

# 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

Our team communication plan will be as follows:

- Issues: GitHub/GitHub Projects
- Weekly Meetings: In-person on campus
- Meetings outside the weekly meetings: Teams
- Meeting minutes: GitHub
- Meetings with stakeholders: Zoom
- Project discussion (asynchronous): Teams

## 6 Team Member Roles

See Table 2: Member Roles for team assignments.

**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.

## 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

The application will follow the steps below (high-level):

1. A user will record them performing a physio exercise (decide on body part?)
2. Determine if the exercise is “good” form or “bad” form based on requirements from the stakeholders
3. From the analysis, the application will provide guidance to improve the efficacy of the exercise

Some potential risks from this process flow are:

- Variations in how different users will record each video.
  - Having a guide overlay to indicate where they should film the exercise from.
- Inaccuracies with depth or monitoring a 3D motion on a 2D plane
  - Prompting the user to also perform the exercise from another angle and then analyzing two different recordings.

- Hardware limitations - as the application would be easiest to use on a phone due to the camera already being on there, it may have limitations when processing the footage.
  - Using a lightweight solution to process the recording to reduce lag.
- Accuracy in the footage may be different in a non-controlled environment and lead to inaccurate assessment of the exercise.
  - A demonstration can consist of seeing if we are able to obtain the same feedback in different light levels/clothing to see if it will affect our input data.

## 10 Expected Technology

It's important to note the technologies listed in this section are expected to be used within this project but are not set as requirements or bounding limitations on the technology for the project. See Table 3: Expected Technologies.

Continuous integration within the project will be used for simple checks like latex build success, compilation success, and simple test cases throughout the project to ensure to changes to the main branch break the project and cause unnecessary pain points for the team. Additionally, using continuous integration to enforce our coding standards makes sure our formatting and naming conventions stay consistent throughout the entire duration of the project.

## 11 Coding Standard

The coding standards used for this project will follow those made by the creators of the language. For example, if we proceed with using Python as the main development language for our project then PEP8 will be enforced as it was made by the language developers. Some standards made by developers lack clear cut rules in certain cases where developers didn't intend for the language's usage, but we will discuss as a team to negate any ambiguity in such cases. Linters will be used within our development platforms to enforce the selected coding standards, this will keep a clean and consistent format to the code as well as clear naming conventions for well manageable variables.

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

Table 3: Expected Technology

Technology Role	Name	Explanation
Repository	GitHub	Used to store project code and documentation for combined contributions.
Project Management	GitHub Projects	Used to track timelines for deliverables, meetings, and specific issues for implementation.
Language	Python	Backend object-oriented programming language.
Linters	Pylint	Linters for consistent formatting and coding standards.
Platform	VSCode	Standard development platform for the team.
Platform	Android Studio	Platform for creating AndroidOS applications. Has integrated linter.
Library	OpenCV	Library for extracting data from video footage.
Plugin	Chaquopy	Plugin for integrating Python with Android Studio applications.
Unit Testing	Python Extension	Plugin for additional Python support including unit testing and code coverage.
Unit Testing	Pytest	Specific unit testing software with more options.

## 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.

### **Matthew**

I think it is important to create a development plan prior to starting the project because it helps our team navigate conflict down the line. Not everyone works the same and so this development plan acts as a formal document to communicate with the team and find common ground for everyone to perform well on. Furthermore, it is a form of having an awkward conversation now compared to

down the line when this work wasn't done and there is no basis to critique a member's performance.

An advantage of CI/CD is the reduced manual effort to test and evaluate code. If there is a problem with one of our documents that we upload or a file that is incorrect, we can have the tests evaluate it and be flagged for issues with it. A downside of CI/CD is the overhead or the amount of time that it takes to set up. For a shorter project it will take a bit longer to set up and thus less time is put towards coding the features at the cost of automating the building and testing.

