AutoVox

Requirements Standard Plan

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

Version 1, 2025-10-06

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.

CC BY SA

Control Information

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

CC BY SA

(G) Goals

(G.1) Context and Overall Objectives

At McMaster University, the Engineering Research 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 magnetize voxels 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. 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

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. This solution reduces design time and supports the department's research in multi-material 3D printing.

For researchers and students, the tool will streamline workflows by allowing faster iteration and enabling edits or re-magnetization of voxel properties without rebuilding models from scratch.

For supervisors and the university, the system offers a more user-friendly and productive platform, allowing larger CAD models to be automatically voxelized and then edited efficiently. This makes complex designs more manageable and accelerates experimental research.

The following goal model diagram illustrates how these benefits map to the project's stakeholders. image::models/G3_uml.svg[scale=70%,align="center"]

(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:

- **CAD-to-voxel conversion.** The system shall allow importing standard CAD models and automatically

CC BY SA

voxelizing them into a structured grid for further editing.

- **Voxel editing and re-magnetization.** The system shall enable assigning and re-assigning material and magnetization properties at the voxel level, allowing modifications after the initial design without rebuilding the entire model.
- **Batch voxel operations.** The system shall support batch editing and magnetization of groups of voxels in specified directions, reducing repetitive manual work.
- **Visualization tools.** The system shall provide visualization of material distribution and magnetization vectors across the voxel grid, assisting researchers in verification before fabrication.
- **Export compatibility.** The system shall export voxelized designs into formats compatible with multi-material 3D printers, embedding metadata about material properties and magnetization.

Of the functional requirements listed above, the two most critical are **CAD-to-voxel conversion** and **voxel editing and re-magnetization**, as these form the foundation of the system's usability and research value.

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

- **Usability.** The platform should have an intuitive interface suitable for both researchers and students, minimizing training overhead.
- **Performance.** The system should handle CAD-to-voxel conversion and property assignment in a reasonable time, even for large and complex models.
- **Compatibility.** The platform should support common CAD file formats and integrate smoothly with existing 3D printing workflows.

(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, take a look at the S.4.

CC BY SA

Figure 1. High Level use cases diagram

The use case diagram 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 3D 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 voxel grid with fixed dimensions.
  4. System displays the voxelized model ready for magnetization 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, sliced into layers from bottom to top.
3. User navigates through layers to view the voxel structure.
4. User selects individual voxels or multiple voxels within a layer.
5. System highlights selected voxels for material and/or magnetization vector assignment.
6. User can modify selections across different layers.

- **UC3: Assign Properties to Voxels.**
  1. User opens the system and selects an existing project file with the intention of assigning material IDs and magnetization vectors to voxels.
  2. User selects voxels requiring a specific set of properties assigned to them.
  3. User specifies the magnetization direction for the selected voxels, which the system applies.
  4. System provides visual feedback indicating the voxels have been magnetized.
  5. User specifies the material ID for the selected voxels, which the system applies.
  6. System provides visual feedback indicating the voxels assigned material ID.
  7. System maintains editing history, providing user the option to undo/redo.
  8. User completes property assignment for all required voxels.
- **UC4: Add and Delete Voxels.**
  1. User opens the system and selects an existing project file with the intention of adding and deleting voxels.
  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.
  6. User completes all needed voxel additions and deletions.
- **UC5: Export Property assignment Data.**
  1. User completes property assignment for the entire model.
  2. User initiates export process from the system.
  3. System validates that all voxels have both a material and a magnetization vector assigned.
  4. System generates a file containing per-voxel data, including voxel location, magnetization vector, and material ID, ensuring compatibility with custom printer software.
  5. User receives the exported file for printer pipeline integration.

(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.

CC BY SA

(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.

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.

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).

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.

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.

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.

CAD (Computer Aided Design) Software

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

CAD File

A file containing a 3D model, generated by CAD 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.

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.

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.

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).

Model Slicing

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.

(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.** 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.** 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.
- **Export File Support.** 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.** 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:** 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:** We assume that the lab computers possess at least the resources required to handle rudimentary 3D graphics.
- **Browser Support:** We assume that the system will only support Chromium browsers. This will capture a majority of the most popular browsers in use.
- **Language Support:** 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.

