Development Plan
Flow

Team 9, min-cut
Ethan Patterson
Hussain Muhammed
Jeffrey Doan
Kevin Zhu
Chengze Zhao

Table 1: Revision History

Date	**Developer(s)**	**Change**
22/09/2025	Hussain, Oliver, Jeff, Kevin, Ethan	Initial Release

1 Confidential Information?

The project does not use any confidential information.

2 IP to Protect

There is no IP to protect in this project.

3 Copyright License

The group has agreed that the project should use an MIT License.

4 Team Meeting Plan

The team will meet 1-3 times per week, with one in-person meeting recurring at the same time every week. The meeting will be at a previously agreed-upon location that will be discussed privately. Alternatively, the meeting may be done virtually. We will meet with industry advisers and other experts on a scheduled basis. These meetings will be announced beforehand.

5 Team Communication Plan

The team will use a private Discord server along with GitHub for communication. Team members are expected to check both of these daily for updates to meeting times or other milestones.

6 Team Member Roles

We will need a dedicated Team Lead, Keynoter, Team Liaison, Reviewer, and Meeting chair. The team lead is in charge of scheduling meetings and other events that may be needed to complete the project. The Keynoter is in charge of taking notes on any meetings for later use. The Team Liaison is in charge of communicating with the Professor and other external helpers for the project. The reviewer is in charge of reviewing work to ensure it is of a high standard. The meeting chair is in charge of managing meeting topics and making sure that the meeting stays productive and relevant.

7 Workflow Plan

The project will be using GitHub for version control and issue tracking. It will be set up with a main branch that is always stable, and a dev branch shall be used for integration of new features. Each feature or fix will first be developed in a separate branch off of dev. Once the feature is complete and tested, a pull request shall be created to merge the feature branch into dev. The pull request would be reviewed by at least one other team member before being merged. For smaller changes, the dev branch may just be used directly. The dev branch will be periodically merged into main after thorough testing to ensure stability.

The GitHub Issues will be used to track the various tasks and features that need to be completed, as well as any meetings that are held. Templates are to be used for each issue to ensure relevant information is included. Issues will also be labelled based on their type, such as bug, feature, meeting, etc, where applicable. This will help in organizing and prioritizing the work that needs to be done. Assigning of issues will be done where necessary to track who is responsible for what.

GitHub Actions will be used for CI/CD, mainly for linting and running tests, as well as updating the PDFs from the TeX files. Whenever a pull request is created, the CI/CD pipeline will automatically run the actions, which would include the linter and any tests that have been written. This will help catch any problems early, making sure any code merged is the correct format. Testing will most likely be set up later on as the project progresses and more code is written.

8 Project Decomposition and Scheduling

We will use GitHub projects in the Kanban board format to organize our issue items. To begin, the columns will be ”To Do”, ”In Progress”, and ”Done”. As we work on the project, more columns may be added to better suit our workflow. Each issue will be added to the ”To Do” column, and when a team member starts working on it, it will be moved to ”In Progress.” . Once the issue is completed and reviewed, it will be moved to ”Done”. This will help us keep track of what needs to be done, what is currently being worked on, and what has been completed. The project board can be found at https://github.com/users/ritual-17/projects/1/views/2.

9 Proof of Concept Demonstration Plan

The primary risk for the success of Flow is the feasibility of creating, recognizing and manipulating geometric elements within a graphical user interface, all while only using a keyboard. Generating both basic and custom shapes that can be manipulated and modified through keyboard inputs is a complex task, but this represents the core functionality of the application.

To address this risk, the proof of concept demonstration will focus on verifying that:

- Basic geometric shapes (e.g., rectangles, circles, triangles) can be created using keyboard, textual inputs.
- These shapes can be manipulated within and delete from the GUI. The includes the shapes being selected, moved, and resized using keyboard commands.
- Custom geometry (e.g., user-created/user-defined shapes) can be generated and modified.

The demonstration itself will involve a technically proficient user performing a sequence of note-taking tasks within Flow, such as inputting text, creating shapes, and manipulating them, all using only keyboard inputs. The note to be taken will be complex enough to require the use of multiple geometric elements (e.g. it may include building a finite state machine or something similar). Successful completion of these tasks will confirm the technical feasibility of the core functionality of the application. It would show that users can effectively create and modify geometric elements within a GUI for their notes. Further evaluation of efficiency and intuitiveness can be done once this core functionality is proven.

