

# Development Plan

## Software Engineering

Team #5, Money Making Mauraders  
Zhenia Sigayev  
Justin Ho  
Thomas Wang  
Michael Shi  
Johnny Qu

Table 1: Revision History

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

This document outlines the development plan for the MES Club Payment Tracking System, a web application designed to streamline the reimbursement process for McMaster Engineering Society (MES) clubs. The current process using Google Forms and spreadsheets is inefficient, error-prone, and difficult to search. This plan provides the roadmap for design and implementation, describes team and workflow organization, lists the expected technologies, and identifies major risks and mitigation strategies.

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

## 1 Confidential Information?

This project does not contain industry confidential information.

## 2 IP to Protect

The McMaster Engineering Society (MES) will be the primary owner of the system and its deployed instance. The student development team will be credited as contributors. Unless MES requests otherwise, the project will be published under an open-source-friendly license allowing MES and future student teams to maintain and extend the system. If MES requests proprietary handling, we will follow the agreement provided by MES and notify course staff.

## 3 Copyright License

Given that the MES's website source code [is open-source](#), we will be using the MIT License.

## 4 Team Meeting Plan

- Weekly team meetings: 60 minutes, fixed weekday time (set by team availability) for planning and progress updates.
- Bi-weekly stakeholder check-ins: 30 minutes with MES representative(s) to gather requirements and demo progress.
- Meeting format: rotating chair; agenda created as a GitHub issue before each meeting; minutes recorded in the issue and a short summary added to the project wiki.
- Ad-hoc technical sessions as needed for design decisions or integration tasks.
- Meeting Notes: Will be documented in GitHub Issues linked to the relevant sprint or task.

## 5 Team Communication Plan

- GitHub Issues for tasks, bugs, feature requests and meeting agendas.
- Pull requests for all code changes with at least one reviewer before merging.
- Discord for quick coordination.
- Email for formal stakeholder communications.
- Weekly status updates posted to a shared project board or README for the MES contact.

## 6 Team Member Roles

[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

- Git branching: single protected **main** branch for production; feature branches named **feature/short-description**; pull requests for merges.
- Pull request policy: descriptive PRs, link to issues, at least one approving review and passing CI.
- Issue management: issue templates for bug, feature, and documentation; labels for priority and component (frontend/backend/ops).
- CI/CD: automatic test runs on PRs; deployments to a staging environment on merges to **develop** (optional) and production on **main** after review.
- Code reviews: enforce linting and basic tests; use GitHub Actions for automated checks (linters, tests, security scanning).

## 8 Project Decomposition and Scheduling

- We will use GitHub Issues and GitHub Projects to manage work and track progress. Issues will represent tasks, bugs, and feature requests; Projects will be used for sprint planning and high-level roadmaps.
- GitHub Projects board (create and link when available): [Project board placeholder](#)

- Capstone Repository: <https://github.com/zheniasigayev/MES-Finance-Tracking>
- MES Website Repository: <https://github.com/McMaster-Engineering-Society/MES-Website-App-Router>

[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 javascript with node.js MongoDB
- Specific libraries Not exactly libraries but used extensions: Error Lens, Tailwind CSS intellisense, babel javascript, auto close tag, auto rename tag, better comments
- Pre-trained models
- Specific linter tool (if appropriate) ESLint, prettier
- 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 Github actions

- 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

[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 of 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

Team members are expected to:

- Attend scheduled meetings and be on time.
- Notify the Project Manager and meeting chair in advance if they cannot attend.
- Act professionally and responsibly when absent (review minutes and follow up on assigned actions).

Preferred notification is the day before; same-day notification is acceptable for emergencies.

#### Acceptable Excuse

Any one of the following reasons that McMaster University recognises, will also be accepted. Below is a list of McMaster's acceptable extenuating circumstances:

- **Medical issues:** a physical or mental health condition that negatively affected participation or performance. Provide a physician's note or equivalent documentation when available.
- **Family crises:** significant personal or family situations (e.g., bereavement, serious illness) that create hardship.
- **Other personal emergencies:** unforeseen and significant events that disrupt academic or project commitments.
- **Religious observances:** conflicts with religious obligations where accommodation is required.
- **Varsity reasons:** official university sports commitments.

- **Business commitments (part-time students):** job-related duties that prevent attendance.

For urgent or sensitive circumstances where documentation cannot be produced immediately, notify the Project Manager promptly and follow up with documentation as soon as reasonably possible. Trivial reasons (e.g., social events) are not considered acceptable without prior agreement.

### **In Case of Emergency**

If a team member has an emergency and cannot attend a meeting or complete agreed work, follow these courteous steps:

1. Notify the Project Manager and meeting chair as soon as possible (direct message and a short message in the team chat). If unavailable, notify any team lead.
2. Create a short GitHub Issue titled "emergency: {YourName}" describing current status, outstanding tasks, and the expected return time (if known). Link related issues or PRs.
3. Where appropriate, assign a backup or delegate specific issues/tasks to another team member and update issue ownership. Use the handover checklist in the issue description.
4. Share any quick access notes or pointers needed to continue work (location of credentials, relevant commands, how to run tests). Do not post sensitive credentials in chat — use the team's secure credential store or inform the Project Manager privately.
5. If the absence affects an upcoming demo or deadline, request an immediate short meeting or asynchronous decision from the Project Manager about scope adjustments or re-assignment.
6. When able, provide a brief follow-up update and, if requested, supporting documentation. On return, do a short handover with whoever covered your work.

These steps prioritise clear communication and minimise disruption while respecting the privacy of the person experiencing the emergency.

## **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]