CC BY SA

(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

Overall structure expressed by the list of major software and, if applicable, hardware parts. [1]

- **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 3D CAD..
- **Visualization Manager:** This component is in charge of handling all tasks associated with recreating a 3D rendition of the CAD design 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 MagData Platform, which is a desktop application. Users will be required to directly install the software on their laptop, as the MagData Platform will be a locally deployed software system.

CC BY SA

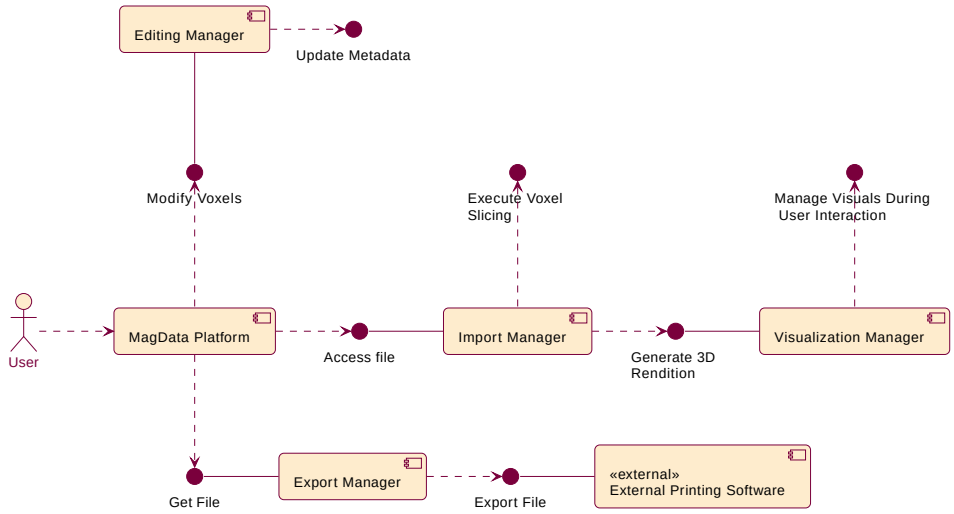

Figure 2. Components involved in MagData platform

(S.2) Functionality & Prioritization

***This is the bulk of the System book, describing elements of functionality (behaviors)**. This chapter corresponds to the traditional view of requirements as defining "**what the system does**". It is organized as one section, S.2.n, for each of the components identified in S.1, describing the corresponding behaviors (functional and non-functional properties).* [1]

The following section outlines the specific functional and non-functional requirements, where each requirement has been defined within the scope of a specified component. 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.

(2.2.1) Import Manager

Functional Requirements

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

   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:** Import Manager shall allow a user to import a past project file in order to reopen a project with all magnetization and material properties preserved. (F212)

   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 because it is the simplest method to streamline future edits to past models.

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

   Priority: Low (could-have)
   Likelihood of change: Medium
   Justification: It's not required by the user for current usage but 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:** 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. (F214)

   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:** Import Manager shall resolve partial voxels and interpret them in the best way that preserves the integrity and accuracy of the provided model. (F215)

   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:** Import Manager shall partition the model into user-manageable display sections that do not surpass the MAX_DISPLAY threshold. (F216)

   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 visuals.

Non-Functional Requirements

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

   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:** Import Manager shall generate the voxel data file at a minimum rate of MIN_RATE. (NF212)

   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.

(2.2.2) Visualization Manager

Functional Requirements

1. **3D image rendition:** Visualization Manager shall recreate a 3D model with clear visualization of sliced voxels. (F221)

   Priority: High (must-have)
   Likelihood of change: Low
   Justification: It provides vital visualization of the model, which is a critical guide for property assignment. Due to its critical nature, there's limited flexibility for change.

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

   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:** Visualization Manager shall provide users with an intuitive interface that permits seamless navigation across multiple perspectives of the 3D model. (F223)

   Priority: High (must-have)
   Likelihood of change: Medium
   Justification: It provides users with a quality experience by ensuring complete and extensive visualization. Variations may occur depending on the perspectives required to satisfy the user's visualization needs.

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

   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:** Visualization Manager shall provide users with visualization that showcases which voxels are currently selected within a specific layer. (F225)

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

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

   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:** Visualization Manager shall update any changes to the perspective of the 3D model with a latency of less than MAX_VIEW_LATENCY to allow a seamless user interaction. (NF221)

   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:** Visualization Manager shall support the visual display of voxel models that contain up to MAX_VOXELS without significant degradation of performance or loss of functionality. (NF222)

   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:** Visualization Manager shall ensure all colours that highlight voxels have unique hex codes and can be distinctively recognized under standard lighting conditions. (NF223)

   Priority: Medium (should-have)
   Likelihood of change: Low
   Justification: Its priority is linked to any tracker requirement. Once visual tracking is implemented, its priority is elevated to support distinct visualization of desired features for an improved user experience. It contains critical standards for easy visualization which means there's limited flexibility for change.