10 Expected Technology

The programming language we expect to use is Electron with TypeScript. These languages are chosen due to their ability to create cross-platform desktop applications with graphical user interfaces, which are essential for our note-taking application. Some libraries we may use include React for building the user interface components, and others for rendering shapes. GitHub and GitHub Projects will be used for version control and project management, respectively. Testing and linting are important aspects of our development process. Jest and ESLint are potential choices for unit testing and linting, respectively, but these decisions will be finalized as the project progresses. As previously mentioned, GitHub Actions will be used for CI/CD to automate testing and linting processes, as well as updating documentation.

11 Coding Standard

The Standard JavaScript Style Guide will be used as the coding standard for this project. This guide provides a simplistic and easy-to-follow set of rules for writing clean and consistent code. It covers various aspects of coding style, including naming, variables, functions and more. The specific details of the standard can be found at https://www.w3schools.com/js/js_conventions.asp. standard style can also be integrated with ESLint, which may be used for linting the code to ensure it adheres to the coding standard.

Appendix — Reflection

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. Why is it important to create a development plan prior to starting the project?

   - Jeff: A good development plan is important when working on a large-scale project such as this capstone. This lays out the foundation for the rest of the project. From this development plan, we can ensure that all members are on the same page for future deliverables.
   - Ethan: It is important to create a development plan prior to starting the project because it establishes clear expectations, which minimizes miscommunication. This helps to make sure all team members are on the same page, which greatly improves the chance of success and efficiency at the implementation stage. It also requires us to outline future risks, which helps with planning for mitigation.
   - Oliver: A development plan is important since it helps organize the project and break it down into manageable pieces. It also helps identify potential risks and challenges early on, allowing the team to plan for them and mitigate them.
   - Hussain: Creating a development plan is important because it allows all stakeholders of the project to be on the same page, and to have a clear understanding of the goals and objectives of the project. It also ensures that the resources needed for the project is available and allocated properly. A development plan also highlights any problems that you may run into during the project, allowing you to plan around and mitigate them before they arise.
   - Kevin: Having a thought-out development plan is important, as the rest of the project's development will follow the plan. If the plan covers all possible issues, the development will be smooth, as all the issues have been accounted for and will be worked around.

2. In your opinion, what are the advantages and disadvantages of using CI/CD?

- Jeff: The advantages of using CI/CD include:
  (a) Faster Feedback and Dev Cycles
    - Developers get immediate feedback via automated tests and have issues caught earlier in review.

  (b) Improved Code Quality
    - Automated testing, linting, and code analysis help with coding standards.

  (c) Increased Efficiency
    - Automated CI bots help reduce repetitive manual tasks.
- Ethan: The advantages of CI/CD are that it continuously checks that your codebase is in a good state and can automatically run actions for you. This is already the case for the existing LaTeX action, but in the future, we will add automatic testing, linting, etc. It can also be used to automatically handle deployments, which is convenient. One disadvantage is that it is something that you sometimes have to wrestle with with. It can add an extra layer of complexity to your development environment, and therefore it is another thing that can break and add headaches. However, the benefits greatly outweigh the negatives in my opinion.
- Oliver: The advantages of CI/CD are that it allows for faster and more frequent releases, as well as improved collaboration among team members. It also helps catch bugs and issues early on in the development process, leading to higher quality software. The disadvantages include the initial setup and configuration can be time-consuming and complex, and it may require additional resources and infrastructure to maintain.
- Hussain: CI/CD has many advantages. Having an effective CI/CD pipeline set up allows for faster development and deployment of code as it automates many of the tasks that would be done manually. This allows developers to focus on more complex tasks of the project. The disadvantages of CI/CD are that it can be difficult to initially set u,p as developers may not know what is to be automated and what is not. It can also become difficult to fix constantly breaking pipelines if the code is not properly reviewed.
- Kevin: The advantages of CI/CD are that if any issues arise with the project, as it is constantly being deployed, these issues can be found and fixed quickly. This prevents said issues from propagating, causing other issues that make the project bad.

  The disadvantage is that it can require proper planning and is more costly in terms of time and money. This is because each development must be vetted (requires time and money) before development can continue, slowing down project development.

3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

- The group did not have any disagreements for this deliverable. The group did their work smoothly and decided in meetings on elements of this document and the Problem Statement.

Appendix — Team Charter

External Goals

Our team has many external goals for this capstone. It would go without saying that we would like to achieve a high grade for our work in this course; however, there are many other factors as to why we are trying to succeed with this project. First off, as engineering students, we understand the hardships of having to take notes in class and not being able to smoothly incorporate diagrams. As we continue our careers and education, we would like to have an application that allows us to execute exactly what we wished we had in our previous undergraduate years. Additionally, as some of us are graduating and moving on to new career opportunities, it would be a great benefit to us if we had a successful project to give us leverage when applying for new graduate positions.

