Development Plan
Software Engineering

Team 4, EventHub
Virochaan Ravichandran Gowri
Omar Al-Asfar
Rayyan Suhail
Ibrahim Quraishi
Mohammad Mahdi Mahboob

Table 1: Revision History

Date	**Developer(s)**	**Change**
September 21, 2025	Ibrahim, Mohammad, Omar	Revision 0

This document will outline the development plan for **EventHub**: handling of confidential information; IP protection and copyright licensing; team roles, collaboration, and organization guidelines; project development guidelines; project workflow and version control guidelines; and expected programming tools, technologies, and standards.
This project will be conducted in collaboration with two other Capstone teams to create one large unified product, and as such, the contents in these sections are subject to change based on updated requirements from the project supervisors or design decisions made in conjunction with the collaborating teams.

1 Confidential Information?

The project will deal with information confidential to the McMaster Engineering Society (MES) and the Canadian Federation of Engineering Students (CFES). This information includes program generated analytics and financial reports for events hosted by the MES through the proposed system, and results from surveys conducted by the CFES.

2 IP to Protect

There is currently no IP which requires protection, and the project is planned to be open-source.

3 Copyright License

This project will use the GNU General Public License version 3 (GPLv3). The license can be found here.

4 Team Meeting Plan

The members of the team are expected to meet at least once weekly virtually through Microsoft Teams. The purpose of these meetings is to provide updates on current work, plan accordingly for submission of deliverables, discuss next steps, and distribute tasks among team members. The structure of this meeting will be as follows.

1. Each team member will provide a 5–6-minute update on the work completed within the last week. After each member has completed their update, the other team members are free to provide feedback.

2. The meeting leader will go over next steps on current deliverables, as well as remind team members of upcoming deliverables

3. If any new work is to be assigned, the meeting leader will split the work into a set of tasks. Team members will then select which tasks they are comfortable with taking
4. The remaining meeting time will be spent on preparation for the upcoming sync meeting with the supervisors

Additionally, a weekly sync meeting will be scheduled with the industry advisor and team supervisor to ensure that the direction of the project remains aligned with the needs of the supervisors. The meeting will ensure the supervisors are kept up to date with the state of the project and allow the supervisors to provide feedback or request changes. If there are no significant updates within the week before the meeting (i.e. during exam season), the meeting will be cancelled with at least 24 hours notice for the supervisors.

5 Team Communication Plan

- Issues: GitHub
- Meetings: MS Teams, Discord
- Meetings (with advisor): MS Teams
- Project Discussion: Discord

6 Team Member Roles

The following administrative roles will be assigned to team members:

- Team Liason: Main contact point between the team members and the supervisors. This member is responsible for sending emails for communication between the team and supervisors as well as scheduling and leading the weekly sync meetings.
- Inter-Team Liason: Responsible for communication between capstone teams. Since this project is a sub-component of a larger system for the MES with multiple teams working together, it is important to have a representative for the team who will be responsible for communication between teams to ensure consistency and compatibility between design decisions for the project.
- Meeting Leader: Responsible for facilitating the weekly meetings between team members. Ensures that team meetings happen consistently and that team members are up to date with deliverables.
- Note Taker: Responsible for taking notes summarizing key points from every meeting, including team meetings and supervising meetings. This member is also responsible for creating GitHub Issues for every meeting and providing a summary of attendance and key takeaways.

- Reviewer: Responsible for final review of any main deliverables before final submission to ensure they align with the course and supervisors' expectations.

7 Workflow Plan

The repository will have two primary branches, one for development and of for documentation, responsible for technical development and software documentation respectively.

