Team Contributions: POC
Software Engineering

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

This document summarizes the contributions of each team member up to the POC Demo. The time period of interest is the time between the beginning of the term and the POC demo.

1 Demo Plans

For our proof of concept demonstration, we will showcase the core functionality of our voxel magnetization system for 3D printing. The demonstration will focus on the following key aspects:

1. **Import and Voxelization:** We will demonstrate how our system imports CAD files (STL/OBJ format) and converts them into a voxel grid with fixed dimensions, showing the file parsing and voxelization of a simple model.

2. **3D Rendering:** We will demonstrate the 3D rendering of the voxelized model using Three.js, showing how users can view the model from different angles.

3. **Export Functionality:** We will demosntrate how our system exports the voxel data (magnetization, material ID, location) to a CSV file format, demonstrating the file export capability for a simple model. Dummy data will be used for the export.

2 Team Meeting Attendance

Student	Meetings
Total	8
Omar Hassan	7
Daniel Maurer	8
Andrew Bovbel	7
Olivia Reich	8
Khalid Farag	7

We often try to schedule our meetings such that everyone can attend; if people missed, it is due to them not able to attend for a previously discussed reason.

3 Supervisor/Stakeholder Meeting Attendance

Supervisor's Name: [Dr.Onaizah]

Student	Meetings
Total	3
Omar Hassan	3
Daniel Maurer	3
Andrew Bovbel	3
Olivia Reich	3
Khalid Farag	3

Scheduling meetings with the supervisor has been difficult given our schedules and Dr. Onaizah's schedule as well, however we do our best to remain in contact when possible.

4 Lecture Attendance

Student	Lectures
Total	13
Omar Hassan	9
Daniel Maurer	13
Andrew Bovbel	7
Olivia Reich	9
Khalid Farag	8

We aim to have at least one person attending every lecture in order to be most up to date (ideally whoever can attend, attends).

5 TA Document Discussion Attendance

TA's Name: [Chris Schankula]

Student	Lectures
Total	2
Omar Hassan	2
Daniel Maurer	2
Andrew Bovbel	2
Olivia Reich	2
Khalid Farag	2

Our team had fewer meetings with our TA due to his unavailability at the beginning of the term.

6 Commits

Student	Commits	Percent
Total	240	100%
Omar Hassan	16	6.7%
Daniel Maurer	81	33.7%
Andrew Bovbel	49	20.4%
Olivia Reich	35	14.6%
Khalid Farag	59	24.6%

Some members have lower commits due to commit sizes being larger and less frequent (e.g. only commiting once an entire assigned section is complete), and some have more due to commits being smaller and more frequent (this is especially true for hot-fix commits).

7 Issue Tracker

Student	Authored (O+C)	Assigned (C only)
Omar Hassan	0	12
Daniel Maurer	52	12
Andrew Bovbel	0	11
Olivia Reich	0	12
Khalid Farag	3	11

The large discrepancy in issue authorship is due to the assumed roles within our group; assigned much clearly reflects the work divisions.

8 CICD

This section explains the Continuous Integration and Continuous Deployment (CI/CD) plan for our project. The plan will help ensure consistent code quality, testing, and deployment processes.

8.1 Source Control and Branching Strategy

- **Repository Setup**: Our team will use GitHub for version control and collaboration.
- **Branching Strategy**:
  - `main`: The main branch will be used to store the production-ready code.
  - `<name of team member>-<issue name>`: The temporary branches will be created for each team member to work on their assigned issues. These branches will be merged into the main branch after the task is completed.
- **Merging Policy**: All pull requests should have at least two reviews before merging, as outlined in the Development Plan. This will help ensure that the code is of high quality and meets the requirements of the project.

8.2 Testing and Code Analysis Pipeline

Our team will use GitHub Actions for CI/CD to automate testing and code analysis on pull requests. They will include the following:

- **Static Code Analysis & Linting**
  - **Frontend**: `ESLint` will be used to enforce React coding standards and identify potential issues.

  - **Backend**: PyLint will be used to handle both code smells for static analysis and enforce the coding standards for the backend.

- **Testing**
  - **Frontend Unit Tests**: Unit tests will be written using Jest for React components and utility functions, the GitHub Actions will run the tests using the command ‘npm test‘.
  - **Backend Unit Tests**: Unit tests will be written using pytest for API endpoints and core functionality, the GitHub Actions will run the tests using the command ‘pytest‘.
- **Documentation**: LaTeX documentation files will be automatically compiled to PDF on commits to the main branch. The GitHub Actions will run the compilation using the command ‘make all‘.

9 Team Charter Trigger Items

Our team has experienced no violations of the triggers in the team charter.

10 Additional Productivity Metrics

Our team has no other productivity metrics in use.