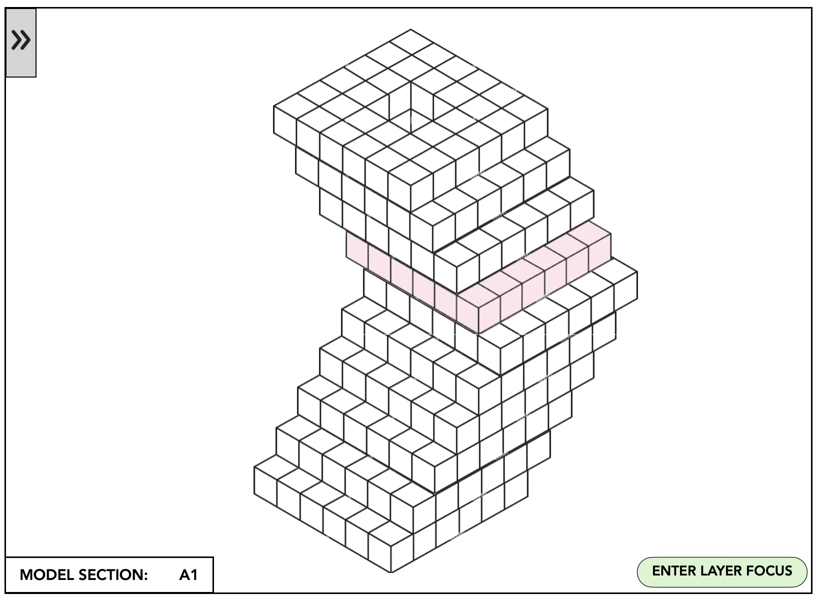


*Figure 5. Interface after initialize 3D rendition is generated.*

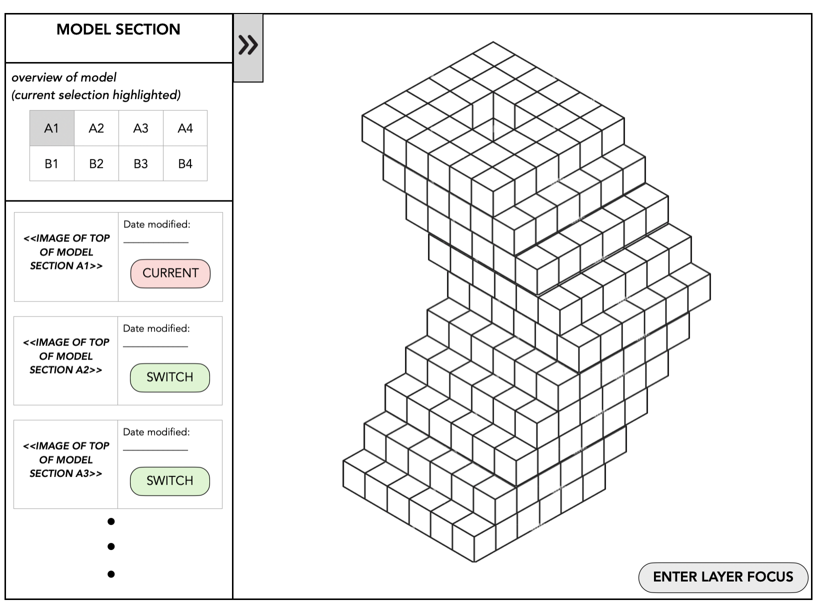


*Figure 6. Accessing partition division through interface.*

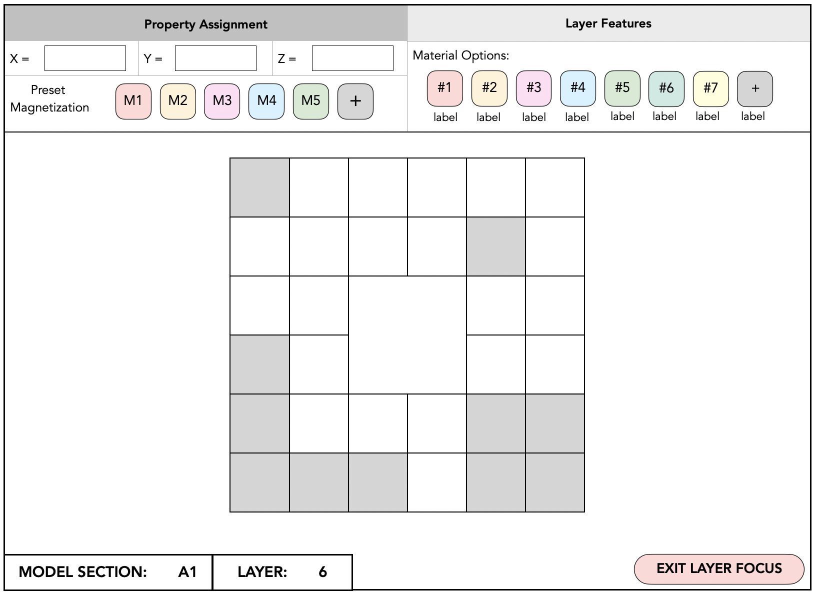


*Figure 7. Entering layer focus interface for property assignment.*

# (S.4) Detailed usage scenarios

This section provides detailed usage scenarios that expand on the high-level scenarios described in G.5. These scenarios incorporate the system components and secondary scenarios.

## (S.4.1) Import and Voxelize CAD Model

- **Use case**: UC1, UC2
- **Related requirements**: [F211], [F213], [F214], [F215], [F216], [NF211], [NF212], [NF213], [F221]
- **Primary Actor**: Researcher/Engineer
- **Supporting Actors**: System
- **Precondition**: User has a completed 3D CAD file and the system is running on their desktop/laptop.
- **Trigger**: User wants to begin the property assignment process for a new 3D model.
- **Main Success Scenario**:
  - 1. User opens the voxel property assignment system application.
  - 2. User navigates to "Import Model" and selects their CAD file.
  - 3. System validates the file format and displays file information (dimensions, triangle count, etc.).
  - 4. User specifies voxel size and model dimensions, before selecting option to import.
  - 5. System converts the file to a voxel grid with specified dimensions.
  - 6. System displays the voxelized model in the 3D viewer with layer navigation controls.
  - 7. System reports total number of voxels and layers in the model.
  - 8. User can now begin layer-by-layer property assignment.
- **Secondary Scenarios**:
  - 2.1. User wishes to begin with a blank project - System skips steps to import CAD file and begins with a blank project file.
  - 3.1. File format is invalid or corrupted - System displays error message and requests valid file.
  - 4.1. File is too large for available memory - System suggests reducing model complexity or

increasing system resources.
  - 5.1. Voxelization fails due to mesh issues - System provides diagnostic information and suggests mesh repair.
  - 6.1. System performance is slow during rendering - System provides optimization options or simplified view modes.
- **Success Postcondition** : CAD model is successfully converted to voxel grid and ready for magnetization assignment.

This scenario is important as it establishes the foundation for the entire project. The ability to reliably import and convert a CAD file to voxels is critical for users who need to work with existing designs rather than starting from scratch.

### (S.4.2) Assign Properties to Voxels within a Layer

- **Use case**: UC2, UC3
- **Related requirements**: [F221], [F223], [F224], [F225], [NF221], [NF222], [F231], [F232], [F233], [F234], [F235], [F237], [F2310], [NF231], [NF232]
- **Primary Actor**: Researcher/Engineer
- **Supporting Actors**: System
- **Precondition**: CAD model has been successfully imported, or the model being constructed has a set of voxels, user is viewing a specific layer in the 3D viewer.
- **Trigger**: User wants to assign magnetization vectors and/or material IDs to specific voxels within the current layer.
- **Main Success Scenario**:
  - 1. User navigates to a specific layer using the layer slider (bottom to top).
  - 2. System displays the 2D cross-section of voxels for the selected layer, including current magnetization states and material assignments.
  - 3. User selects individual voxels or multiple voxels using click/drag selection.
  - 4. System highlights selected voxels.
  - 5. User opens the magnetization panel and specifies a magnetization direction for that set of chosen voxels.
  - 6. System applies the selected magnetization vector to the chosen voxels.
  - 7. System updates the voxels being displayed to indicate that the selected voxels have been magnetized.
  - 8. User opens the material assignment panel and selects a material ID.
  - 9. System applies the selected material to the chosen voxels.
  - 10. System updates the voxels being displayed to indicate that the selected voxels have a material assignment.
  - 11. System saves the changes to the project file automatically.
  - 12. User can navigate to other layers and repeat the process.
- **Secondary Scenarios**:
  - 3.1. User accidentally selects wrong voxels - System offers an option to unselect all voxels.
  - 4.1. User wants to clear magnetization from selected voxels - System provides a 'Clear Vectors' option within magnetization panel.
  - 4.2. User wants to clear material assignments from selected voxels - System provides a 'Clear

Materials' option within material assignment panel.
  - 5.1. User does not need to specify a custom vector - System presents an option to select from a list of specified 'favourite' magnetization vectors.
  - 6.1. System performance degrades with many selected voxels - System displays error message and asks users to select fewer voxels.
  - 8.1. User needs to specify a new material ID - System provides a 'Specify new material ID' option within material assignment panel.
  - 11.1. User accidentally edits wrong voxels - System offers an 'Undo' option to revert changes to previous edit.
  - 12.1. User wants to copy magnetization pattern to other layers - System provides copy/paste functionality.
- **Success Postcondition**: Selected voxels in the current layer have been assigned the desired magnetization directions.

This scenario is improtant because it provides the core functionality of this system, which is allowing users to precisly control both magnetization vectors and material assignments at the voxel level. The layer by layer approach allows users to assign properties while also maintaining a 3D view of the structure. The ability to select multiple voxels and apply the same properties to all of them significantly reduces the time required compared to manual voxel-by-voxel assignment.

## (S.4.3) Edit and Modify Existing Voxel Properties

- **Use case**: UC3
- **Related requirements**: [F224], [F225], [F226], [F227], [NF223], [F231], [F233], [F236], [F238], [NF231], [NF232]
- **Primary Actor**: Researcher/Engineer
- **Supporting Actors**: System
- **Precondition**: User has already assigned properties to some voxels and wants to make changes without restarting the entire process.
- **Trigger**: User discovers an error in property assignment or wants to optimize property patterns.
- **Main Success Scenario**:
  - 1. User navigates to the layer containing voxels that need modification.
  - 2. System displays the current states of all voxels, with the option to indicating those with assigned magnetization vectors and colors indicating material assignments.
  - 3. User selects the voxels that need to be changed (individual or multiple selection).
  - 4. System highlights selected voxels and presents the magnetization direction and material assignment panels.
  - 5. User specifies a new magnetization vector within the presented direction panel.
  - 6. System applies the new magnetization to the selected voxels and updates the visual display accordingly.
  - 7. User selects a new material ID within the presented material panel
  - 8. System applies the new material assignment to the selected voxels and updates the visual display accordingly.
  - 9. System automatically saves the changes and maintains edit history.
  - 10. User continues with other modifications as needed.