The average workflow will be as follows:

- Pull changes from primary branch depending on the task at hand
- Create an issue for the assigned task if one does not already exist
- Create a new branch with the naming convention of: [assigned task - name of contributor]
  - If it is a new task, create a branch direcly under the primary
  - If it is a follow up/subtask, create a sub-working branch branch from previous branch
- Commit changes with detailed message
- Create unit tests for changes
- Open a pull request to the primary branch
- Link issue to pull request
- Wait for approval from other members and for testing to pass
- Merge pull request

Issues will be managed through GitHub Issues. Through GitHub Project boards, issues will be linked to notable meetings and milestones, and classified into categories: Meetings, To Do, In Progress, and Done. Each issue will be assigned to the relevant team members, who can provide feedback, report errors, and organize tasks. Issues will be tagged with distinct labels to indicate their priority, timeline, and type (e.g., bug, feature, documentation).

8 Project Decomposition and Scheduling

The project is hosted on GitHub at the following link:
A Kanban Board project setup is currently being used to track the progress of the project. The board is divided into four columns: Backlog, To Do, In Progress, and Done. Each task is represented as a card that can be moved across the columns as it progresses through different stages of completion. This visual representation allows team members to easily see the status of each task and identify any bottlenecks in the workflow.
The cards can pertain to:

- Lectures
- Group and Supervisor Meetings
- Project Milestones

The project is scheduled to be completed over the course of two semesters. However, we aim to have a working prototype by January to be tested across various events, including the Fireball Formal, and CALE Conference.

Deliverable	Due Date
Problem Statement and Goals, and Development Plan	Week 4
Requirements Documentation and Hazard Analysis (Revision 0)	Week 6
Verification and Validation Plan (Revision 0)	Week 8
Proof of Concept Report	Week 9
Design Document (Revision 0)	Week 10
Proof of Concept demos	Week 11
Functional Prototype	January

9 Proof of Concept Demonstration Plan

The main challenges we foresee for this project are creating the form builder and integrating user-side and admin-side interfaces for the application. The form-builder needs to be modular and track all data against a user to perform analytics in order to minimize redundancy and improve user experience. The interface integration is also challenging because they demand an entirely different set of features for the client, yet they must interact with the same underlying system.
To get a working proof-of-concept, we expect to demonstrate a simplified interface with at least singular interactions from either endpoints which can communicate correctly with the underlying database. If the database is correctly and efficiently track all data, then we can expect that the final product can be developed with confidence.

10 Expected Technology

At the request of the supervisor, the following tools and technologies must be used for development of the project to ensure compatibility and seamless integration between capstone teams.

- JavaScript, and HTML/CSS shall be used in development of the frontend for both the web-based admin portal, and the user mobile application. React and Next.js will be used as the primary framework for the admin portal and React Native will be used to develop the mobile application.
- PostgreSQL shall be used as the primary database for storage of any system and user data.

Additionally, the following programming languages, frameworks and libraries are expected to be used during development and may change at the discretion of the team.

- JavaScript along with Node.js may be used in development of the backend server for both the web portal and mobile application. Additionally, Node.js modules such as node-postgres may be used to communicate with the database.
- A styling library/framework such as Tailwind or Bootstrap may be used to simplify UI styling
- A REST API will be used for communication between the frontend applications and the backend server. The default Fetch API provided within React and React Native may be used on the frontend, along with Node.js modules such as Express to handle requests on the backend server
- Jest will be used as the primary unit testing framework for testing both the frontend and backend code with JavaScript, including unit testing and code coverage. Additionally, the React Testing Library may be used to test frontend components.
- A linter such as Prettier will be used to ensure the source code adheres to a specific standard.

GitHub will be used as the project repository for documentation, project planning and administration, and source code management.

- A GitHub Issue will be created for each attended lecture, tutorial, and team meeting outlining attendance, and key takeaways.
- A GitHub Issue will be created for each course deliverable, such as documentation, and code submissions
- GitHub Projects will be used to host a Kanban Board of all GitHub Issues and their status

- GitHub Actions will be used as the main CI platform. CI will be used to maintain code styling requirements and perform regression tests on pull requests before they can be merged. Additionally, GitHub actions may be used build and deploy source code into a staging environment.
- Git will be used as the source code management platform for making code changes. The specific development environment in which these changes are made, such as the IDE, are subject to the preferences of the team members.

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?

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]