Development Plan
ProgName

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

Table 1: Revision History

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

[Put your introductory blurb here. Often the blurb is a brief roadmap of what is contained in the report. —SS]

[Additional information on the development plan can be found in the lecture slides. —SS]

1 Confidential Information

This project doesn't contain any confidential information.

2 IP to Protect

This project doesn't contain any IP to protect.

3 Copyright License

Our team is adopting the MIT License, which can be found here

4 Team Meeting Plan

The team will meet weekly on Tuesdays from 3:00pm to 4:00pm virtually on Discord or in person on campus if needed. The team will meet with the industry advisor biweekly on Thursdays from 2:30pm to 3:30pm. These meetings with the industry advisor will be conducted either online on Microsoft Teams or in person on campus.

The meetings will be structured as follows:

1. An agenda prepared by the meeting chair (who rotates among team members each week) will be made to use as a guide for the meeting.
2. The team will go over any announcements or completed To-Dos from the previous week's meeting if needed.
3. Each member will present what they have worked on so far and ask the remaining group members for feedback or any questions if needed.
4. The team will discuss and document any decisions needed about the deliverables or the project.
5. Any concerns/questions will be documented for the next team meeting or for the next industry advisor meeting.

5 Team Communication Plan

- **Discord**: Our main method of communication between group members. It will be used to discuss detailed deliverable and any code questions. Additionally, group meetings will be hosted on discord.
- **Instagram**: Our secondary method of communication between group members. It will be used to discuss less technical details and for any urgent messages that require a quicker response.
- **Teams**: Our main method of communication with our Supervisor. We will utilize our group chat with our supervisor for any quick questions or updates. Online meetings with our supervisor will be hosted on teams.
- **GitHub**: The issues feature will be utilized to communicate any bugs observed and meeting attendances. Additionally, as a way to see what feature each team member is working on and their progress.

6 Team Member Roles

The team will work collaboratively to develop and refine this project. To ensure a clear division of tasks, each team member have been assigned roles that align with their areas of expertise and contribute to achieving the goals of this project. These roles will rotate throughout the year to prevent overspecialization and to ensure that all members can gain experience and knowledge in every aspect of the project.

The defined roles and responsibilities per team member is as follows:

- **Fiza Sehar**: *Developer, Documentation, Model Training Specialist*
  Fiza will be responsible for developing features and maintaining documentation for the project. She will also be leading the model training for the project to ensure efficient and accurate performance.
- **Dhruv Sardana**: *Developer, Documentation, Full-Stack Specialist*
  Dhruv will work across both frontend and backend development, and in ensuring a seamless integration and functionality. He will also support in writing documentation for this project.
- **Nawaal Fatima**: *Developer, Documentation, Data Specialist*
  Nawaal will also work on developing features with a focus on data management, pre-processing, and analysis in this project. She will also contribute in the documentation of the project.
- **Moly Mikhail**: *Developer, Documentation, Backend Specialist*
  Moly will be handling the APIs, database management, and system logic focusing on the backend of the project. She will also support in the documentation of the project.

- **Casey Francine Bulaclac**: *Developer, Documentation, Frontend Specialist*
  Francine will be responsible for the design and implementation of the user interface, ensuring correct usability and accessibility while assisting in the project documentation.

7 Workflow Plan

- **Git Workflow**
  We will be using git, branches and pull requests in order to divide work between group members and complete tasks concurrently. Furthermore, we will follow a feature-branch based approach, our process will follow these steps:
  - **Step 1: Permanent Branches**: The project repository will have two permanent branches.
    - **Develop**: This branch will be used integrate different features and ensure that they work successfully together. Code can only be merged into the develop branch after being reviewed by one other team member who wasn't working on the feature.
    - **Main**: This branch will be used to maintain the most stable version of the application. Code in the develop branch can only merged into the main branch after its been extensively tested and been reviewed and approved by a majority of team members.
  - **Step 2: Feature/Bug Branches** Group member will create a branch from the `develop` branch (after pulling the most recent changes) for each feature they work on.
    Naming Conventions:
    - **Features:** `feature_[contributorName]/D#_[featureDescription]`
    - **Bugs:** `bug_[contributorName]/D#_[bugDescription]`
  - **Step 3: Pull Request** Once a group member is done working on their feature, they will open a pull request to merge into the develop branch. Group members will use the Github comment feature to connect their feature to any applicable issue numbers. They will also assign a group member to review their code.

- **Continuous Integration Continuous Development**

  For continuous integration (CI), we will be using Github actions to automatically trigger a project build and run unit tests whenever a commit occurs or pull requests are opened. This ensures errors are discovered as early as possible and that the project remains in a working state during development.