- **Secondary Scenarios**:
  - 3.1. User wants to select all voxels in current layer - System provides "Select All" option.
  - 4.1. User wants to find voxels with specific magnetization - System provides a visual tool to highlight and select all voxels with a specified magnetization.
  - 4.2. User wants to find voxels with specific material ID - System provides a visual tool to highlight and select all voxels with a specified material ID.
  - 8.1. System loses power during editing - System has auto-save functionality to prevent data loss.
- **Success Postcondition**: Magnetization assignments have been successfully modified without losing other work.

This scenario is important because it addresses the critical need for iterative design refinement. Users often need to make adjustments to property assignments based on simulation results or design requirements. The ability to edit existing assignments without restarting the entire process saves significant time and reduces frustration, making the system practical for real-world research workflows.

## (S.4.4) Add and Delete Voxels from within a Layer

- **Use case**: UC4
- **Related requirements**: [F224], [F236], [F237], [F239], [NF231], [NF232]
- **Primary Actor**: Researcher/Engineer
- **Supporting Actors**: System
- **Precondition**: User has created a project (via import or from scratch) and wants to edit the voxels present.
- **Trigger**: User wishes to add or remove a voxel from the model.
- **Main Success Scenario**:
  - 1. User navigates to the layer they wish to add voxels to, or remove voxels from.
  - 2. User selects within the layer where they wish to place new voxel(s).
  - 3. System highlights the space where the new voxel(s) would be placed.
  - 4. User selects to add new voxel(s) at the indicated space.
  - 5. System creates new blank voxel(s) at the highlighted space and updates the visual display accordingly.
  - 6. User selects the voxel(s) they wish to delete.
  - 7. System highlights the voxel(s) that will be deleted.
  - 8. User selects to delete the voxel(s).
  - 9. System removes the voxel(s) and updates the visual display accordingly.
  - 10. System automatically saves these changes and maintains edit history.
- **Secondary Scenarios**:
  - 5.1 User makes a mistake in where voxels are being created - System provides 'Undo' option to revert to previous state.
  - 5.1.2 User makes a mistake in removing created voxels - System provides 'Redo' option.
  - 5.2 User wants to create voxels with pre-defined properties - System provides option to enter a material ID or vector.
  - 9.1 User makes a mistake in which voxels are deleted - System provides 'Undo' option to revert deletion.
- **Success Postcondition**: Voxels within the current layer have been successfully created or deleted to

the user’s desire.

This scenario is important as it reflects both the user’s ability to create voxel models from scratch, or correct any undesired voxel placements after conversion from a CAD file. Being able to edit models at this level is critical to success in satisfying the designer’s vision.

## (S.4.5) Export Voxel Property Assignment Data for 3D Printing

- **Use case**: UC5
- **Related requirements**: [F241], [F242], [F243], [F244], [NF241], [NF242], [NF243]
- **Primary Actor**: Researcher/Engineer
- **Supporting Actors**: Lab Operator, System
- **Precondition**: User has completed property assignments for all layers and is ready to prepare the data for 3D printing.
- **Trigger**: User wants to export the voxel metadata to be used by the custom 3D printer software.
- **Main Success Scenario**:
  - 1. User navigates to the "Export and Save" section of the application.
  - 2. System displays export options and file format information.
  - 3. User reviews property assignment summary (total voxels, layers, counts of each material ID, number of unique magnetization vectors).
  - 4. User selects the designated location for the system to export the file to.
  - 5. System validates all voxel data and checks for completeness.
  - 6. System generates the export file containing per-voxel location, magnetization vector, and material assignment metadata.
  - 7. System displays export confirmation with file size and location information.
  - 8. Lab operator receives the file and can load it into the custom printer software.
- **Secondary Scenarios**:
  - 5.1. System detects incomplete magnetization data - System warns user of missing magnetization assignments, gives user option to export anyway and highlights voxels with missing assignments.
  - 5.2. System detects incomplete material assignments - System warns user of missing material assignments, highlights voxels with missing assignments and prevents export.
  - 5.3. Export file is too large for available storage - System displays detailed error about storage space
  - 6.1. System encounters errors during export - System provides detailed error log and recovery options.
  - 7.1. Lab operator reports issues with exported file - System provides validation tools and format verification.
- **Success Postcondition**: Complete voxel magnetization data has been exported in the correct format for the 3D printer.

This scenario is important because it represents the last step in the workflow, ensuring that the users can take what they have been working on and successfully transfer it to a physical printing process. The export functionality must be reliable and produce a file that is compatible with the existing printing pipeline, while maintaining the magnetization data (CSV) throughout the transition.

### (S.4.6) Validate and Review Magnetization Design

- **Use case**: UC1, UC5
- **Related requirements**: [F221], [F222], [F223], [F224], [F226], [F227], [NF221], [NF222], [NF223]
- **Primary Actor**: Researcher/Engineer
- **Supporting Actors**: Supervisor, System
- **Precondition**: User has completed magnetization assignment and wants to validate the design before export and printing.
- **Trigger**: User wants to review the complete magnetization/material pattern and ensure it meets the their needs.
- **Main Success Scenario**:
  - 1. System displays the complete 3D model with magnetization and material visualization.
  - 2. User can rotate, zoom, and examine the model from different angles.
  - 3. System provides layer-by-layer navigation to review specific sections.
  - 4. User can filter the view to show only voxels with specific magnetization directions or material properties.
- **Secondary Scenarios**:
  - 4.1. Supervisor requests changes after review - User can return to editing mode.
- **Success Postcondition**: User has thoroughly reviewed the magnetization design and is confident it meets their needs.

This scenario is important because it ensures quality and validation before moving on to the final step of exporting the file. This prevents the expensive and time-consuming 3D printing process on a file that is not what's needed.

### (S.4.7) Save Voxel Project File for Later Editing

- **Use case**: UC5
- **Related requirements**: !!
- **Primary Actor**: Researcher/Engineer
- **Supporting Actors**: System
- **Precondition**: User has begun a project.
- **Trigger**: User wants to save progress to resume at a later time.
- **Main Success Scenario**:
  - 1. User navigates to the "Export and Save" section of the application.
  - 2. System displays export options and file format information.
  - 3. User selects "Save Project".
  - 4. System saves all progress to the current project file and informs the user the save was successful.
  - 5. System displays save information including file location.
- **Secondary Scenarios**:
  - 3.1. User wants to save progress to a different file - System provides "Save as" option and prompts user to enter a new filename and location.
  - 4.1. System encounters an error during save process - System provides detailed error log and recovery options.

- **Success Postcondition**: Project file is saved correctly and it is safe to close the program.

This scenario is important as users may not be able to complete all of the work in one sitting, or they may want to create many variations of the same base project during the design process. Saved project files must be complete and correct to maximise user satisfaction and reduce any user frustration with lost progress.

*Table 1. Traceability Matrix between Use Cases and Functional Requirements*

| | [S.4.1] | [S.4.2] | [S.4.3] | [S.4.4] | [S.4.5] | [S.4.6] | [S.4.7] |
|---|---|---|---|---|---|---|---|
| [F211] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F212] | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F213] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F214] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F215] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F216] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F221] | ☑ | ☑ | ☐ | ☐ | ☐ | ☑ | ☐ |
| [F222] | ☐ | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ |
| [F223] | ☐ | ☑ | ☐ | ☐ | ☐ | ☑ | ☐ |
| [F224] | ☐ | ☑ | ☑ | ☑ | ☐ | ☑ | ☐ |
| [F225] | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ |
| [F226] | ☐ | ☐ | ☑ | ☐ | ☐ | ☑ | ☐ |
| [F227] | ☐ | ☐ | ☑ | ☐ | ☐ | ☑ | ☐ |
| [F231] | ☐ | ☑ | ☑ | ☐ | ☐ | ☐ | ☐ |
| [F232] | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F233] | ☐ | ☑ | ☑ | ☐ | ☐ | ☐ | ☐ |
| [F234] | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F235] | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F236] | ☐ | ☐ | ☑ | ☑ | ☐ | ☐ | ☐ |
| [F237] | ☐ | ☑ | ☐ | ☑ | ☐ | ☐ | ☐ |
| [F238] | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ |
| [F239] | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ |
| [F2310] | ☐ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [F241] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |
| [F242] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |
| [F243] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| [F244] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |
| [F245] | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |

*Table 2. Traceability Matrix between Use Cases and Non-Functional Requirements*

| | [S.4.1] | [S.4.2] | [S.4.3] | [S.4.4] | [S.4.5] | [S.4.6] | [S.4.7] |
|---|---|---|---|---|---|---|---|
| [NF211] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [NF212] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [NF213] | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [NF221] | ☐ | ☑ | ☐ | ☐ | ☐ | ☑ | ☐ |
| [NF222] | ☐ | ☑ | ☐ | ☐ | ☐ | ☑ | ☐ |
| [NF223] | ☐ | ☐ | ☑ | ☐ | ☐ | ☑ | ☐ |
| [NF231] | ☐ | ☑ | ☑ | ☑ | ☐ | ☐ | ☐ |
| [NF232] | ☐ | ☑ | ☑ | ☑ | ☐ | ☐ | ☐ |
| [NF241] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |
| [NF242] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |
| [NF243] | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ |
| [GEN1] | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| [GEN2] | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |

*Table 3. Traceability Matrix between Use Cases and High-Level Use Cases*

| | [S.4.1] | [S.4.2] | [S.4.3] | [S.4.4] | [S.4.5] | [S.4.6] | [S.4.7] |
|---|---|---|---|---|---|---|---|
| UC1 | ☑ | ☐ | ☐ | ☐ | ☐ | ☑ | ☐ |
| UC2 | ☑ | ☑ | ☐ | ☐ | ☐ | ☐ | ☐ |
| UC3 | ☐ | ☑ | ☑ | ☐ | ☐ | ☐ | ☐ |
| UC4 | ☐ | ☐ | ☐ | ☑ | ☐ | ☐ | ☐ |
| UC5 | ☐ | ☐ | ☐ | ☐ | ☑ | ☑ | ☑ |

# (S.5) Verification and acceptance criteria

This section serves as a brief overview of AutoVox's verification and validation plan; for more details, visit VnVPlan/VnVPlan.pdf within the docs folder. As such, full coverage of all requirements and all tests to be executed are not detailed here; high-level details and the most important tests are explained.

## Functional Requirements

To ensure that the requirements are fulfilled to their intended functionality, several validation approaches will be applied.

