Development Plan
Software Engineering

Team 11, technically functional
Matthew Huynh
Cieran Diebolt
Vaisnavi Shanthamoorthy
Maham Siddiqui
Eman Ashraf

Table 1: Revision History

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

This document contains details pertaining to the proposed development plan by the team. Sections 1 - 3 contain legal and regulation information, sections 4 - 6 contain team dynamics related information and section 7-10 contain timeline and technology logistics. The team charter is also included within this document.

1 Confidential Information?

There is no confidential information involved in this project. Any information from supervisors or stakeholders is trade-wide information and contains no trade-secrets or other types of confidential information.

2 IP to Protect

Guaranteed intellectual property for the project consists solely of written content, namely the code and documentation. This written content is covered under our copyright agreement; more detail in the copyright license section below.

No additional creations requiring a trademark agreement or any further protection of intellectual property, such as logos or slogans, are foreseen at this point of the project.

3 Copyright License

The copyright license adopted by the team for this project is the MIT license. This license provides others the right to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the software under the necessity that the copyright notice and permissions are provided in all copies of the software.

The copyright license for the project can be found in the team repository.

4 Team Meeting Plan

The team will be meeting in person on a weekly basis in a library at McMaster University. The time of the meetings needs to be adjusted in order to accommodate all the members' time constraints. Additional virtual meetings may be required when approaching a deliverable deadline.
Industry supervisor meetings will be held virtually on a biweekly or as needed basis.

Matthew H. will be in charge of the agenda. The rest of the team will be recording meeting minutes interchangeably. All members will be encouraged to chair meetings throughout the duration of the project.

5 Team Communication Plan

[Issues on GitHub should be part of your communication plan. —SS]

6 Team Member Roles

Editor: This member will read through all deliverables throughout development and a final read-through before submission. They will be responsible for any text alterations to ensure conciseness and effective communication within documents/code.

Liaison: This member will be responsible for communicating with all external parties and relaying their comments back to the team.

Researcher: They will conduct initial research before the team commences a task and convey their findings to the rest of the team.

Task Manager: They will decide how to allocate the workload between the members.

Github Manager: They will ensure that all proposed rules pertaining to the use of Github are followed and that information provided on the platform is complete and precise.

Back-end developer: They will be developing and ensuring that the back-end is functional and up to code standards.

Front-end developer: They will be responsible for development of a user friendly interface.

Comment Reviewer: This member will review code and verify the presence of and clarity of comments. Maham will be reviewing the final version.

Github Wiki Manager: This member is responsible for verifying the appropriateness of all content on the Github Wiki.

Table 2: Member Roles

Role	**Name**	**Possibility of Change**
Editor	All	Yes
Liaison	Matthew H	No
Researcher	All	Yes
Task Manager	Eman A	Yes
Front end developer	All	Yes
Back end developer	All	NA
Github Manager	Cieran D	Yes
Comment Reviewer	All	NA
Github Wiki Manager	All	NA

7 Workflow Plan

We will primarily be using GitHub for managing the project files.
Each team member has their individual branches on which they will engage with. The member should open a pull request after completion of their assigned task. If multiple members are collaborating on the same task, they will use either member's branch. Once completed, a pull request can be opened which will be reviewed by the other team members.

Issues that focus on a single aspect of the project may be opened and will combine all relevant items. Sub-issues can be used if additional refinement is needed.

- How will you be using git, including branches, pull request, etc.?
- How will you be managing issues, including template issues, issue classification, etc.?
- Use of CI/CD

8 Project Decomposition and Scheduling

- How will you be using GitHub projects?
- Include a link to your GitHub project

[How will the project be scheduled? This is the big picture schedule, not details. You will need to reproduce information that is in the course outline for deadlines. —SS]

9 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?

10 Expected Technology

[What programming language or languages do you expect to use? What external libraries? What frameworks? What technologies. Are there major components of the implementation that you expect you will implement, despite the existence of libraries that provide the required functionality. For projects with machine learning, will you use pre-trained models, or be training your own model? —SS]

[The implementation decisions can, and likely will, change over the course of the project. The initial documentation should be written in an abstract way; it should be agnostic of the implementation choices, unless the implementation

choices are project constraints. However, recording our initial thoughts on implementation helps understand the challenge level and feasibility of a project. It may also help with early identification of areas where project members will need to augment their training. —SS]

Topics to discuss include the following:

- Specific programming language
- Specific libraries
- Pre-trained models
- 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
- Tools you will likely be using?

[git, GitHub and GitHub projects should be part of your technology. —SS]

11 Coding Standard

[What coding standard will you adopt? —SS]

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?

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

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

Maham

Having a plan eliminates a lot of uncertainty, whether it is timeline, team roles or conflict resolution methodologies. It allows the team to focus on what is important without being bogged down by frivolous details. Plus, it allows effective allocation of resources. For example, if one person is better at front end development, members who do not want to be or cannot be involved in it, do not have to start from scratch and master a new skill within a small time frame.

The main advantage of CI/CD is a smoother integration of code while reducing the amount of mistakes and code conflicts that would need to be fixed at the end of the project. However, there is one troublesome disadvantage - conflicting merge requests (this is in the context of using github) and the very time consuming endeavour of fixing them.

As for group disagreements, I do not think we had any. All members carefully listened to any input from their teammates and implemented any changes that were deemed to be beneficial. We were professional and respectful of each other's opinions and gave equal importance to everyone's feedback.

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

[What are your team's external goals for this project? These are not the goals related to the functionality or quality fo the project. These are the goals on what the team wishes to achieve with the project. Potential goals are to win a prize at the Capstone EXPO, or to have something to talk about in interviews, or to get an A+, etc. —SS]

Attendance

Expectations

[What are your team's expectations regarding meeting attendance (being on time, leaving early, missing meetings, etc.)? —SS]

Acceptable Excuse

[What constitutes an acceptable excuse for missing a meeting or a deadline? What types of excuses will not be considered acceptable? —SS]

In Case of Emergency

[What process will team members follow if they have an emergency and cannot attend a team meeting or complete their individual work promised for a team deliverable? —SS]

Accountability and Teamwork

Quality

[What are your team's expectations regarding the quality of team members' preparation for team meetings and the quality of the deliverables that members bring to the team? —SS]

Attitude

[What are your team's expectations regarding team members' ideas, interactions with the team, cooperation, attitudes, and anything else regarding team member contributions? Do you want to introduce a code of conduct? Do you want a conflict resolution plan? Can adopt existing codes of conduct. —SS]

Stay on Track

[What methods will be used to keep the team on track? How will your team ensure that members contribute as expected to the team and that the team performs as expected? How will your team reward members who do well and manage members whose performance is below expectations? What are the consequences for someone not contributing their fair share? —SS]

[You may wish to use the project management metrics collected for the TA and instructor for this. —SS]

[You can set target metrics for attendance, commits, etc. What are the consequences if someone doesn't hit their targets? Do they need to bring the coffee to the next team meeting? Does the team need to make an appointment with their TA, or the instructor? Are there incentives for reaching targets early? —SS]

Team Building

The team will engage in social activities together if schedules allow. Team building activities will strongly be encouraged. There may be sessions of working together in person and spending some time together with any member that is available. This will also depend on work habits and preferences.

Decision Making

[How will you make decisions in your group? Consensus? Vote? How will you handle disagreements? —SS]