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. The members will alternate being the designated note-taker and chair for each meeting. 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. 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. Members should prepare for meetings by noting down their current progress, roadblocks and any concerns that have arisen.

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

8 Project Scheduling

[How will the project be scheduled? —SS]