- **Automated Testing**: The Import Manager includes the voxel configuration, model resizing, and project importing. These features are the basis of the problem, so they need to be tested to ensure correctness and consistency. To do this, automated testing will be implemented to oversee the correctness of the features.
- **Manual Testing**: The usability elements in this project includes the voxel navigation, model visualization, and property assignment, all of which are beneficial for the core functionality of the system. These features need to be tested to evaluate the overall user experience during the workflows of the system.
- **Integrated Testing**: This testing strategy will be used to confirm the integration between the Import, Visualisation, Property, and Exportation Managers. To ensure the exchange of any voxel and metadata between managers is handled correctly.
- **Stakeholder Pre-Development Meeting**: Gather feedback from stakeholder to fix or improve the system. The goal is to gather actionable feedback on the core functionality of the system in terms of the correctness of features. The quantitative metrics may not apply here directly however, the feedback will promote system improvement.

## Non-Functional Requirements

- **Scalability**: The system should handle voxel models containing up to MAX_VOXELS, with the ability to scale futher without significant performance degradation. The parts of the system that need to be tested are the Import and Visualisation Managers, as they include many of the voxel model features. This will be validated using load and stress testing, simulating large voxel models to confirm performance stability.
    - Covered requirements [NF211], [NF222]
- **Performance**: An important requirement that this system should meet is the performance which is required for the different components. The Import Manager should generate voxel files at a minimum rate of MIN_RATE, the Visualisation Manager should update perspectives in under MAX_VIEW_LATENCY, the Magnetization Manager must update metadata in under MAX_EDIT_LATENCY, and the Export Manager should generate voxel metadata files at a minimum rate of MIN_RATE. These performance requirements will be validated by setting benchmarks and running stress tests to confirm that each component consistently meets the thresholds.
    - Covered requirements [NF212], [NF221], [NF222], [NF232], [NF242]
- **Ease of Use**: To ensure that the system is easy to use for the stakeholders, the interface and workflows will be tested for clarity, minimal steps to complete the core tasks, and accessibility of commonly used features. For example, the Property Manager should enable group edits in MAX_INTERACTIONS or fewer.
    - Covered requirement [NF213], [NF223], [NF231], [NF243], [GEN1]
- **Compliance**: As this system will interact with other systems such as AutoCAD and the existing Java printing software, the system should be validated for compatibility with both input files it receives and output files it generates. This will be validated through integration testing with the receiving and target software.
    - Covered requirement [NF241]
- **Maintainability**: To ensure that the resulting implementation is extensible in future revisions, the system will be inspected for modularity, presence of comments, and naming conventions. Associated documentation detailing the design will also be evaluated based on completeness and readability.
    - Covered requirement [GEN2]

The most important scenario in S.4 is [S.4.2]: Assign Properties to Voxels within a Layer. This scenario is one of the primary reasons for this system to be created. Therefore, these are five tests that validate this scenario across multiple levels of testing include:

- **Layer Navigation and Voxel Selection Test**: Validate that the system correctly displays 2D cross-sections for selected layers and allows users to select individual or multiple voxels using click/drag selection. The system must highlight selected voxels and show their current magnetization state and material assignments.
  - Corresponds to [F223], [F224], [F225]
- **Magnetization Vector Assignment Test**: Validate that the system correctly applies selected magnetization directions to a chosen voxel and updates the visual display with appropriate colouring upon request. The system must maintain the magnetization state and allow users to navigate between layers while preserving assignments.
  - Corresponds to [F231], [F227]
- **Material Assignment Test**: Validate that the system correctly applies selected material ID to a chosen voxel and updates the visual display with appropriate colors. The system must maintain the material ID and allow users to navigate between layers while preserving assignments.
  - Corresponds to [F233], [F226]
- **Multi-Voxel Selection and Batch Assignment Test**: Validate the functionality for selecting multiple voxels and applying consistent properties to the entire selection. The system must handle performance gracefully when many voxels are selected and provide appropriate feedback.
  - Corresponds to [F225], [F231], [F233], [F238]
- **Undo and Clear Operations Test**: Validate the system’s ability to revert selections and clear property assignments. The system must provide 'Undo' options for accidental assignments and 'Clear' functionality for removing properties from selected voxels.
  - Corresponds to [F237], [F2310]
- **Auto-Save and Data Persistence Test**: Validate the system’s automatic saving functionality and data persistence capabilities. The system must save changes to the project file automatically and maintain data integrity during layer navigation and magnetization assignment operations.
  - Corresponds to [F236]

*Table 4. Traceability Matrix between Tests and Requirements*

| Test | Test Description | Link to Requirements |
|---|---|---|
| Test 1 | Validate layer navigation and voxel selection functionality | [F223], [F224], [F225] |
| Test 2 | Validate magnetization vector assignment and visual updates | [F231], [F227] |
| Test 3 | Validate material assignment and visual updates | [F233], [F226] |
| Test 4 | Validate multi-voxel selection and batch assignment | [F225], [F231], [F233], [F238] |
| Test 5 | Validate undo and clear operations | [F237], [F2310] |

| Test | Test Description | Link to Requirements |
|---|---|---|
| Test 6 | Validate auto-save and data persistence | [F236] |

# (P) Project

## (P.1) Roles and personnel

This section defines the roles and personnel that are involved in this project:

- Researchers
  - Researchers are the primary users of the voxel magnetization system and key collaborators in the development process. They inform the development team about functional requirements, provide feedback on system usability and workflow efficiency, and validate that the system meets experimental needs. They are responsible for designing and creating 3D CAD models for magnetized objects, using the system to assign magnetization directions to voxels, and validating that magnetization assignments meet experimental requirements.
- Lab Operators/Printer Owners
  - Lab Operators/Printer Owners are technical personnel who operate the custom 3D printer and serve as the first testers of the completed system. They consume exported voxel data files from the system, execute the custom printer workflow using the generated files, and ensure correct printer operation and output quality. They provide critical feedback on the integration between the voxel magnetization system and the existing printer workflow, helping identify issues and improvements needed for practical deployment.
- Manager
  - Managers are responsible for monitoring project deadlines and ensuring the team stays on track. They coordinate with course deadlines and group-established milestones, facilitate communication between team members, and track progress against project timeline. They are also responsible for creating comprehensive meeting reports as GitHub issues, and maintaining records of decisions, action items, and deadlines.
- Communicators
  - Communicators organize and schedule team meetings, facilitate meeting discussions and ensure agenda coverage, and coordinate meeting logistics (location, time, format). They also handle primary communication with supervisor and stakeholders, send formal emails to the Researcher, and convey important information between external parties and team.
- Document Reviewer
  - Document Reviewers review project documentation for consistency and quality, organize peer review processes for deliverables, and ensure proper formatting, grammar, and technical accuracy.
- Code Reviewer
  - Code Reviewers review all code contributions and pull requests, ensure code quality, consistency, and adherence to standards, and monitor integration of individual contributions. They should have strong programming skills, experience with code review processes and best practices, understanding of software testing and quality assurance, and knowledge of version control and development workflows.
- Software Developer
  - Software Developers are responsible for implementing the voxel magnetization system across frontend and backend components. They should have strong programming skills, experience with

web development frameworks, understanding of database systems, familiarity with version control systems, and ability to work with 3D visualization libraries and APIs.

# (P.2) Imposed technical choices

The following are the imposed technical choices that this system will take:

- The system will use a web-based architecture with browser-accessible frontend. This choice is imposed to ensure cross-platform compatibility and ease of deployment.
- The system will use a local server backend with local data storage on the client machine. It should also use a relational database management system, like PostgreSQL. This is required to keep track of all edits done.
- The system will use Git and GitHub for version control and project management. This choice is imposed by the development plan to ensure proper collaboration, task tracking, and issue management throughout the project lifecycle.
- The system will use GitHub Actions for continuous integration. This choice is imposed to automate code quality checks, testing, and deployment processes, ensuring consistent code quality and reducing manual overhead.

# (P.3) Schedule and Milestones

The development of AutoVox is planned over multiple sprints, beginning with comprehensive documentation and requirements analysis before transitioning to implementation. Our development strategy focuses on thorough planning and risk mitigation through proof of concept demonstrations, ensuring the system meets the specific needs of researchers working with magnetized 3D printed objects. The system addresses the critical bottleneck in current CAD workflows where researchers must manually rebuild objects voxel-by-voxel to assign magnetization directions, consuming hours per part and making iteration fragile.

**Sprint 1: Documentation and Requirements Foundation (September - October 2025)**

- **Epic 1: Problem Statement, Proof of Concept, and Development Plan** (September 22, 2025) Complete initial project documentation including problem analysis, stakeholder identification, and development methodology. This establishes the foundation for all subsequent work and ensures team alignment on project scope and approach. The problem statement defines the core challenge: enabling per-layer voxel viewing, selection (single or multiple voxels), and assignment of magnetization vectors and materials on top of an imported CAD design, eliminating the need to rebuild objects manually in COMSOL.
- **Epic 2: SRS and Hazard Analysis Development** (October 6, 2025) Develop comprehensive software requirements specification and conduct hazard analysis to identify potential risks and safety considerations in the voxel property assignment workflow. This phase ensures all functional and non-functional requirements are properly documented and validated, including the system's ability to handle STL mesh files, maintain configurable voxel dimensions, and export Java-readable files for the custom printer pipeline. This epic includes the completion of system architecture documentation (S1-S5) defining the four core components: Import Manager, Visualization Manager, Editing Manager, and Export Manager, along with their functional requirements, interfaces, and prioritization framework.

- **Epic 3: Verification and Validation Planning** (October 27, 2025) Create detailed verification and validation plan to ensure the system will meet all specified requirements. This includes defining test strategies, acceptance criteria, and validation methods for the voxel magnetization system, particularly focusing on the accuracy of voxelization, magnetization assignment, and export format compatibility with the existing 3D printer workflow.

**Sprint 2: Design and Proof of Concept (November 2025)**

- **Epic 4: System Architecture and Design Specification** (November 10, 2025) Develop initial system architecture and detailed design specifications for the voxel magnetization system, including web-based frontend UI design, local server backend architecture, and database schema. The design will incorporate frontend technologies for 3D visualization, backend services, and local data storage to maintain edit history and enable undo/redo functionality. This epic will detail the four core components: Import Manager (CAD file import and voxelization), Visualization Manager (3D visualization with responsive update latency), Editing Manager (layer-based voxel selection and property assignment), and Export Manager (Java-readable file export at minimum required rate).
- **Epic 5: Proof of Concept Demonstration** (November 17-26, 2025) Demonstrate core functionality including CAD file import, voxelization, and basic layer viewing capabilities. This milestone validates the technical feasibility of the approach and identifies any major technical risks that need to be addressed. The PoC will show the process of importing a CAD file, converting it to voxels with configurable dimensions, and exporting the altered file to the existing 3D printing software. UI is not needed for the PoC, only a capable product that can create a standard print within defined limits.

