Table 1: Revision History

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

Development Plan
ProgName

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

[Put your introductory blurb here. —SS]

1 Team Meeting Plan

The team will meet twice a week on Tuesday and Thursday evenings between 6:00 PM and 7:30 PM. The length of meeting time may be adjusted according to the demands of the upcoming deliverable, setbacks and points of discussion. The meetings will take place primarily in the engineering library meeting rooms. Members may join through Discord or Messenger call or chat, and some meetings may be held completely virtually when appropriate.

1.1 Rules for Agenda

The rules for the agenda for all team meetings are as follows:

1. The members will alternate being the designated note-taker and chair for each meeting

2. Meeting minutes will be taken accordingly

3. The meeting agenda will focus on planning for the next milestone, agreements for distribution of the work, discussion of any pain points or concerns from any members, and review of current progress as necessary

4. All members are expected to attend meetings either in person or virtually

5. Members should prepare for meetings by noting down their current progress, roadblocks and any concerns that have arisen prior to the meeting

6. The meeting will primarily be lead by the chair of each meeting, however the remaining members are expected to contribute to discussions and provide insight

7. Meetings should conclude with decisions being made regarding the topics discussed

2 Team Communication Plan

The team will primarily communicate in person as most meetings will occur in person. The secondary avenue of communication will be through Messenger text chat or Discord voice chat. These may be used to address critical issues as they occur outside of meeting times, or issues and concerns that may be resolved quickly and efficiently in a virtual manner.
The codebase and documentation will be managed through GitHub. Code quality reviews and tasks will assigned and communicated through GitHub comments and tasks. Members are expected to make appropriate merge and pull requests when contributing to the repository.

3 Team Member Roles

4 Workflow Plan

- How will you be using git, including branches, pull request, etc.?
- 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

Code written should conform to the official rust style guideline, found at: https://doc.rust-lang.org/1.0.0/style/README.html and to google's JavaScript style guide, found at: https://google.github.io/styleguide/jsguide.html.
Classes and methods are to be named using *PascalCase*, and variables and parameters are to be named using *camelCase*.
The goal is to maintain readability and ease of maintenance for any other developers.

8 Project Scheduling

8.1 Scheduling software

The project schedule is organized as epics and issues in the team's Zenhub board. The epics are based of major milestones such as POC, Revision 0, Revision 1, etc. The issues are various tasks associated to these epics.

8.2 Task breakdown

The big tasks are broken down to issues which can be completed by the individual developers in 1-5 days any bigger tasks will be needed to be broken down. There will be some tasks such as the documents which will be worked on by multiple developers at once.

8.3 Task assignment

The developer assigned to each of the task will be assigned based on their proficiency related to the task, interest and available capacity. For example, one of our developer is extremely proficient at working with front end, thus will work on tasks related to front end. Some developers that are really interested in gaining experience in certain subjects will be assigned tasks related to that subject. Lastly, leftovers tasks will be assigned to developers who have leftover capacity.