Hazard Analysis
Software Engineering

Team #10, Five of a Kind
Omar Abdelhamid
Daniel Maurer
Andrew Bovbel
Olivia Reich
Khalid Farag

Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

Contents

[You are free to modify this template. —SS]

1 Introduction

[You can include your definition of what a hazard is here. —SS]

2 Scope and Purpose of Hazard Analysis

[You should say what **loss** could be incurred because of the hazards. —SS]

3 System Boundaries and Components

As discussed in the SRS, the system consists of four primary modules, each responsible for a specific part of the transformation from the initial CAD file to the completed magnetized voxel file. All components operate locally, with dependencies limited to the library files needed to interact with necessary files.

3.1 Slicing Module

- **Purpose:** Converts input external files into internal native type and interprets existing native files.
- **Key Functions:**
  - Parse CAD files (as per F213)
  - Generate output voxel files for program use (as per F213, NF211)
- **Integration:** Indirectly receives CAD files from external CAD applications.

3.2 Visualization Module

- **Purpose:** Renders interactible 3D images for model visualization.
- **Key Functions:**
  - Interpret voxel files (as per **)
  - Generate an interactible 3D visualization of the voxel model (as per F221, F222)
  - Visualize voxels selected by the user (as per F223)

3.3 Magnetization Module

- **Purpose:** Tracks and stores user edits.
- **Key Functions:**
  - Set magnetization vectors for sets of selected voxels (as per F232)
  - Store per-voxel magnetizations within internal voxel files (as per F232, NF232)

3.4 Exportation Module

- **Purpose:** Handles saving and exporting of user files.
- **Key Functions:**
  - Save voxel models for later editing (as per **)
  - Export internal voxel files as *java-readable* format (as per F243, NF241*)
  - Validate that each voxel has been magnetized (as per **)
- **Integration:** **

4 Critical Assumptions

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS]

5 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

6 Safety and Security Requirements

[Newly discovered requirements. These should also be added to the SRS. (A rationale design process how and why to fake it.) —SS]

7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?
4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?