**Sprint 3: Core Implementation and Testing (January - February 2026)**

- **Epic 6: Final System Design Integration** (January 19, 2026) Complete final system design incorporating lessons learned from proof of concept demonstration and any requirement refinements. This includes detailed specifications for the layered UI that allows browsing from bottom to top, voxel selection capabilities, and property assignment with undo/redo functionality. The design will implement the system as a locally deployed desktop application with wireframe mockups showing the main interface for 3D model interaction, layer navigation, and property assignment controls. Priority implementation will focus on must-have requirements: CAD file import ([F211]), 3D image generation ([F221]), layer viewing ([F224]), magnetization assignment ([F231]), material assignment ([F233]), and export functionality ([F242]).
- **Epic 7: Revision 0 System Demonstration** (February 2-11, 2026) Demonstrate working system with core functionality including voxel selection, property assignment, and export capabilities. This milestone shows a functional system that meets the primary requirements: importing mesh CAD files, providing layered UI for voxel selection, enabling magnetization and material assignment, and generating Java-readable exports for the custom printer. The demonstration will showcase the complete workflow from CAD import through the Import Manager, 3D visualization via the Visualization Manager, property assignment through the Editing Manager, and final export via the Export Manager, validating the component architecture and interface design.

**Sprint 4: Validation and Final Delivery (March - April 2026)**

- **Epic 8: Verification, Validation, and Extras** (March 9, 2026) Complete verification and validation activities, documenting test results and system performance. Deliver user manual and usability report

as specified in the problem statement. The user manual will provide comprehensive guidance on using the software's key features that differentiate it from normal CAD software, while the usability report will document findings from testing with researchers and lab operators. Validation will ensure all must-have requirements are met: system supports up to MAX_VOXELS ([NF211], [NF222]), maintains MAX_VIEW_LATENCY image update latency ([NF221]), provides MAX_EDIT_LATENCY metadata updates ([NF232]), and achieves MIN_RATE import/export performance ([NF212], [NF242]).
- **Epic 9: Final System Demonstration** (March 23-26, 2026) Present completed system with all features implemented, tested, and validated. Demonstrate full workflow from CAD import to printer-ready export, showing how the system eliminates the manual voxel-by-voxel rebuilding process and enables efficient iteration on property assignment designs.
- **Epic 10: Final Documentation and EXPO** (April 6, 2026 & TBD) Deliver final documentation package including updated SRS, design documents, user manual, and all supporting materials. Present project at capstone expo, showcasing the voxel magnetization system to broader academic and industry audience, highlighting its potential impact on research workflows involving magnetized 3D printed objects.

---

**Justification for Proof of Concept Focus**

The proof of concept demonstration is strategically positioned to address the two primary risks identified in the development plan: unfamiliarity with dependencies for file conversion and ensuring the automated process is more efficient than manual rebuilding. By demonstrating successful CAD file import, voxelization with configurable dimensions, and export to the existing Java-based printer program, we validate the core technical approach before investing in full UI development.

This approach allows us to identify and address any major technical challenges early in the development process, particularly around file format handling, voxelization algorithms, and integration with the existing printer workflow. The PoC will use command-line interfaces to focus on the core functionality, ensuring that the fundamental technical approach is sound before building the user interface layer.

The sprint-based approach also aligns with the team's development methodology, which emphasizes thorough planning, risk mitigation, and iterative validation. By completing comprehensive documentation and requirements analysis first, we ensure that the implementation phase is well-informed and addresses the specific needs of researchers working with magnetized 3D printed objects.

# (P.4) Tasks and Deliverables

**Sprint 1 Backlog**

The following user stories comprise the backlog for the first sprint, focusing on delivering the core documentation and requirements foundation for AutoVox. Each story is written from the stakeholder's perspective and includes an assessment of its business value and a technical estimate in story points. The story points are assigned based on complexity and effort, following the scale:

- **1 point**: Simple task using existing tools and templates.
- **2 points**: Task requiring research, analysis, and documentation.

- **5 points**: Complex task with significant effort and potential unknowns.

---

**User Story 1: Understand Current Workflow Pain Points**

- **Story**: As a researcher, I want the development team to understand my current CAD workflow bottlenecks so that they can design a solution that addresses my real pain points with voxel-by-voxel property assignment.
- **Business Value**: High Understanding current workflow bottlenecks is fundamental to defining the problem and ensuring our solution provides genuine value to researchers and lab operators.
- **Estimate**: **2 story points**
- **Reasoning**: This task involves conducting research on current CAD workflows, interviewing stakeholders, and documenting the specific bottlenecks in voxel-by-voxel property assignment processes. It requires analysis of existing tools like COMSOL and understanding the manual rebuilding process. The research component and stakeholder interaction justify **2 story points**.

---

**User Story 2: Define System Requirements Based on User Needs**

- **Story**: As a lab operator, I want the development team to interview me and other stakeholders to understand our needs and constraints so that the system will actually work for our daily operations.
- **Business Value**: High Direct stakeholder input ensures our solution meets real user needs and aligns with research goals and operational constraints.
- **Estimate**: **2 story points**
- **Reasoning**: Conducting structured interviews with multiple stakeholder groups requires scheduling, preparation of interview questions, conducting sessions, and synthesizing findings. The coordination effort and analysis required justify **2 story points**.

---

**User Story 3: Establish Project Foundation and Scope**

- **Story**: As a supervisor, I want the development team to clearly define the project scope and constraints so that we can focus efforts on achievable goals within our timeline and budget.
- **Business Value**: High Clear scope definition prevents scope creep and ensures the project delivers value within the available timeframe and resources.
- **Estimate**: **1 story point**
- **Reasoning**: This task involves synthesizing stakeholder input and technical constraints into a clear project scope document. It's primarily documentation work using established templates and frameworks, making it straightforward and assigned **1 story point**.

---

**User Story 4: Validate Technical Approach with Proof of Concept**

- **Story**: As a researcher, I want the development team to create a detailed proof of concept plan to validate their technical approach before full implementation so that we can be confident the solution will work.

- **Business Value**: High A well-planned proof of concept reduces technical risk and ensures we can deliver the core functionality within project constraints.
- **Estimate**: **2 story points**
- **Reasoning**: Creating a comprehensive PoC plan requires technical analysis of file formats, voxelization algorithms, and integration requirements. It involves research into existing libraries and tools, and planning the demonstration approach. The technical complexity and research component justify **2 story points**.

---

**User Story 5: Establish Development Team Structure**

- **Story**: As a supervisor, I want the development team to establish clear roles and communication protocols so that the project runs efficiently and we can track progress effectively.
- **Business Value**: Medium Clear roles and communication protocols improve team efficiency and reduce coordination overhead.
- **Estimate**: **1 story point**
- **Reasoning**: This task involves defining team roles based on the development plan, setting up communication channels, and establishing meeting schedules. It's primarily organizational work using established best practices, making it simple and assigned **1 story point**.

---

**User Story 6: Set Up Project Management Infrastructure**

- **Story**: As a supervisor, I want the development team to set up proper project management tools so that we can track progress, manage version control, and ensure efficient collaboration throughout the project.
- **Business Value**: Medium Proper tool setup enables efficient development, version control, and project tracking throughout the project lifecycle.
- **Estimate**: **1 story point**
- **Reasoning**: Setting up GitHub repository, project boards, and development tools involves following established procedures and templates. It's straightforward configuration work, assigned **1 story point**.

---

**User Story 7: Define Complete System Requirements**

- **Story**: As a lab operator, I want the development team to define detailed functional requirements for all system components so that the system will have all the features I need for my workflow.
- **Business Value**: High Detailed functional requirements provide the foundation for system design and implementation, ensuring all stakeholder needs are addressed.
- **Estimate**: **5 story points**
- **Reasoning**: This task requires detailed analysis of each component's responsibilities, defining specific requirements (F211-F245), and ensuring consistency across the system. It involves complex technical analysis and coordination between different functional areas, justifying **5 story points**. ---

**User Story 8: Establish Performance and Quality Standards**

- **Story**: As a researcher, I want the development team to define performance targets and quality attributes so that the system will be fast enough and reliable enough for my research needs.
- **Business Value**: High Non-functional requirements ensure the system meets performance expectations and can handle the required workload.
- **Estimate**: **2 story points**
- **Reasoning**: Defining performance targets (NF211-NF213, NF221-NF223, NF231-NF232, NF241-NF243) requires analysis of system requirements, stakeholder expectations, and technical constraints. It involves research into performance benchmarks and setting realistic targets, justifying **2 story points**. ---

**User Story 9: Create System Architecture Documentation**

- **Story**: As a supervisor, I want the development team to create visual architecture diagrams so that I can understand how the system will work and ensure it meets our technical requirements.
- **Business Value**: Medium Architecture diagrams provide clear visualization of system design and facilitate communication with stakeholders and team members.
- **Estimate**: **2 story points**
- **Reasoning**: Creating PlantUML diagrams requires understanding component interactions, data flow, and system architecture. It involves design work and technical documentation, justifying **2 story points**.

---

**User Story 10: Design User Interface Mockups**

- **Story**: As a lab operator, I want the development team to create wireframe mockups of the main user interface so that I can see how the system will look and provide feedback before development starts.
- **Business Value**: High Wireframe mockups provide early visualization of the user experience and help identify potential usability issues.
- **Estimate**: **2 story points**
- **Reasoning**: Creating wireframes requires understanding user workflows, component interactions, and UI/UX principles. It involves design work and stakeholder feedback integration, justifying **2 story points**.

---

**User Story 11: Identify System Risks and Safety Measures**

- **Story**: As a supervisor, I want the development team to conduct a comprehensive hazard analysis so that we can identify potential risks and ensure appropriate safety measures are implemented.
- **Business Value**: High Hazard analysis helps identify potential failure modes and ensures appropriate safety measures are implemented.
- **Estimate**: **2 story points**
- **Reasoning**: Conducting hazard analysis requires systematic evaluation of potential failure modes, their consequences, and mitigation strategies. It involves technical analysis and risk assessment methodologies, justifying **2 story points**.

**User Story 12: Prioritize Features by Business Value**

- **Story**: As a supervisor, I want the development team to prioritize all requirements using MoSCoW methodology so that we focus on must-have features first and deliver the most value within our constraints.
- **Business Value**: High Requirement prioritization ensures critical features are delivered first and helps manage scope within project constraints.
- **Estimate**: **1 story point**
- **Reasoning**: Prioritizing requirements involves applying MoSCoW methodology to categorize features based on stakeholder value and technical feasibility. It's primarily analytical work using established frameworks, assigned **1 story point**.

