Problem Statement and Goals
Software Engineering

Team #12, Streamliners
Mahad Ahmed
Abyan Jaigirdar
Prerna Prabhu
Farhan Rahman
Ali Zia

Table 1: Revision History

Date	Developer(s)	Change
2025-09-20	Prerna Prabhu	Initial draft of Problem Statement and Goals
...	...	...

1 Problem Statement

1.1 Problem

The McMaster Engineering Society (MES) organizes large-scale events such as Fireball Formal, Graduation Formal, and Pub Nights, often drawing several hundred attendees. Currently, registration, ticketing, waiver collection, and check-in processes are fragmented across multiple platforms (Google Forms, spreadsheets, Discord, Instagram, etc.). This lack of centralization increases the administrative burden on student organizers and creates a confusing, inconsistent experience for attendees.

Students struggle to:

- Access clear, consolidated event information.
- Register efficiently for events and related logistics (tables, buses).
- Receive timely updates or reminders.

Organizers struggle to:

- Manage ticket sales, waitlists, and attendee data in one place.

- Track accessibility/dietary requirements effectively.
- Minimize repetitive manual effort across multiple tools.

The absence of a centralized system reduces event memorability, increases missed opportunities, and wastes volunteer time that could be spent on event quality rather than administration.

1.2 Inputs and Outputs

Inputs:

- Student event registrations (tickets, RSVPs, bus signups, table preferences).
- Waiver acknowledgements and personal details (e.g., dietary and accessibility requirements).
- Payment details for ticket purchases.
- Admin inputs for event setup (ticket types, capacities, schedules, notifications).

Outputs:

- Confirmation of event registration and digital tickets/QR codes.
- Notifications and reminders about events.
- Waitlist updates and allocations.
- Admin dashboards showing ticket sales, attendee demographics, accessibility/dietary data, and financial tracking.
- Check-in validation at event entry points.

1.3 Stakeholders

Primary Stakeholders:

- MES Event Attendees (students): Use the platform to register, purchase tickets, receive updates, and check in.
- MES Event Organizers/Volunteers: Use the platform to manage events, track registrations, ticket sales and payments, waivers, handle check-ins, and communicate with attendees.

Secondary Stakeholders:

- MES Executives: Oversee finances, risk management, and reporting.

Tertiary Stakeholders:

- Sponsors and partners: Interested in gaining visibility, engagement, and smooth execution.
- McMaster University Administration: Indirectly involved for compliance, liability via waivers, and student satisfaction.

1.4 Environment

Hardware Environment:

- Attendees: Smartphones (iOS/Android) and laptops for registration, notifications, and check-in.
- Organizers: Laptops/desktops for backend dashboards, mobile devices for on-site management and QR code scanning.

Software Environment:

- Web-based admin dashboard for event creation and analytics.
- Cross-platform mobile app for students (primary focus) with fallback web access.
- Payment integration (e.g., Stripe, Square, Paypal).
- Backend database to store user, event, and financial data.

2 Goals

Centralized Registration & Ticketing

- **Description:** The platform must consolidate ticket purchasing, registration, and RSVPs into one place.
- **Justification:** This eliminates confusion caused by scattered tools and ensures students always have access to the latest event information.

Payment Integration

- **Description:** Provide secure and flexible payment options (Stripe, Square, PayPal).
- **Justification:** Enables students to pay quickly with widely used systems while reducing cash-handling risks for organizers.

Role-Based Access Control (RBAC/FBAC)

- **Description:** Implement granular permissions so organizers only see/ manage the tools relevant to their role.
- **Justification:** Reduces errors, ensures security, and improves efficiency in large event teams.

Bus & Table Sign-ups

- **Description:** Allow attendees to reserve buses and tables with automatic capacity tracking.
- **Justification:** Simplifies logistics and avoids overbooking, replacing spreadsheets and manual coordination.

Notifications & Reminders

- **Description:** Send push notifications and reminders for registrations, updates, and cancellations.
- **Justification:** Improves event memorability, prevents missed opportunities, and reduces no-shows.

Analytics & Reporting

- **Description:** Provide organizers with real-time dashboards on sales, demographics, and waitlists.
- **Justification:** Supports decision-making, improves resource allocation, and helps executives evaluate event success.

Attendee Experience

- **Description:** Ensure the platform is mobile-first, intuitive, and accessible.
- **Justification:** Reduces barriers to use, supports inclusivity, and maximizes student engagement.

3 Stretch Goals

- Personalized event recommendations based on student interests.
- Calendar integration (Google/Outlook/Apple).
- Post-event engagement features (photo gallery, feedback surveys, lost & found updates).
- Sponsor visibility tools (logos, surveys, branded features).
- Dietary/accessibility matching algorithms for meal and seating planning.

4 Extras

- Usability report and testing with MES organizers and student participants.
- User documentation for both attendees and admin organizers.

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. What went well while writing this deliverable?

   The team was able to understand the problem quickly because of each member's familiarity with MES events and how they are currently run. From our own experiences, we already understood the main pain points as well as the features that would be most valuable to both organizers and attendees. This made it easier to distinguish between core goals of the project and additional features. Since we had a clear outline of what the system should do provided by the supervisor, our team was able to divide tasks quickly and effectively for this deliverable. Collaboration went smoothly, and we tackled the work for this deliverable in a fast and organized way. Having shared context gave us confidence in identifying the users' needs and building a well-structured document. Being able to use github issues to organize and divide the work also helped us stay on track and ensure that everyone contributed equally.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   One of the biggest challenges was that we did not know which project we would be assigned until very close to the deadline. Especially, since this project is not a traditional capstone project, in the way that it is divided amongst two other groups, and the scope is quite large. This made it difficult to start early and limited the amount of feedback we could receive from the TA before submission. Even though we already understood the problem well, it was sometimes difficult to frame our knowledge at a high-level overview instead of diving into details. Another challenge was technical: setting up LaTeX and repositories was new for some of us and was time-consuming to get familiar with. We also struggled to clearly differentiate between goals vs. stretch goals, and primary vs. secondary stakeholders (especially since both students and organizers are highly active users). Finally, balancing the level of detail such as being specific

enough to show thought in requirements, but high-level enough to avoid over-scoping was an ongoing challenge. We resolved these issues through group discussions, reviewing example documents, and iterating on the structure.

3. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?

   At the beginning, we had many ambitious ideas, including projects involving AI and large language models, as well as other advanced projects that were highly related to specific team members' interests. However, we realized that not all team members had the technical background or confidence to pursue those ideas effectively. This made us step back and focus on a project that was both practical and meaningful to all of us. The MES event management platform came naturally from our own past experiences and frustrations, which made it more grounded and relatable. We also discussed how it could connect to our co-op experiences: it would challenge us to learn new skills such as payment integration, role-based access control, and mobile app development, which some of us have prior experience in while others will be completely new to these concepts, while still refining technical skills we already had from previous work terms. By focusing on a project with clear real-world impact but narrowing it to a feasible scope, we ensured it was the right balance of ambitious and achievable for a capstone. Additionally, since the project is divided among two other groups, we made sure to clearly define our team's specific responsibilities and the features that we can work on without overlapping or depending on other groups' contributions. This way we are able to manage our workload effectively and ensure that we can deliver a complete and functional product within the given timeframe and not be blocked by other teams' progress or takuing on too many features.