Attendance

Expectations

Our team expects all members to show up to every scheduled meeting unless they have an acceptable reason for not being able to attend. The specifics on what constitutes an acceptable excuse will be discussed in the next section.

Acceptable Excuse

For in-person meetings, the team understands that some members may have issues commuting and will not be able to physically attend. In this case, members will be allowed to attend online; however, this should not be a regular occurrence, and members should make an effort to attend in person. Moreover, the team believes that there is no acceptable excuse for missing a deadline other than for emergencies. This is further elaborated in the coming sections.

Excuses that involve prioritizing extracurriculars or recreational activities will not be tolerated. Excuses that are a result of one's own poor time management or lack thereof will also not be tolerated. We all understand the busy schedule of an engineering student; however, it is expected that each member manages their time wisely and puts the same amount of effort or more into this capstone as they would into their other courses.

In Case of Emergency

Group members may be met with unexpected issues, which will prevent them from attending a meeting or potentially miss a deadline. For meetings, if a member is not able to attend due to an emergency, they are expected to communicate that and follow up with the team on what they missed. They are expected to also review the meeting minutes before reaching out to the team about what they missed. Moreover, if a member is met with an emergency that prevents them from finishing work for a team deliverable, the team will be understanding and assist to the best of their capabilities. Depending on the situation, the member should follow the proper McMaster procedure for these emergencies and reach out to the TA/Instructor to discuss the potential use of an MSAF.

Accountability and Teamwork

Quality

Team mates should have proper tools for meetings to discuss and work on the project. Code will be reviewed by other team-mates and code of poor quality will be rewritten until it is of proper quality.

Attitude

Each team member is expected to treat all other members with respect and professionalism. Members should feel comfortable sharing ideas and interacting with each other. Members should come into meetings with a positive attitude and a willingness to actively cooperate. Members should also be considerate of each other's time and should stay on track during meetings and work periods.

Additionally, we will be incorporating the already exisitng "Contributor Covenant" as our code of conduct. The exact code of conduct can be found here.

In terms of conflict resolution plan, we decdied to implement a short and decisive process:

1. Direct Communication
    - Members directly involved in the conflict should try to resolve the issue on their own first.
2. Team Discussion
    - If the issue remains unresolved, the team may hold a brief meeting to resolve said conflict. The meeting will be used to discuss the concerns and work towards a solution approved by the majority of the group.
3. Seek Support from Advisor / Instructor
    - If a resolution still cannot be reached after the previous steps, the team should schedule a meeting with the TA or Instructor and ask for guidance on how to resolve the conflict. Note, this will only be used as a last resort after exhausting all options.

Stay on Track

Keeping a team on track is very important when working on a long-term project, such as this capstone. To ensure that the team stays on track, we decided to consider implementing methods such as using project management tools, setting up regular update meetings, and defining clear roles and responsibilities at the beginning of each deliverable. The regular update meetings and assigned responsibilities will help ensure that members contribute their fair share and that the team is performing as expected. Each member will hold the others accountable for their share of the deliverables, which will allow our team to perform as expected.

It is understandable that some members may not feel motivated or may not be performing as expected, even with all the previous methods in place. To prevent this, we will be incorporating fun consequences, such as having these members bring hot chocolate, cookies, or other food items/treats to the next meeting. If the team member continues to underperform despite the above methods are in place, the other members will discuss potentially meeting with a TA or instructor to resolve this issue. Please note that this will only be used as a last resort after exhausting all other options.

Contrarily, if a member does exceptionally well or contributes more than their fair share, we've decided to reward these members by having the rest of the team split their cost at the next team event. For instance, the other members could contribute to the cost of that member's next meal during a team lunch or dinner.

Additionally, if members complete their targets early, they won't be assigned any more work and can focus on their other responsibilities. Incentives for finishing deliverables and tasks are viewed more as a team-building activity. These are discussed in the following section.

Team Building

As engineering students, we understand that there are many stressors when working on a major project, such as a capstone. These stresses can have a big effect on productivity and overall work quality. To combat this issue, we've decided to partake in a group activity or outing for each major goal/milestone we accomplish. For instance, after a stressful deliverable, we'll try to wind down by enjoying online games together. If we accomplish a tough goal together, we think it's a good idea to celebrate with a lunch/dinner to acknowledge our hard work. By doing this, we hope to build team cohesion throughout this project.

Decision Making

We believe that decisions should be made as a group and not by one individual. From this, we decided that we should discuss with all members present when making decisions or handling disagreements. From that discussion, we will hold a vote where the majority rules. We all agree to handle these discussions/meetings professionally and with respect for each other. That way, we can hold a responsible and respected discussion when disagreements arise.