---

**User Story 13: Document System Integration Requirements**

- **Story**: As a lab operator, I want the development team to document all system interfaces and external dependencies so that the system will integrate properly with our existing equipment and software.
- **Business Value**: Medium Clear documentation of interfaces and dependencies helps identify integration challenges and external system requirements.
- **Estimate**: **1 story point**
- **Reasoning**: Documenting interfaces involves cataloging component interactions and external system dependencies. It's primarily documentation work using established templates, assigned **1 story point**.

---

**User Story 14: Establish Quality Assurance Strategy**

- **Story**: As a supervisor, I want the development team to define comprehensive testing strategies so that the system will be reliable and defect-free when delivered.
- **Business Value**: High Testing strategies ensure code quality and reduce the risk of defects in the final system.
- **Estimate**: **2 story points**
- **Reasoning**: Defining testing strategies requires understanding component responsibilities, identifying test scenarios, and selecting appropriate testing frameworks. It involves technical analysis and planning, justifying **2 story points**.

---

**User Story 15: Plan System Integration Testing**

- **Story**: As a lab operator, I want the development team to plan integration testing so that all system components work together correctly and the complete workflow functions as expected.
- **Business Value**: High Integration testing ensures the system functions as a cohesive whole and components interact properly.
- **Estimate**: **2 story points**

- **Reasoning**: Planning integration testing requires understanding component interfaces, data flow, and interaction patterns. It involves technical analysis and test scenario design, justifying **2 story points**.

---

**User Story 16: Design End-to-End Testing Scenarios**

- **Story**: As a researcher, I want the development team to design comprehensive system testing scenarios so that the complete workflow from CAD import to printer export functions correctly.
- **Business Value**: High System testing scenarios ensure the complete workflow functions correctly and meets user requirements.
- **Estimate**: **2 story points**
- **Reasoning**: Designing system test scenarios requires understanding complete user workflows, edge cases, and performance requirements. It involves analysis of user stories and technical requirements, justifying **2 story points**.

---

**User Story 17: Define User Acceptance Criteria**

- **Story**: As a lab operator, I want the development team to define user acceptance testing criteria so that I can validate the system meets my expectations before it's considered complete.
- **Business Value**: High User acceptance testing criteria provide clear validation standards and ensure stakeholder satisfaction.
- **Estimate**: **1 story point**
- **Reasoning**: Creating acceptance criteria involves translating stakeholder requirements into testable conditions. It's primarily documentation work using established templates, assigned **1 story point**.

---

**User Story 18: Establish Performance Validation Procedures**

- **Story**: As a researcher, I want the development team to establish procedures for measuring and validating system performance so that the system will meet the speed and reliability requirements for my research.
- **Business Value**: Medium Performance benchmarking procedures ensure the system meets specified performance targets.
- **Estimate**: **1 story point**
- **Reasoning**: Establishing benchmarking procedures involves defining measurement approaches and success criteria. It's primarily planning work using established methodologies, assigned **1 story point**.

---

**User Story 19: Plan User Validation Activities**

- **Story**: As a supervisor, I want the development team to plan validation activities with researchers and lab operators so that we can ensure the system meets real user needs before final delivery.
- **Business Value**: High Stakeholder validation ensures the system provides real value and meets user expectations.

- **Estimate**: **1 story point**
- **Reasoning**: Planning validation activities involves scheduling stakeholder sessions and preparing validation materials. It's primarily coordination work, assigned **1 story point**.

---

**User Story 20: Prepare Comprehensive Test Documentation**

- **Story**: As a supervisor, I want the development team to document test data requirements and create comprehensive test cases so that we can ensure thorough testing coverage and quality assurance.
- **Business Value**: Medium Well-documented test cases ensure thorough testing coverage and facilitate test execution.
- **Estimate**: **1 story point**
- **Reasoning**: Documenting test cases involves creating test scenarios and identifying required test data. It's primarily documentation work using established templates, assigned **1 story point**.

---

**Total Estimated Story Points for Sprint 1**: **35 story points**

---

**Notes on Business Value and Estimates**

- **High Business Value**: Features critical to project success that directly impact system quality and stakeholder satisfaction.
- **Medium Business Value**: Features that enhance project efficiency and quality but are not critical for initial delivery.
- **Story Point Assignments**:
- **1 point**: Simple tasks using existing tools, templates, and established procedures.
- **2 points**: Tasks requiring research, analysis, stakeholder interaction, and technical planning.
- **5 points**: Complex tasks involving significant technical analysis and coordination across multiple areas.

---

**Professional Summary**

The sprint backlog for Sprint 1 is strategically designed to establish a solid foundation for AutoVox by focusing on comprehensive requirements analysis and system design. We prioritize tasks that deliver the most business value while efficiently utilizing resources through established methodologies and tools.

Simple tasks, assigned **1 story point**, are quick wins that utilize existing templates and procedures, allowing for rapid progress in documentation and planning activities. Tasks with **2 story points** involve research, stakeholder interaction, and technical analysis, requiring moderate effort and coordination. The **5 story point** task represents substantial work in defining functional requirements, which is essential for system success but involves complex technical analysis and coordination.

By providing detailed accounts for each user story and the rationale behind the story point estimates, we ensure clarity and alignment within the development team. This detailed planning allows for effective resource allocation, risk management, and sets clear expectations for deliverables.

Our approach aligns with professional software engineering practices, emphasizing:

- **Thoroughness**: Comprehensive requirements analysis and system design before implementation.
- **Stakeholder Focus**: Direct engagement with researchers, lab operators, and supervisors to ensure real value delivery.
- **Risk Mitigation**: Early identification of technical challenges and critical assumptions.
- **Quality**: Detailed planning and validation strategies to ensure high-quality deliverables.

By the end of Sprint 1, we aim to have a comprehensive understanding of the problem domain, detailed system requirements, and a clear plan for implementation, laying a solid foundation for subsequent development phases.

---

**Sprint 2 Backlog**

The following user stories comprise the backlog for Sprint 2, focusing on system design and proof of concept development for AutoVox. This sprint transitions from requirements analysis to technical design and validation of core functionality.

---

**User Story 21: Design Complete System Architecture and User Interface**

- **Story**: As a lab operator, I want the development team to design the complete system architecture and user interface so that the system will efficiently handle my voxel data, property assignment workflows, and provide an intuitive interface for interacting with my models.
- **Business Value**: High A well-designed system architecture and user interface are essential for storing and retrieving voxel data efficiently while providing the 3D visualization capabilities that are central to the system's functionality.
- **Estimate**: **5 story points**
- **Reasoning**: This comprehensive task involves designing the complete system architecture, user interface design, and creating data flow documentation. It requires understanding user workflows, system requirements, and technical architecture decisions. The complexity and coordination across multiple areas justify **5 story points**.

---

**User Story 22: Design File Import and System Reliability Features**

- **Story**: As a researcher, I want the development team to design robust file import functionality and system reliability features so that I can work with my existing CAD designs, convert them to voxel representations, and have confidence the system will handle errors gracefully.
- **Business Value**: High Reliable file import functionality and system reliability features are essential for users to work with their existing CAD designs and have confidence in the system's ability to handle issues gracefully.

- **Estimate**: **3 story points**
- **Reasoning**: This task involves designing file import capabilities for STL files, comprehensive error handling and validation strategies, and edit history/undo-redo functionality. It requires understanding file formats, error handling, validation strategies, and user interaction patterns, justifying **3 story points**.

---

**User Story 23: Implement Core CAD Processing and Export Functionality**

- **Story**: As a researcher, I want the development team to implement the ability to import my STL files, convert them to voxel grids, and export them to printer-compatible files so that I can complete the full workflow from CAD import to printer-ready output.
- **Business Value**: High Core CAD processing and export functionality is fundamental to the system's ability to process user CAD files, convert them to the required format, and integrate with the existing 3D printer workflow.
- **Estimate**: **8 story points**
- **Reasoning**: This comprehensive implementation involves creating the file parser for STL, implementing voxelization algorithms, creating efficient voxel data structures, implementing export functionality for printer integration, and creating a command-line interface for testing. It requires understanding file format specifications, 3D geometry algorithms, performance optimization, and integration testing. The significant development effort and technical complexity justify **8 story points**.

---

**User Story 24: Validate Proof of Concept and Create Demonstration**

- **Story**: As a supervisor, I want the development team to validate the proof of concept with sample data and create demonstration materials so that we can showcase the technical feasibility and ensure the system works correctly before full development.
- **Business Value**: High Proof of concept validation and demonstration materials are essential for validating the technical approach, identifying issues early, and showcasing system capabilities to stakeholders.
- **Estimate**: **3 story points**
- **Reasoning**: This task involves testing with sample CAD files, validating outputs, testing export compatibility with existing printer software, documenting performance metrics, identifying bottlenecks, preparing demonstration materials, and ensuring reliable execution. It requires coordination with stakeholders, technical troubleshooting, analysis, and preparation work, justifying **3 story points**.

---

**Total Estimated Story Points for Sprint 2: 16 story points**

---

**Sprint 3 Backlog**

The following user stories comprise the backlog for Sprint 3, focusing on core implementation and testing of AutoVox. This sprint delivers the working system with core functionality.

---

**User Story 25: Refine System Design and Create Complete User Interface**

- **Story**: As a lab operator, I want the development team to refine the system design based on proof of concept learnings and create detailed user interface specifications so that the final system will be optimized and I can see exactly how it will work.
- **Business Value**: High System design refinement and complete user interface specifications ensure the system is optimized for performance and reliability while providing clear visualization of the user experience.
- **Estimate**: **5 story points**
- **Reasoning**: This comprehensive task involves refining system architecture based on PoC results, creating detailed UI/UX specifications, expanding wireframes, defining interaction patterns, designing responsive layouts, and finalizing system architecture documentation. It requires design work, stakeholder feedback integration, technical analysis, and architecture decisions, justifying **5 story points**.

---

**User Story 26: Implement Complete Voxel Property Assignment System**

- **Story**: As a researcher, I want the development team to implement the complete voxel property assignment system so that I can import my CAD files, view them in 3D, navigate through layers, select individual voxels, assign magnetization vectors and material IDs, and export the results to printer-compatible files.
- **Business Value**: High The complete voxel magnetization system provides all core functionality including CAD import, 3D visualization, magnetization assignment, material assignment, and printer integration - the primary purpose of the entire system.
- **Estimate**: **13 story points**
- **Reasoning**: This comprehensive implementation delivers the complete end-to-end workflow including CAD file import and voxelization, interactive 3D visualization with rotation and zoom capabilities, layer-by-layer navigation and voxel selection tools, magnetization and material property assignment interfaces, auto-save functionality for data preservation, and export capabilities for printer integration. It requires significant development effort to create the full user interface, implement complex 3D rendering and interaction, manage application state and data persistence, and ensure all features work together seamlessly. The scope spans the entire system functionality from initial file import through final export, justifying **13 story points**.