- **Use of Labels**
  The team will utilize labels to help us organize ongoing work as well as

work that needs to be completed. The tags will enable us to see the status of ongoing work and their respective priority. The issue tags we will have available are:

  - Bug, feature, question, in-progress, review needed, changes needed, ready for merge, done, high priority, low priority, meeting, frontend, backend, feedback

- **Managing issues**
  We will utilize the capTemplate template issues for tracking attendance, peer review, supervisor, TA, and team meetings. Furthermore, issues that fall outside these categories will utilize the blank issue and the labels mentioned above.
  The issue will have a clear title, description and an owner to be responsible for the issue. This will ensure issues are easily identified and managed. Once an issue has been resolved, it will be linked to the PR containing the code, allowing the team to keep track of the issues status.

- **Use of Checklists**
  The group will utilize the deliverable checklists to ensure our work meets the expectations. We will also have checklists in the Pull Request templates to ensure that all required items are completed prior to merging. The checklist will include the following items:

  - Unit Tests Passed
  - Coding Standard is followed
  - Code Compiles without errors
  - Reviewed by 2 team members

8 Project Decomposition and Scheduling

Project Schedule:

Deliverables	Due Date
Problem Statement, Proof of Concept, and Development Plan	Week 04
Software Requirements Specificaiton and Hazards Analysis (Revision 0)	Week 06
Verification & Validation Plan (Revision 0)	Week 08
Design Document (Rev-1)	Week 10
Proof of Concept Demonstration	Week 11 + 12
Design Document (Revision 0)	Week 16
Project Demonstration (Revision 0)	Week 18 + 19
Verification & Validation Report (Revision 0)	Week 22
Final Demonstration (Revision 1)	Week 24
Final Documentation	Week 26
Capstone EXPO	Week 26

- 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

11 Coding Standard

[What coding standard will you adopt? —SS]

Table 2: Expected Technologies

Technology	**Choice**	**Reasoning**
Backend Language	Python	The team is most familiar with this language and it provides many free and easy to use machine learning (ML) libraries.
Frontend Language and Framework	JavaScript with React	The team is also familiar with using JavaScript and React. The combination allows you to make interactive and maintainable interfaces through components.
UI Design	Figma	Figma will allow us to design and visualize our UI prior to building it. Figma also allows collaboration.
Libraries	Pandas, Tensorflow or Pytorch	These Python libraries will be utilized to build our own ML model.
Pre-trained models	None	We will be making our own model and not used a pre-trained model.
Linter	Pylint(Python) and ESLint (JavaScript)	We will use these linters as they are free tools that will allow us to detect potential errors early on and follows a consistent coding style.
Unit Testing Framework	Pytest(Python) and Jest (Javascript)	These unit testing frameworks will allow us to easily write and run unit tests. They also support parameterized unit tests, allowing us to run the same test with different data.
Coverage Tools	Coverage.py (Python) and JSover(JavaScript)	We will use these these free tools to confirm that our unit tests reach all possible code paths.
Version Control	Git and GitHub	Git and GitHub will allow the team to easily collaborate and present our code.
Continuous Integration	GitHub Actions	GitHub Actions will allow us to automates builds and unit tests on commits.
Project Management Tool	GitHub Projects	GitHub Projects will be used to plan and organize tasks as well as track progress in order to meet deadlines.

Appendix — Reflection

[Not required for CAS 741 —SS]

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:

Moly Mikhail Reflection

1. **Why is it important to create a development plan prior to starting the project?**
   I believe it is important to create a development prior to starting a project as it lets you consider ahead of time the different components. It allows you to plan out the work to be done and the different technologies that will be required. Furthermore, if a technology is needed such as machine learning or a python library that some team members are not familiar with, planning prior to starting the project will give them a chance to schedule time to learn and practice with the technology. Another reason it is important to create a development plan is that it ensures that the project begins with a strong foundation. For example, discussing unit testing, code coverage and git practices prior to starting the project will ensure the project remains organized and stable throughout the development process. It also sets expectations between group members of the code quality expected and the pull request review process to be completed.

2. **In your opinion, what are the advantages and disadvantages of using CI/CD?**
   I believe there are many advantages and disadvantages to using CI/CD. One advantage is that it ensures that the project remains stable and behaves as expected when new features are implemented. For example, when a developer completes a feature, CI/CD checks that all previously functioning application features remain working as expected and no bugs are introduced. Another advantage is that when the application is in production and being used by customers, CI/CD will help ensure that releases only contain code that has passed all written tests. This improves customer experiences as it reduces the chance of them encountering bugs or errors. One disadvantage of CI/CD is that it can be very time-consuming and require many resources. For example, if a project is hosted on AWS, continuously redeploying the application can require many resources and

increase cost. Another disadvantage of CI/CD is that when the CI/CD pipeline fails, it often requires a more experienced individual or a specialist to investigate the reason for the failure.

3. **What disagreements did your group have in this deliverable, if any, and how did you resolve them?** We didn't face any disagreements when completing this deliverable.

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

[How will you build team cohesion (fun time, group rituals, etc.)? —SS]

Decision Making

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