Our Awesome Project

Requirements Standard Plan

Author 1, Author 2, Author 3

Version 1, 2025-10-11

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.

Author 1

As I submit my work, I, **Author 1**, 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.

Author 2

As I submit my work, I, **Author 2**, 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.

Author 3

As I submit my work, I, **Author 3**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Control Information

Version	Delivery		Feedback	
	Deadline	*Delivered*	*Received*	*Integrated*
V1				
V2				
V3				

Author 1

Here is a quick biography of **Author 1**. You can contact them at **a1@mcmaster.ca**

Author 2

Here is a quick biography of **Author 2**. You can contact them at **a2@mcmaster.ca**

Author 3

Here is a quick biography of **Author 3**. You can contact them at **a3@mcmaster.ca**

Glossary

Vim

Vim is a highly configurable, keyboard-driven text editor built to make creating and changing any kind of text very efficient.[3]

(G) Goals

Control Information

Table 1. Our Awesome Project — Versionning Information — Goal Book

Section	Version	Lead	Delivered on	Reviewer	Approved on
G.1					
G.2					
G.3					
G.4					
G.5					
G.6					
G.7					

(G.1) Context and Overall Objectives

Our Awesome Project is designed to address gaps in existing note-taking systems, which lack efficient methods for integrating text and diagrams through a unified keyboard-driven workflow. Current solutions impose trade-offs in speed, flexibility, and workflow continuity, as explored in G.2.

The goal of Our Awesome Project is to create an intuitive system that allows for keyboard-driven creation and editing of notes involving text and diagrams. Although there is an expectation that there will be a learning curve to become proficient with the system, the intended benefit is that the user will be able to create and edit notes more quickly than with existing systems.

(G.2) Current situation

Our Awesome Project arises from the fact that students, especially in technical disciplines such as engineering, often need to create notes containing both structured text and quick sketches of diagrams like state machines, circuits, or tables. Often times these notes also need to be created in a timely manner, as lectures can move quickly.

There are many existing options for taking notes, each with their own trade-offs:

- Pen and paper
  - Not easily editable or shareable
  - Handwriting may be slower than typing
- Tablets with stylus support (e.g. iPad with Apple Pencil)
  - Can be expensive
  - Handwriting may be slower than typing
- Traditional text editors (e.g. Microsoft Word, Google Docs)

    - Limited support for creating diagrams within the app
- Mouse-driven editors (e.g. Draw.io, OneNote)
    - Can be slow to navigate menus for specific geometry
    - Limited keyboard shortcut support for creating and editing geometry
    - May require exporting and importing images into a separate text editor for note-taking (e.g. Draw.io)
- Text-based diagram languages (e.g. Mermaid, PlantUML)
    - Separate definition from rendering, forcing users to manage an inefficient write-compile-insert cycle
    - Requires integration with another text editor for note-taking text
- Document markup languages (e.g. LaTeX, Typst)
    - Limited to a more traditional page format rather than infinite workspace like OneNote for example
    - Editing underlying text rather than geometry itself

On top of inefficiencies to workflow, an important note is that none of these solutions are fully keyboard-driven. This is a key aspect of the Vim ideology that Our Awesome Project aims to embody by providing a unified system for creation and editing of notes involving both text and diagrams exclusively through the keyboard.

(G.3) Expected Benefits

🔄 Nothing available at this point.

(G.4) Functionality overview

🔄 Nothing available at this point.

(G.5) High-level usage scenarios

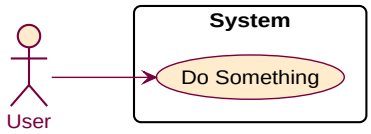

Figure 1. High Level use cases diagram

🔄 Nothing available at this point.

(G.6) Limitations and Exclusions

🔄 Nothing available at this point.

(G.7) Stakeholders and requirements sources

🔄 Nothing available at this point.

(E) Environment

The Environment book describes the application domain and external context, physical or virtual (or a mix), in which the system will operate. [1]

Control Information

Table 2. Our Awesome Project — Versionning Information — Environment Book

Section	Version	Lead	Delivered	Reviewer	Approved
E.1					
E.2					
E.3					
E.4					
E.5					
E.6					

(E.1) Glossary

Nothing available at this point.

(E.2) Components

Nothing available at this point.

(E.3) Constraints

Nothing available at this point.

(E.4) Assumptions

Nothing available at this point.

(E.5) Effects

Nothing available at this point.

(E.6) Invariants

Nothing available at this point.

(S) System

the System book refines the Goal one by focusing on more detailed requirements.

Control Information

Section	Version	Lead	Delivered	Reviewer	Approved
S.1					
S.2					
S.3					
S.4					
S.5					
S.6					

(S.1) Components

Nothing available at this point.

(S.2) Functionality

Nothing available at this point.

(S.3) Interfaces

Nothing available at this point.

(S.4) Detailed usage scenarios

Nothing available at this point.

(S.5) Prioritization

Nothing available at this point.

(S.6) Verification and acceptance criteria

Nothing available at this point.

(P) Project

Control Information

Section	Version	Lead	Delivered	Reviewer	Approved
P.1					
P.2					
P.3					
P.4					
P.5					
P.6					
P.7					

(P.1) Roles and personnel

Nothing available at this point.

(P.2) Imposed technical choices

Nothing available at this point.

(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

Nothing available at this point.

(P.6) Risk and mitigation analysis

Nothing available at this point.

(P.7) Requirements process and report

Requirements Process The requirements elicitation process was conducted in a simplified and collaborative manner, reflecting the project's scope and resources. The team collectively engaged in discussions to identify and document the essential features, goals, and requirements for the project.

The process involved the following steps: * **Brainstorming Sessions**: Initial brainstorming sessions to gather the main ideas and requirements. Each member contributed their perspectives on what the project should achieve. Factors considered included key stakeholders, user needs, technical feasibility, and project risks. * **Informal Discussions**: Ongoing informal discussions to refine and clarify the requirements. This included addressing any ambiguities and ensuring a shared understanding among team members. Finer details, including specific requirements and project scope were discussed and agreed upon. * **Documentation**: The requirements were documented in a collaborative manner, with team members contributing to the creation of this Software Requirements Specification (SRS) document. The document was reviewed and updated as needed to reflect any changes or new insights gained during the process. * **Feedback Integration**: Although formal stakeholder feedback was limited, the team acted as both developers and stakeholders, ensuring that the requirements aligned with the project's goals and constraints. Further feedback from peers and mentors will be sought in future iterations.

For future iterations, a more structured elicitation process may be adopted, including user research, prototyping, and systematic requirements validation to ensure comprehensive coverage of user needs and project goals.

References

- [1] Bertrand Meyer. *Handbook of Requirements and Business Analysis*. Springer. 2022.
- [2] Ian Sommerville and Peter Sawyer. *Requirements Engineering: A good Practice Guide*. Wiley. 1997.
- [3] Vim - the ubiquitous text editor. 2025. https://www.vim.org/