---

**User Story 27: Create Comprehensive Testing and Demonstration**

- **Story**: As a supervisor, I want the development team to create comprehensive testing for all system components and prepare demonstration scenarios so that the system will be reliable, defect-free, and we can showcase its capabilities.
- **Business Value**: High Comprehensive testing ensures system reliability and facilitates future

maintenance, while well-prepared demonstrations showcase system capabilities and validate functionality.
- **Estimate**: **6 story points**
- **Reasoning**: This task involves creating comprehensive unit and integration tests for all system components, conducting integration testing, and preparing demonstration scenarios and test data. It requires understanding component behavior, defining test scenarios, implementing automated tests, and coordination work, justifying **6 story points**.

---

**Total Estimated Story Points for Sprint 3**: **24 story points**

---

### Sprint 4 Backlog

The following user stories comprise the backlog for Sprint 4, focusing on validation, final delivery, and project completion for AutoVox.

---

#### User Story 28: Implement Additional User Experience Features

- **Story**: As a researcher, I want the development team to implement additional features like model size adjustment, favorite magnetization settings, voxel block size configuration, and CSV export options so that I can work more efficiently and customize the system for my specific needs.
- **Business Value**: Medium Additional user experience features improve workflow efficiency and provide flexibility beyond core requirements, enhancing the overall user experience.
- **Estimate**: **4 story points**
- **Reasoning**: Implementing these features requires UI development, state management, user interaction design, configuration UI, file generation logic, and user interface updates. It involves moderate development effort, user experience considerations, and testing, justifying **4 story points**.

---

#### User Story 29: Validate System Performance and Conduct User Testing

- **Story**: As a researcher, I want the development team to validate that the system can handle large models with up to MAX_VOXELS and conduct user acceptance testing so that I can work with complex models and the system meets real user needs.
- **Business Value**: High Performance validation ensures the system meets scalability requirements, while user acceptance testing validates the system meets real user needs and works effectively in actual workflows.
- **Estimate**: **8 story points**
- **Reasoning**: This comprehensive task involves validating MAX_VOXEL support, testing latency requirements, validating export performance, conducting user acceptance testing with stakeholders, and system integration testing with external printer software. It requires creating large test datasets, measuring performance metrics, optimizing bottlenecks, test planning, stakeholder coordination, and technical testing, justifying **8 story points**.

---

**User Story 30: Create Complete User Documentation and Installation Guide**

- **Story**: As a lab operator, I want the development team to create comprehensive user documentation and installation guides so that I can effectively use the system, troubleshoot issues, and set it up successfully in our lab environment.
- **Business Value**: High Complete user documentation and installation guides are essential for user adoption, effective system utilization, and successful deployment.
- **Estimate**: **3 story points**
- **Reasoning**: Creating comprehensive documentation involves understanding user workflows, creating clear instructions, including screenshots and examples, documenting usability testing results, creating installation procedures, and testing installation processes. It requires technical writing, user experience analysis, and documentation work, justifying **3 story points**.

---

**User Story 31: Prepare Final Demonstration and Complete Project Delivery**

- **Story**: As a supervisor, I want the development team to prepare final demonstrations, conduct system validation, complete all project documentation, and prepare EXPO materials so that we can showcase the complete system capabilities and properly conclude the project.
- **Business Value**: High Final demonstration preparation, system validation, complete documentation, and EXPO preparation ensure successful project delivery, stakeholder confidence, and proper project closure.
- **Estimate**: **11 story points**
- **Reasoning**: This comprehensive final delivery task involves preparing demonstration scenarios, creating sample files, practicing demonstrations, preparing presentation materials, setting up demonstration environments, conducting final system validation, addressing bugs, completing all project documentation, preparing handover materials, creating EXPO presentations, and conducting project closure activities. It requires significant coordination, testing, documentation work, presentation design, and project analysis, justifying **11 story points**.

---

**Total Estimated Story Points for Sprint 4**: **26 story points**

---

**Overall Project Summary**

**Total Estimated Story Points Across All Sprints**: **101 story points**

- **Sprint 1**: 35 story points (Documentation and Requirements Foundation)
- **Sprint 2**: 16 story points (Design and Proof of Concept)
- **Sprint 3**: 24 story points (Core Implementation and Testing)
- **Sprint 4**: 26 story points (Validation and Final Delivery)

The comprehensive user story breakdown provides detailed planning for the entirety of AutoVox. Each story is written from the stakeholder's perspective (researchers, lab operators, and supervisors) and focuses on user value rather than technical implementation details. Technical tasks have been consolidated into meaningful user-facing features, ensuring thorough coverage of all requirements while eliminating implementation-specific details like "plan Three.js integration" or "design React component hierarchy." This approach provides clear task definitions and realistic effort estimation for successful project delivery.

---

# (P.5) Required technology elements

This section outlines the essential external technology elements, encompassing software libraries, hardware, and development environments, that are important for the successful development and operation of the voxel magnetization system.

- CAD File Processing Capabilities
  - The system requires the ability to read and process mesh CAD files in the STL format, as discussed within E3. This capability is critical for the voxelization process that converts 3D models into voxel grids. The availability and reliability of CAD file processing directly impacts the system's ability to import designs from common CAD tools like AutoCAD and SolidWorks. Without this capability, the core functionality of the system cannot be implemented.
- 3D Visualization and Interaction Capabilities
  - The system requires interactive 3D visualization capabilities for layer-by-layer voxel viewing and selection. These capabilities must support efficient rendering of large voxel grids and provide user interaction for voxel selection. The performance and capabilities directly affect the user experience and system responsiveness, particularly when handling complex 3D models with many voxels.
- Data Persistence and Storage Capabilities
  - The system requires structured storage capabilities for voxel geometries, materials, metadata, user configurations, and project files. The storage system must provide reliable data persistence, efficient querying capabilities, and data integrity features. The availability and performance of storage capabilities is critical for maintaining system state and ensuring data consistency across user sessions.
- Development and Deployment Environment Requirements
  - The system requires commodity hardware (laptop/desktop) sufficient for interactive layer viewing and voxel selection. The development environment must support both Windows and Linux platforms. The availability of appropriate development hardware is essential for system development and testing.

# (P.6) Risk and Mitigation Analysis

This section identifies potential risks and threats that could impact the successful delivery of the system. It outlines proactive mitigation strategies to address these challenges:

## (P.6.1) Risks

- **Risk #1: Unfamiliarity with CAD File Processing Libraries**
  - The team has limited experience with libraries required for reading and processing mesh CAD files in the STL format. This unfamiliarity may result in difficulties with file conversion and voxelization, which are core features of the system. The risk is much larger during the proof of concept demonstration phase when the team will be largely unfamiliar with both the required libraries and Dr. Onaizah's existing codebase. To mitigate this risk, the team will allocate additional time for library research and experimentation during the early development phases. The proof of concept demonstration will focus specifically on validating the file conversion pipeline with simple CAD files, and the team will seek guidance from Dr. Onaizah and Kaitlyn Clancy on existing printer workflow integration.
- **Risk #2: Performance Issues with 3D Rendering and Large Voxel Grids**
  - The system must efficiently render and interact with large voxel grids, which may contain thousands of individual voxels. Initial iterations may experience performance issues due to lack of team experience with 3D rendering libraries. High memory usage and slow responsiveness could result from inefficient handling of large voxel grids, reducing the user experience quality. To mitigate this risk, the proof of concept will focus on rendering still images as a first step, and the team will research optimization strategies for 3D rendering. The system will implement progressive loading and level-of-detail techniques to manage memory usage and maintain responsiveness.
- **Risk #3: Integration Challenges with Existing 3D Printer Workflow**
  - The system must export data in a sufficient format for integration with Dr. Onaizah's existing custom 3D printer workflow. The team has limited understanding of the existing printer system and may face challenges in ensuring compatibility and correct data format. This integration is critical for the system's success, as it enables the exported voxel data to be consumed by the existing printer system. To mitigate this risk, the team will schedule regular meetings with Dr. Onaizah and Kaitlyn Clancy to understand the existing workflow requirements. The proof of concept will include validation of the export format, and the team will maintain close communication with the printer operators throughout development.

## (P.6.2) Threats

- **Threat #1: Technology Dependency Failures**
  - The system depends on several external technologies including CAD file processing libraries, 3D rendering libraries, and database systems. If any of these technologies become unavailable, the project could be derailed. To address this threat, the team will research multiple library options for each critical component and maintain awareness of technology roadmaps. The system will be designed with modular layered architecture (pipeline) to allow for technology substitution if needed.
- **Threat #2: Schedule Delays Due to Technical Complexity**
  - The technical complexity of voxelization, 3D rendering, and printer integration may lead to schedule delays that could impact deliverable deadlines. The team's limited experience with these technologies increases the likelihood of unexpected technical challenges. To address this threat, the team will maintain conservative time estimates for technical tasks and implement regular milestone check-ins. The proof of concept demonstration will serve as an early validation point to

identify and address major technical risks before full implementation.

- **Threat #3: Stakeholder Availability and Communication Gaps**
    - The project success depends heavily on communication with Dr. Onaizah, Kaitlyn Clancy, and other stakeholders. If these key personnel become unavailable or communication breaks down, the project could lose critical guidance and validation. To address this threat, the team will maintain regular bi-weekly updates with Dr. Onaizah and establish backup communication channels. All stakeholder interactions will be documented to ensure efficient knowledge transfer.

### (P.6.3) SWOT Analysis

**Strengths:** - Strong team collaboration and established communication protocols - Clear project scope and well-defined requirements - Access to domain expertise through Dr. Onaizah and Kaitlyn Clancy - Comprehensive development plan with regular milestone check-ins

**Weaknesses:** - Limited team experience with CAD file processing and 3D rendering libraries - Unfamiliarity with existing 3D printer workflow and integration requirements - Potential performance challenges with large voxel grid rendering

**Opportunities:** - Potential for significant impact on research workflow efficiency - Opportunity to learn advanced 3D visualization and rendering techniques - Possibility of extending system capabilities beyond initial scope - Potential for publication or presentation of research results

**Threats:** - Technology dependency risks and library availability - Schedule delays due to technical complexity - Integration challenges with existing systems

## (P.7) Requirements process and report

The following subsections describe the key stakeholders involved in the project, the planned methods for eliciting their requirements, and the rationale behind these choices.

