Development Plan
Software Eng

Team #, Team Name
Student 1 Matthew Collard
Student 2 Sam Gorman
Student 3 Ethan Kannampuzha
Student 4 Kieran Gara

Table 1: Revision History

Date	Developer(s)	Change
2023/09/15	Matthew Collard	Team Member roles added
Date2	Name(s)	Description of changes
...	...	...

[Put your introductory blurb here. —SS]

1 Team Meeting Plan

Our Plan is to have a meeting every capstone lecture that is not software eng, or is not running. When we are behind on a deliverable, we will schedule a meeting during our time between classes to meet and go over what we will need to accomplish before the due date.

2 Team Communication Plan

For most issues, we will communicate over Discord with each other. Matthew is in charge of communicating with the profs/supervisor. In addition, we will also have a meeting with Irene Yuan every week during Irene Yuan's Office hours.

3 Team Member Roles

Role	Name	Responsibility
GitHub Administrator	Sam	Merge and maintain Github Branches
Final Revision Editor	Kieran	Last editor of requirements documents, makes sure we adhere to writing guidelines
Communication Director	Matthew	Communicates with the Supervisor/Prof and any stakeholders
Meeting Minute Writer	Ethan	Keeps track and writes down the meeting minutes
Lead Developer	Kieran	Leads the development process, makes sure we are on the right track to complete our goals at the agreed upon due dates
Lead UI Designer	Matthew	Makes sure the user interface is clear to the user, and functioning. Implements UI based code such as buttons
Functional Requirement Lead	Ethan	Makes sure every function requirement is met during the coding process
Non-Functional Requirement Lead	Sam	Makes sure every NFR is met during the coding process

4 Workflow Plan

- For Git, we will have a master branch that will always have a working codebase, and upto date documentation. We will have a second branch called develop, this is the development branch, and all our new changes get merged into this branch. It is meant to be unstable and most of our issues will pop up in this branch. Every time develop has a stable build, we will push the code into master. We will have branches off of develop that we are going to use as our feature branches, every new line of code gets written in these branches, and when the feature is done it will get merged into develop. This double buffer system ensures we always have working code easily accessible in master. Sam will be responsible for merging develop into master.
- How will you be managing issues, including template issues, issue classificaiton, etc.?

5 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

6 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

7 Coding Standard

8 Project Scheduling

[How will the project be scheduled? —SS]