(2.2.3) Editing Manager

Functional Requirements

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

   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:** Editing Manager shall allow users to define and maintain a list of 'favourite' magnetization vectors for quick selection and easy reuse. (F232)

   Priority: Low (could-have)
   Likelihood of change: Low
   Justification: It improves user satisfaction by allowing quicker magnetization, 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:** Editing Manager shall allow users to select an individual or group of voxels and assign a specific material to the them. (F233)

   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:** Editing Manager shall ease the process of recalling what material number correlates to a chosen material by providing the ability to assign a label to a material number. (F234)

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

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

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

   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** Editing Manager shall allow users to select an entire layer at once to enable easy

assignment of a common material and magnetization amongst all voxels within a layer. (F238)

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

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

    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:** Editing Manager shall ensure that the process of a material or magnetization property to a large group of selected voxels can be completed in MAX_INTERACTIONS or less by a user who is familiar with how the printer operates, creating an intuitive interface. (NF231)

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

   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:** Export Manager shall validate that all voxels have received an assigned magnetization value upon receiving a request to export a file for printing (even if the assigned value is simply null to indicate no magnetization required). (F241)

   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:** Export Manager shall produce a standalone file, containing all metadata for each voxel, that the user can name and save locally on their personal device outside of the software. (F242)

   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:** Export Manager shall allow the user to export their model manually in the native format, that the user can name and save locally on their personal device. (F243)

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

   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:** Export Manager shall allow users the option to export a file that summarizes model statistics upon producing a standalone file of the model. (F245)

   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.

CC BY SA

Non-Functional Requirements

1. **Fail safe:** Export Manager shall ensure voxel data and metadata remain intact and unaltered in the event of an incomplete export, preserving the integrity of the project file without loss. (NF241)

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

   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.

(S.3) Interfaces

How the system makes the functionality of S.2 available to the rest of the world, particularly user interfaces and program interfaces (APIs). It specifies how that functionality will be made available to the rest of the world, including people (users) and other systems. These are interfaces provided by the system to the outside; the other way around, interfaces from other systems, which the system may use, are specified in E.2. [1]

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.

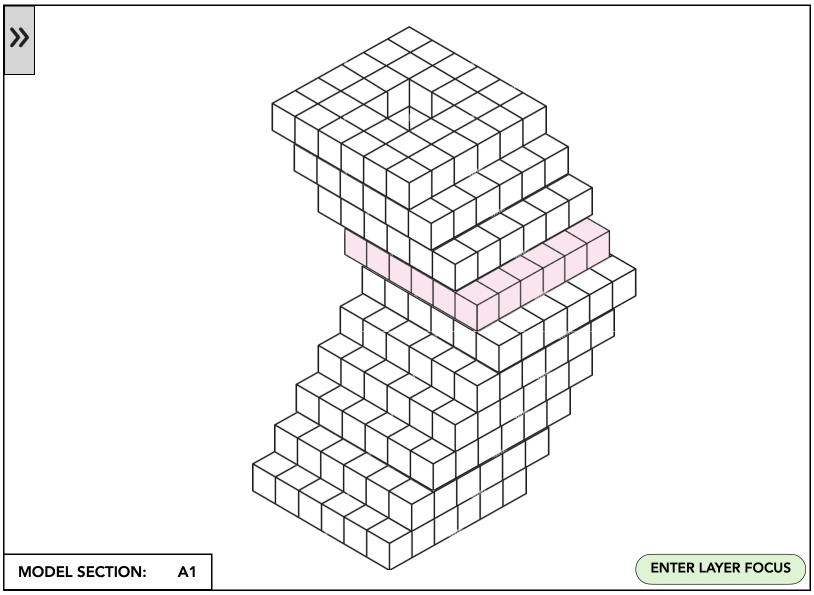

Figure 3. Interface after initialize 3D rendition is generated.

