Problem Statement and Goals
ProgName

Team #, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

Table 1: Revision History

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

1 Problem Statement

[You should check your problem statement with the problem statement checklist. —SS]

[You can change the section headings, as long as you include the required information. —SS]

1.1 Problem

1.2 Inputs and Outputs

[Characterize the problem in terms of "high level" inputs and outputs. Use abstraction so that you can avoid details. —SS]

1.2.1 Inputs

- Combinations of keystrokes that:
  - Spell out defined commands/shortcuts
  - Spell out note text for display
- User defined keybinds

1.2.2 Outputs

- Displayed text on the screen.
- Displayed diagrams/geometry on the screen.
- Content (i.e. text and diagrams) on screen that can be saved in a file and re-opened to display the same content

1.3 Stakeholders

Broadly, the stakeholders for this project will all fall into a group of more technically savvy individuals who are likely comfortable using keyboard shortcuts in other familiar software programs. The target users should be able to memorize a combination of shortcuts and commands to align with the keyboard based usage.

- People who use keyboard based workflows and prefer not to use their mouse to take notes that involve simple diagrams.
- People who do not have access to a drawing tablet, but would like to take digital notes involving diagrams in an efficient way.
- People who often take notes on a computer and see value in learning a new system to improve their note taking efficiency.

1.4 Environment

1.4.1 Hardware Environment

The hardware environment will be a personal computer running a modern operating system (i.e. Windows, MacOS, or Linux). The computer will have a keyboard and mouse for user input.

1.4.2 Software Environment

The software environment will be a desktop application.

2 Goals

- Allow users to create, edit, and save documents involving text and diagrams using only their keyboard.
- Allow users to create, edit, and move simple diagrams in a document using only their keyboard.
- Allow users to edit geometry directly (i.e. no text to visual render step).
- Allow users to create simple custom geometry involving lines and predefined shapes (e.g. circle, rectangle, triangle).

- Allow users to save custom geometry as reusable commands for future use.
- Allow users to define custom keyboard shortcuts mapped to commands for diagram creation or insertion.
- Provides the user the potential to take notes involving diagrams more efficiently (time-wise) than other individual/combined note taking applications.
- Cross platform support (Windows, MacOS, Linux).
- Support for Vim style keybindings when editing text.

3 Stretch Goals

- Provides the user the potential to take notes involving diagrams more efficiently than by hand on paper/drawing tablet.
- Ability to export documents as images and/or PDFs.
- Support for collaborative note taking.
- Support for LaTeX math rendering.
- Support for Emacs style keybindings when editing text.
- Support for custom themes.

4 Extras

- Usability testing
- User manual

[For CAS 741: State whether the project is a research project. This designation, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

[For SE Capstone: List your extras. Potential extras include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. (The full list is on the course outline and in Lecture 02.) Normally the number of extras will be two. Approval of the extras will be part of the discussion with the instructor for approving the project. The extras, with the approval (or request) of the instructor, can be modified over the course of the term. —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. Chengze - I came up with many ideas for on stakeholders and goals based on our initial problem statement. And I was able to clearly articulate these ideas in the document.

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

4. Chengze - I found it difficult to limit the number of goals of the project to be within the appropriate number of main goals suggested. I resolved this by discussing with my TA and my teammates, we agreed to focus on the core functionalities of the project and leave some features as stretch goals.

5. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?

6. Chengze - We discussed the goals and stretch goals as a team and prioritized the core functionalities of the project. Instead of focusing on the features/goals that satisfy all the stakeholders, we decided to go with the functionalities that we desire to have in the project. And for those goals that are challenging to implement, we put them as stretch goals so that we can focus on the main goals first, and then work on the stretch goals if we have time.