

Development Plan

Software Engineering

Team #11, OKKM Insights
Mathew Petronilho
Oleg Glotov
Kyle McMaster
Kartik Chaudhari

Table 1: Revision History

Date	Developer(s)	Change
9/18/2024	Mathew Petronilho	Added Member Roles and Coding Standards
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?

[State whether your project has confidential information from industry, or not. If there is confidential information, point to the agreement you have in place. —SS]

[For most teams this section will just state that there is no confidential information to protect. —SS]

2 IP to Protect

[State whether there is IP to protect. If there is, point to the agreement. All students who are working on a project that requires an IP agreement are also required to sign the “Intellectual Property Guide Acknowledgement.” —SS]

3 Copyright License

[What copyright license is your team adopting. Point to the license in your repo. —SS]

4 Team Meeting Plan

The team will meet weekly on Mondays from 11:30 until 13:20, in a study room booked by Kyle McMaster. The schedule to meet with our supervisor will be determined when our supervisor is confirmed.

Weekly meetings will be chaired by Kyle, who will prepare an agenda to be sent in advance of team meetings. Team members can add additional agenda items as needed. Team members are also able to request additional meetings if necessary. In this case, it is their responsibility to chair the meeting, organize a meeting location, and provide an agenda. Following all meetings, the meeting chair will prepare a list of action items.

5 Team Communication Plan

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

6 Team Member Roles

Member Name	Roles	
Mathew Petronilho	Document Manager, Front-End Development Expert	Developer, Tester, PR Reviewer, Issue Creator, Meeting Participant, and Note Taker.
Oleg Glotov	Team Lead, Project Manager	
Kyle McMaster	Meeting Chair, Back-End Development Expert	
Kartik Chaudhari	Customer Relations Manager, Git Expert	

Table 2: Team Roles

- **All Team Members:** Every team member is responsible for developing code and creating tests for the code. Everyone is also responsible for reviewing open pull requests and providing feedback if necessary. Additionally, all members will be tasked with creating issues using the appropriate templates, tracking issue status, and updating issues assigned to them with relevant information. Each member is expected to attend meetings punctually and contribute ideas to discussions, while maintaining respectful and concise communication. The role of meeting note taker will rotate among members in each meeting. The note taker should keep track of meeting attendance, issues discussed, decisions made, and action items. These notes should be well-maintained and easily accessible to all team members.

If we encounter challenges, we may consider switching roles to maintain progress and improve team performance. More specific roles can be assigned as the project evolves and implementation details become clearer.

- **Mathew Petronilho:** Responsible for ensuring that all documents are formatted consistently, that all necessary components are included, and that there are no grammatical or spelling errors. Also responsible for assisting team members with the front-end and taking the lead in implementing this component of the project.
- **Oleg Glotov:** Responsible for liaising with the supervisor, teaching assistant, and professor. Coordinates project tasks among team members, organizes meetings, ensures equitable distribution of work, and monitors deadlines to ensure they are met.
- **Kyle McMaster:** Responsible for creating meeting agendas, guiding discussions, managing meeting time, and resolving conflicts. Also responsible for assisting team members with the application’s back-end logic and deploying services.
- **Kartik Chaudhari:** Responsible for contacting potential customers and managing customer relationships. Oversees the GitHub repository by or-

ganizing files and ensuring it is updated to reflect project progress. Provides guidance to team members on resolving issues related to Git.

[You should identify the types of roles you anticipate, like notetaker, leader, meeting chair, reviewer. Assigning specific people to those roles is not necessary at this stage. In a student team the role of the individuals will likely change throughout the year. —SS]

7 Workflow Plan

- 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

7.1 Git Workflow

7.2 Issue Management

7.3 Usage of CI/CD

7.3.1 Tex Files

To ensure the PDFs found in our repository are consistent with the tex files they are generated from, we have developed a CI workflow in GitHub actions. This workflow detects when a push has been made to the *docs* folder in the *main* branch. When this happens, it automatically regenerates the relevant documents and pushes the new PDFs to the repository. This will ensure that our PDFs are always the most recent version.

7.3.2 Linting & Static Checks

As discussed in sections 10 and 11, we expect we will use Python to develop the backend component of our software. To improve the clarity of our code and our ability to collaborate, we have decided to use static type checking in Python. This is possible with a package called [mypy](#). We will integrate static type checking as part of our CI/CD workflow.

For both the front and back ends of the project, we have selected coding standards as described in section 11. To ensure these standards are met, we will include a linting step before pull requests can be merged into the main branch.

7.3.3 Deployment

Depending on how we decided to host our server, we will also investigate if it would be beneficial to develop a continuous deployment workflow when changes are pushed to the main branch. This will be explored in the future when our requirements are more clear.

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

Our back-end code, written in Python, will adhere to PEP 8 for code formatting and PEP 484 for type annotations. For our front-end code, written in JavaScript, we will follow the JavaScript Standard Style. [\[What coding standard will you adopt? —SS\]](#)

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:

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?

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]