The group did not have any disagreements during this deliverable. Everyone was able to communicate to each other and each meeting went with a plan and left with actionable items to complete. As the term gets busier and there are more deliverables for other classes I think that the team charter below will help guide us for the rest of the project.

### **Cieran**

The development plan sets the expectation for not only team members in their roles and responsibilities but also for the direction of the project as a whole. Creating the development plan prior to starting the project gives everyone a good idea, hopefully, of how to approach any foreseeable situation. I think it also makes everyone think of the project in advance and gets some of the team 'storming' out of the way early on.

The advantages of using CI/CD are excellent in my opinion; continually automatic test cases, compilation checks, and more are extra assurance for the team that the project is still moving forward. It makes it easier for developers on the team to contribute quickly and get assurance they haven't dropped the ball on the way. The downsides are the setup and the upkeep; setting up the CI/CD isn't hard but making sure it's testing the right things takes more in-depth knowledge of the project. Same with upkeep, if you add a new method, how do you apply CI/CD to that now? It becomes trickier, but I do believe the pros outweigh the cons.

There was some disagreement on which project to select initially and a lack of pushback on the team charter. We have group members that are passionate, and that's great, but it felt bad to not be able to realize each person's dream of their capstone project. We settled the former issue with a vote after exploring both options in further detail, and all group members were professional about the outcome. The latter issue is something I fear; ambiguity or oversight here could make future conflicts a little more awkward.



## Appendix — Team Charter

[borrows from [University of Portland Team Charter —SS](#)]

### External Goals

Our group's primary goal is to create a project that we are proud to have our names on, can be talked about in interviews and to peers. It is important for us to build a solution to our problem statement while also adhering to code quality such as readability and documentation. Through this, we seek to develop our technical skills through tackling a real world problem. Additionally, we are also aiming for above-average grades as a secondary goal.

### Attendance

#### Expectations

Our team expects full attendance to meetings scheduled in advance for the entire duration of the meeting. If a member cannot attend a meeting or is going to miss a portion of the meeting, they must communicate it in advance to the team and catch-up on the missed portion of the meeting (i.e. meeting minutes). Missing 3 meetings in a row when a deliverable is not the main point of discussion will trigger a meeting with the rest of the team. Moreover, if a deliverable is the main topic, 2 missed meetings will be the limit. These standards are for non-emergency situations. For emergencies, see below.

#### Acceptable Excuse

Acceptable excuses include: Academic sessions (classes, tutorials), academic requirements (due-dates prior to the capstone deliverable) and medical appointments. An advance notice for medical appointments will be appreciated. It is the responsibility of the missing member to make up for a missed meeting. Repeated excuses will not be acceptable; please refer to attendance expectations. Please see below for emergency cases.

#### In Case of Emergency

In the event of an emergency case, the team member should contact the team. Extended cases of emergency (more than 2 weeks or time taken off that impedes deliverables) should be brought to the attention of the teaching staff to ensure the work load is manageable for the rest of the team.

### Accountability and Teamwork

#### Quality

In order to maintain consistent quality for deliverables, each weekly meeting will briefly go over deliverable expectations. Each deliverable should be checked

against the rubric [on avenue] and also against the repository checklist by the individual that wrote that deliverable but also by a teammate.

Furthermore, for meetings, team members are expected to arrive, review relevant material and provide any problems or status updates with their work to ensure the meeting stays on-topic and productive.

Lastly, during weekly meetings we will discuss “what went well, what to change” to see if there is a way to streamline communication to attain this quality. We also encourage each team member to ask for help in our informal chat in a timely manner while being respectful of each person’s time.

### **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**

When making a decision on the team, we will lean towards a unanimous vote. We want each member of the team to feel their feedback is heard to improve the project communication. In cases where this is not the case, we will hold a vote. Since there are 5 members, no one can abstain that way a solution can be achieved.

Each member is allowed to voice their opinion to the team without interruption so everyone is heard and we can reach an objective solution (without personal feelings swaying the decision).

In the event that voting cannot be held properly, we will select a neutral mediator to guide us further.