**Researchers and Graduate Students (Primary Users)**:

Interview Type: Structured Interviews

Justification: As the primary users who will directly interact with the voxel magnetization system, researchers and graduate students possess the most valuable insights into current workflow challenges and desired system functionality. Structured interviews allow us to systematically gather information about their specific needs, pain points, and expectations.

Most Important Question: "What are the most time-consuming aspects of your current voxel property assignment workflow, and how would an automated system need to function to significantly improve your productivity?"

Relevance: Identifies core functional requirements and performance criteria that will drive system design and development priorities.

**Research Assistants and Undergraduate Users (Secondary Users)**:

Interview Type: Focus Groups and Observation Sessions

Justification: Research assistants and undergraduate users often have different interaction patterns and technical comfort levels compared to graduate researchers. Focus groups allow us to understand how less experienced users would approach the system, while observation sessions reveal usability challenges that might not be expressed in interviews.

Most Important Question: "How would you prefer to learn and use a new voxel property assignment tool, and what interface elements would make the system most intuitive for you?"

Relevance: Informs usability requirements and interface design decisions to ensure the system is accessible to users with varying technical expertise.

**Supervisors and Research Leads (Decision Makers)**:

Interview Type: Stakeholder Consultation Meetings

Justification: Supervisors and research leads provide strategic perspective on how the system should integrate with broader research goals and laboratory workflows. Their input is crucial for ensuring the system delivers value at the organizational level and aligns with long-term research objectives.

Most Important Question: "How should the voxel property assignment system integrate with existing laboratory workflows to maximize research productivity and minimize disruption?"

Relevance: Ensures the system meets organizational needs and integrates effectively with existing research infrastructure and processes.

**3D Printer Technicians and Operators (Integration Stakeholders)**:

Interview Type: Technical Interviews

Justification: Printer technicians and operators have direct experience with the existing 3D printing workflow and understand the technical constraints and requirements for successful integration. Their input is essential for defining the export format and integration specifications.

Most Important Question: "What specific data format and workflow requirements must the voxel property assignment system meet to integrate seamlessly with your existing 3D printer operations?"

Relevance: Defines technical constraints and integration requirements that are essential for system success and user adoption.

# References


- [1] Bertrand Meyer. *Handbook of Requirements and Business Analysis*. Springer. 2022.
- [2] Ian Sommerville and Peter Sawyer. *Requirements Engineering: A good Practice Guide*. Wiley. 1997.
- [3] *COMSOL Multiphysics Simulation Software*. COMSOL. 2026.
- [4] Brittanica Editors. *magnetic field - definition*. Brittanica. 2025.
- [5] Marshall Burns. *The StL Format - Automated Fabrication, Sec. 6.5*. Ennex Research Corp. 1999.
- [6] Shawn Lawton Henry (Editor). *WCAG 2 Overview*. Web Accessibility Initiative. 2025.

# Appendices

## (RF) Reflections

**1. What went well while writing this deliverable?**

Overall, the deliverable progressed smoothly after the development of the problem statement and goals. The peer feedback we received kept us on our toes, ensuring we remained focused on delivering a document that not only followed the rubric but captured our client's needs succintly. Our plan was straightforward; create an initial draft of the SRS, verify any assumptions made during a client meeting, verify our writing style with the TA, and make necessary corrections. This plan was executed successfully, with the client meeting being extremely useful and resulting in numerous changes to project scope (the most prominent addition being material assignments to voxels). Despite the tense deadlines, the group succeeded in completing their pieces with time for review.

**2. What pain points did you experience during this deliverable, and how did you resolve them?**

*Omar*

While working on this deliverable, one of the main challenges I faced was ensuring consistency and logical flow across the G sections; especially between the Expected Benefits, Functionality Overview, and Stakeholders sections. Because multiple sections were written in parallel, some phrasing and terminology drifted apart, which created minor inconsistencies in how goals and benefits were described. I also found the goal model diagram difficult to create, as it required translating abstract project goals into a clear and meaningful visual representation that aligned with the written content. Balancing clarity with technical accuracy took several iterations. To resolve these issues, I revisited earlier drafts, created a mapping between benefits and system functions, and worked closely with teammates to refine both the textual content and the diagram so that they complemented each other effectively.

*Daniel*

An issue I personally encountered was challenges with ensuring document consistency in naming, numbering, style, and how concepts are explained. In order to balance our individual workloads and work on the document, we split up work and went our own ways. This was fine; however it led to inevitable inconsistencies between sections. As the primary reviewer of all sections, I was in charge of catching these. Deciding upon standards was tricky at times, especially when one section references another (e.g. a scenario implying the existence of functional requirements we didn't have). Client meetings were able to clear up most serious disagreements, leaving primarily naming and numbering errors, which were easier to deal with.

*Andrew*

I encountered challenges with maintaining consistency across different document sections, particularly when working on the stakeholder groups and requirements prioritization. As sections evolved and requirements were refined, I found myself having to repeatedly update cross-references and ensure that component descriptions remained aligned with the functional requirements. The most significant challenge was balancing the level of technical detail in the system architecture section - too much detail would exceed the requirements scope, while too little would make it difficult for stakeholders to understand the system structure. I resolved this by focusing on component responsibilities and interfaces rather than implementation details, and by establishing clear naming conventions that were consistently applied across all sections. Regular team reviews and feedback sessions helped identify and correct inconsistencies early in the process.

*Olivia*

Prior to creating the requirements section of the document, we did have meetings with stakeholders so they could share their experience and the challenges they currently face. However, I still struggled with creating requirements that would sufficiently cover user needs that were not explicitly stated but would be essential. I found that without first-hand experience, intuitive understanding of the unspoken user needs was limited. To address this issue, I made a first draft of requirements that incorporated assumed user needs. During a subsequent meeting with Dr. Onaizah, I was able to get feedback based on her first-hand experience in direct relation to the scope of each requirement. This provided insight on the value and impact of each requirement, which helped guide future revisions. It definitely strengthened my confidence that the final requirements list is comprehensive and reflective of what the system requires.

*Khalid*:

The challenge that I faced when working with this deliverable is keeping the consistency between the requirements and the detailed usage scenarios. This was particularly difficult because the requirements were frequently evolving, which meant I had to constantly update the usage scenarios to reflect the latest changes. Another challenge was the misunderstanding between the group and the supervisor, as we missed an important requirement of adding and removing voxels, which was cleared up during the meetings.

**3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?**

A large majority of requirements were directly from our talks with Dr. Onaizah, with additional requirements from our own ideas of what may facilitate their work or standard editing software. For instance, the ability to edit each layer individually became viewing it separately; editing voxel properties individually became the panels to edit these properties of selected voxels. Standard editing requirements included undo/redo functionality (not explicitly stated by stakeholders, but a reasonable assumption), the ability to save as/autosave, etc. Usability requirements surfaced from both difficulties discussed about their current solution, as well as our own preferences when editing files (another reasonable set of preferences, given we have all done similar technical work). Ultimately, when requirements came from our conceptions of what the product should be like, cross-checking with our client was top priority; we still retain plans to verify the finer details of this document with Dr. Onaizah as part of our verification plan.

**4. Which of the courses you have taken, or are currently taken, will help your team to be successful with your capstone project?**

*Omar*

- **SFWRENG 3RA3 -** This course directly supported the creation of the Goals and System sections by providing structured methods for eliciting, documenting, and validating requirements. The traceability and goal modelling concepts from this course were especially useful when linking stakeholder needs to system functions.

- **SFWRENG 4HC3 -** The principles of human-computer interaction helped me think critically about usability and stakeholder experience while defining expected benefits and system objectives.
- **SFWRENG 3A04 -** Concepts of large-scale software design aided in structuring our system architecture and ensuring maintainability between interconnected components.
- **SFWRENG 3S03 -** Knowledge from this course guided our approach to defining verification and validation criteria, ensuring that the requirements could be practically tested later in development.

*Daniel*

- **SFWRENG 3A04 -** Large scale system design knowledge will come in handy when developing the overall structure for both the POC and revisions. Code maintainability will be absolutely essential both while working on the project ourselves, and if the project is to be maintained/expanded upon in the future by other people.
- **SFWRENG 3DB3 -** Skills in SQL, as well as database management, will be useful when developing the user edit history module.
- **SFWRENG 4X03 -** When working on the 3D rendering modules, knowledge of scientific computation (both how to accomplish things efficiently and avoiding error) will be useful, given the large amount of computation required.
- **SFWRENG 2AA4 -** Code design on smaller scales will further ensure code written is maintainable and logical to future people who may work on the project.

*Andrew*

- **SFWRENG 3RA3 -** This course provided the foundational knowledge for requirements engineering and system specification, which directly informed my work on the system architecture and component definitions. Understanding how to properly structure requirements and trace them to system components was crucial for maintaining consistency across the document.
- **SFWRENG 3A04 -** Large-scale system design principles from this course helped me understand how to properly decompose the voxel magnetization system into manageable components and define clear interfaces between them. This knowledge was essential for creating a maintainable architecture.
- **SFWRENG 4HC3 -** Human-computer interaction principles from this course informed my understanding of how to design intuitive user interfaces for 3D model manipulation. Concepts like feedback, constraints, and mappings will be crucial for creating an effective voxel selection and property assignment interface.

*Olivia*

- **SFWRENG 3RA3:** This course provided foundational knowledge and theoretical practice in developing an SRS document. That experience served as a significant guide, majorly influencing the development of our current SRS document.
- **SFWRENG 3DB3:** Due to the large amounts of expected data, the significant time complexity of data manipulation exceeds the capabilities of most data structures. Therefore, we will likely have to integrate a database to support any operation that accesses a specific voxel and its related metadata. The teachings from this course allow us to understand how databases work to effectively use them

within our system.

- **SFWRENG 3BB4 & SFWRENG 3SH3:** The teachings from these two courses may help optimize implementation methods during import and export processes. Between both courses, I feel that I have built a fairly comprehensive understanding of threads, concurrency, synchronization and resource handling. I believe that this information could be used to improve the performance of our import and export process by effectively leveraging the capabilities of parallel computation.
- **SFWRENG 3S03:** It is essential that the system we are building is robust and well-tested, as it is being developed for actual implementation and usage after the Capstone is complete. The teachings from this course will help guide the development of a comprehensive testing plan that can build confidence in the reliability of our system when we are no longer around to help fix any bugs.

*Khalid*

- **SFWRENG 3A04 -** This is a large system design course, that helped a lot when working on the overall system architecture and ensuring the project's maintainability.
- **SFWRENG 3RA3 -** This is a software requirements and safety considerations course, this helped me a lot when I was creating the detailed usage scenarios, as it provided a structured approach to defining user needs and system behaviors.
- **SFWRENG 4HC3 -** This is a human computer interfaces course, this helped a lot when thinking about how the system interface would look like, what kind of considerations that need to be taken like Feedback, Mappings, and Constraints, which are essential for an intuitive user experience.