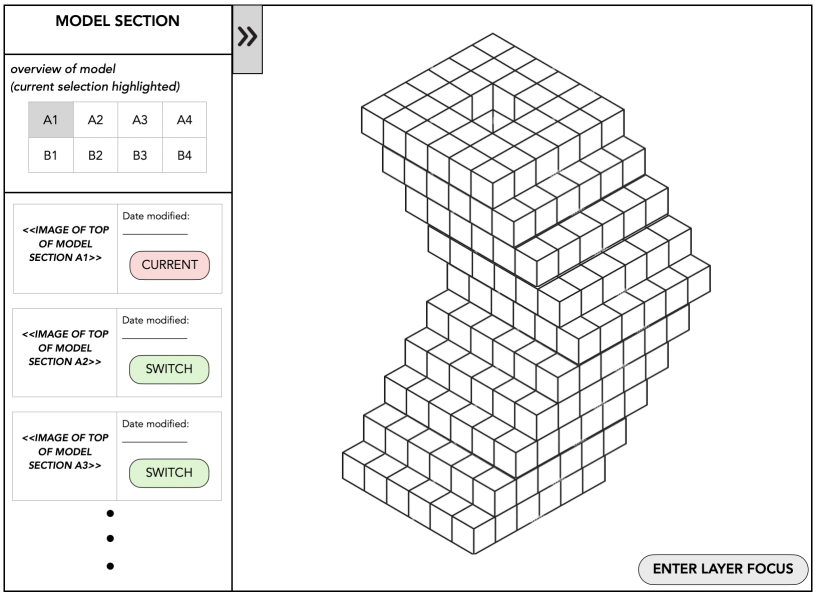

Figure 4. Accessing partition division through interface.

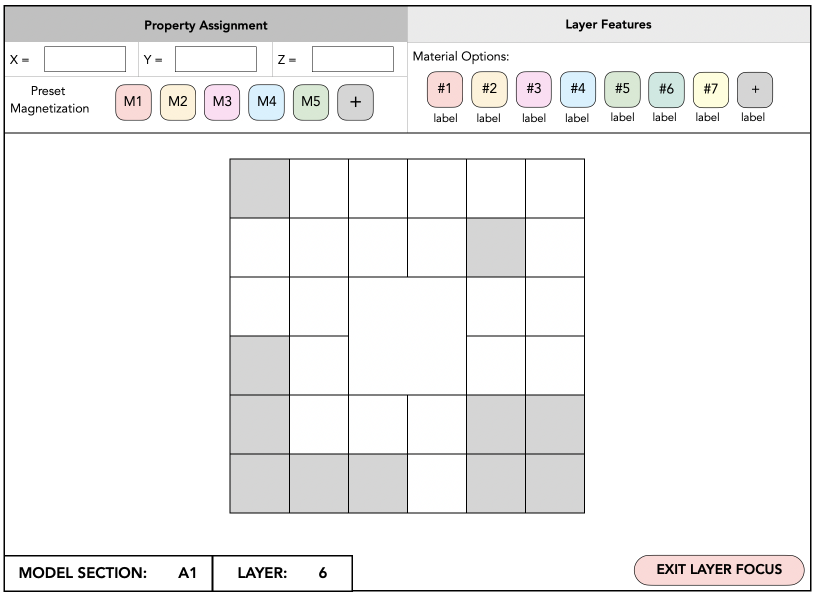

Figure 5. Entering layer focus interface for property assignment.

(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
- **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
- **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
- **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
- **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
- **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
- **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 pattern and ensure it meets the their needs.
- **Main Success Scenario**:
  - 1. System displays the complete 3D model with magnetization 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.
- **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
- **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.

(S.5) Verification and acceptance criteria

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 clairty, minimal steps to complete the core tasks, and accessibilituy of commonly used features. For example, the Property Manager should enable group edits in MAX_INTERACTIONS or fewer.
  - Covered requirement [NF223], [NF231]
- **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]

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]

Traceability Matrix

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	Test Description	Link to Requirements
Test 5	Validate undo and clear operations	[F237], [F2310]
Test 6	Validate auto-save and data persistence	[F236]

(P) Project

(P.1) Roles and personnel

- 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.

CC BY SA

(P.2) Imposed technical choices

- 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

🔄 Nothing available at this point.

(P.4) Tasks and deliverables

🔄 Nothing available at this point.

(P.5) Required technology elements

- 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

🔄 Nothing available at this point.

(P.7) Requirements process and report

🔄 Nothing available at this point.

CC BY SA

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.

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

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

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

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

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

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

Olivia

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.