**5. What knowledge and skills will the team collectively need to acquire to succesfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge, or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.**

There are five major knowledges/skills our group will need:

- *Knowledge of 3D rendering technologies,* notably how to ensure performance is acceptable. This will be intrinsically linked to which library is utilised.
- *Knowledge of the custom 3D printer software.* While we will not be working with it directly, knowledge of how it expects to take input and how it can fail will be extremely useful assets when developing the export modules.
- *Developing intuitive user interfaces.* Despite there being many examples of good interfaces we can use as reference or inspiration, this skill will still be necessary both when creating the more custom parts of the interface, and when discussing with stakeholders what would work best for them (e.g. drawing out tacit knowledge).
- *Conflict resolution.* Over the coming seven months, it is highly unlikely no conflicts between team members will arise, even minor ones. Being able to facilitate tough conversations and the ability to reduce tension and/or the stakes of a situation is important to maintaining group morale during stressful times.
- *Knowledge of STL file specifications.* This arises specifically from the constraint on input files to the system; knowledge of their format, how to validate and modify them will be essential.

**6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?**

*Knowledge of 3D rendering technologies approaches*

- **Online tutorials and documentation for specific libraries/frameworks** This includes looking at the official documentation of the popular 3D rendering libraries, and understanding how they work (Three.js). This also includes using online tutorials to better understand how the libraries are used in a real-world example. The goal is to understand how these libraries deal with 3D images and how they can be used in this system.
- **Experimentation with existing 3D modeling software** By creating a simple 3D project, the team can gain hands-on experience with how 3D models are created, manipulated, and rendered. This provides practical insight into both the user-facing aspects and the underlying principles of 3D graphics.

*Knowledge of the custom 3D printer software approaches*

- **Reviewing existing documentation** Obtaining and reviewing any documentation provided by the supervisor for their custom 3D printer is crucial to our system. This will offer a great idea on how our system will interact with the Java program to print the model, and it will help us better understand what our system's output should be.
- **Interviews with the supervisor** Due to it being a custom 3D printer, a way to understand the it is having scheduled meetings with the supervisor to better know the software's operational details, including its input requirements, common failure modes, and any specific data formats it utilizes.

*Developing intuitive user interface approaches*

- **Utilizing Human Computer Interfaces principles** Our team is actively enrolled in a dedicated course on human-centered design, which provides a structured and collaborative environment for developing this crucial skill through a comprehensive project. The course will give us a better understanding on what a good design looks like and how we can implement it in our system.
- **Online research on best style/practices** There is endless information available online related to developing intuitive, human-centered designs. This type of learning is something we are intimately familiar with from work on both personal projects and to catch up when a course is lacking.

*Conflict resolution approaches*

- **Active-listening–based problem solving:** This approaches conflict resolution with a focus on effective, open communication. It gives each person a chance to explain their perspective while all remaining parties give their full, undivided attention. While listening to someone else's perspective, the goal is to understand where the other person is coming from, even if you still disagree. This approach can help foster trust and strengthen group dynamics by ensuring all group members feel valued and heard when determining a solution to the conflict.
- **Integrated mediation:** This approach integrates a neutral party to help facilitate effective conversation between the two parties that disagree. It is still the responsibility of the two parties to come together with a final decision they both agree on. The mediator is not responsible for making a final decision that ends the disagreement. Rather, the mediator can help defuse tension and keep the conversation productive, ensuring both parties are able to interact with each other in an equitable

and respectful manner. By introducing a mediator, this helps prevent misunderstandings or an imbalance in power.

*Knowledge of STL file specifications*

- **Online research and documentation review.** The STL file format is well-documented, there are numerous of online resources and tutorials that details the structure of the file. Most famously **Adobe**, has a well documented page explaining STL file format and how to create one. This approach allows us to understand the STL file specifications theoretically.
- **Practical implementation through parsing and validation.** Working with existing libraries to write a basic parser for STL files will provide us with hands-on experience, this will help us understand the structure even more and how we can deal with it practically. This will involve reading, interpreting, and validating the data within STL files.

From the identified approaches, these are which each team member will pursue and why they made their choices:

**Omar**

For this project, I want to strengthen both my technical and collaborative skills to better support the team's progress. My main goal is to build a stronger understanding of 3D rendering by exploring online tutorials and experimenting with libraries like **Three.js** to see how voxel visualization can be done efficiently. I also plan to learn more about the custom 3D printer software by reviewing its documentation and speaking directly with our supervisor to ensure our export formats align with their workflow. In parallel, I'll continue improving my approach to designing intuitive interfaces by applying concepts from our **4HC3** course and studying how other 3D tools handle complex visual data. On the teamwork side, I'll focus on practicing active listening to keep communication open and resolve conflicts early. Lastly, I'll deepen my understanding of STL file specifications by reading technical references and working hands-on with sample files to better connect our theoretical design with its practical implementation.

**Daniel**

*Knowledge of 3D rendering technologies:* I plan to follow the first approach as it most closely aligns with the style of learning I am comfortable with; while creating a scaled-down 3D rendering program would be quite helpful before tackling this larger project, fitting this extra step into my schedule would be quite difficult and likely infeasible.

*Knowledge of custom 3D printer software:* I will likely work with both of these approaches, as relying purely on documentation when an expert is available won't paint the entire picture. In my opinion for a specific knowledge such as this, using the documentation as a reference whilst asking any specific questions to the supervisor would be best.

*Developing intuitive user inferfaces:* I likely will pursue a mix of these approaches given what the project ends up warranting. Online research will supplement any knowledge the course does not provide (e.g. specific guidelines). The timeline of course completion aligns perfectly with when UI will likely be developed, so a majority of skill development will lean on the course.

*Conflict resolution:* I find myself gravitating towards the first approach, familiar with it from previous group conflicts both within work and in personal contexts. I already try to see other people's points of view in everyday life, so this approach is natural to me.

*Knowledge of STL file specifications:* I plan to pursue mostly the second approach, consulting documentation when issues are encountered. With STL files being perceived as a much smaller scale knowledge base versus 3D rendering, creating a basic parser is much more manageable.

**Andrew**

*Knowledge of 3D rendering technologies:* I will pursue the documentation and framework approach first, as understanding the theoretical foundations and API capabilities of 3D rendering libraries will be crucial for making informed architectural decisions. Given the complexity of 3D visualization, I want to ensure I have a solid understanding of performance considerations and rendering pipelines before attempting hands-on experimentation.

*Knowledge of custom 3D printer software:* I will focus on interviews with Dr. Onaizah, as this custom software has limited documentation and the supervisor's first-hand experience will provide the most accurate and up-to-date information about input requirements, failure modes, and integration considerations. This approach will also allow for immediate clarification of any technical questions that arise during development.

*Developing intuitive user interfaces:* I will leverage the 4HC3 course material as my primary learning source, as it provides structured, evidence-based principles for interface design. The course's focus on user-centered design aligns well with our stakeholder-driven approach, and the theoretical foundation will inform practical design decisions throughout the project.

*Conflict resolution:* I will pursue active problem solving in real settings, as it aligns with my natural communication style and promotes collaborative decision-making. This approach ensures all team members feel heard and valued, which is essential for maintaining team cohesion during challenging periods of the project.

*Knowledge of STL file specifications:* I will start with online research and documentation review to build theoretical understanding, then move to practical implementation through parsing and validation. This two-stage approach ensures I understand both the file format specifications and the practical challenges of working with STL data in our system context.

**Olivia**

*Knowledge of 3D rendering technologies*: I will pursue documentation for libraries and frameworks because I believe that will be more helpful for learning how to execute 3D rendering. I anticipate that it will be difficult to create a system that supports these visuals within our UI/UX, as it is likely very technical in nature. Learning more about potential libraries and frameworks can allow me to properly leverage their capabilities to tackle this complex problem. I think later experimentation with existing 3D modeling software is good to understand how a UI/UX design effectively supports user interaction with a 3D model.

*Knowledge of STL file specifications*: I will initially pursue online research and documentation review to build the necessary knowledge, as I have no background in working with STL files. This means that to even attempt any practical implementation, I need to build a foundation of theoretical knowledge, which will require online resources. Once I build that theoretical knowledge, I can then leverage hands-on practice with STL file modifications to verify correct understanding.

*Knowledge of the custom 3D printer software*: I will be pursuing interviews with our supervisor, Dr Onaizah. The existing documentation will likely be unnecessarily complex and technical in nature, making it difficult to understand. Dr Onaizah will be able to provide the information we require much more intuitively while avoiding technical details that are irrelevant within our project scope. It is also easier to clarify details or confusion, as you're able to receive immediate feedback during a conversation.

*Developing intuitive user interfaces*: I will be pursuing the avenue of learning through coursework, as it offers a more thorough understanding of principles and considerations that can be accidentally overlooked during online research. This is especially relevant when you don't know where to start when looking at online resources. Therefore, emphasizing the integration of course teachings ensures there is a solid design foundation. From there, online resources can help fill any gaps in knowledge that remain.

*Conflict resolution*: I will be pursuing active listening, as I believe it is ideal based on current group dynamics. Even when we disagree, all group members have been respectful and civil. Therefore, if we all continue to make the conscious effort to show respect and attentiveness when others share their opinions, I feel that our group is more than capable of having an effective conversation to determine a solution.

**Khalid**

*Knowledge of 3D rendering technologies*: For this skill, I will use the online tutorials and documentation for specific libraries/frameworks because it allows me to follow a structured and self-paced learning experience while also focusing on the fundamental concepts of rendering 3D visualizations, gaining a solid theoretical foundation on how it can be implemented for our system.

*Knowledge of the custom 3D printer software*: For this skill, I will take advantage of the interviews with the supervisor because they know best of the custom 3D printer software. This will give me the most up to date and specific information regarding the functionalities of the software. Also, it gives me someone that I can always ask questions to get clarifications from.

*Developing intuitive user interfaces*: For this skill, I will utilize the information and knowledge that I earn from our 4HC3 course. This is because so far the course structure and layout has been very clear and I have been learning a lot, there is also lectures that I can look back on in case I missed anything that can be used when designing this system.

*Conflict resolution*: For this skill, I will pursue the active listening based problem solving because it fouceses on empowering team members to resolve conflicts directly and constructively. By practicing active listening, I can ensure that all team members feel heard and understand and that is a critical first step in de-escalating tension and finding common ground.

*Knowledge of STL file specifications*:I will primarily use the practical implementation approach. While theoretical knowledge is important, the system's requirement of validating and modifying STL files is a necessity, and any practical understanding can only be gained from hands-on experience working with the STL files. However, I will use online research and documentation when needed as a reference for specific details and edge cases encountered during implementation.