

Problem Statement and Goals

Software Engineering

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

Table 1: Revision History

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

1 Problem Statement

[You should check your problem statement with the [problem statement checklist](#). —SS]

[You can change the section headings, as long as you include the required information. —SS]

1.1 Problem

1.2 Inputs and Outputs

[Characterize the problem in terms of “high level” inputs and outputs. Use abstraction so that you can avoid details. —SS]

1.3 Stakeholders

1.4 Environment

[Hardware and Software Environment —SS]

2 Goals

3 Stretch Goals

4 Extras

[For CAS 741: State whether the project is a research project. This designation, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

[For SE Capstone: List your extras. Potential extras include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. (The full list is on the course outline and in Lecture 02.) Normally the number of extras will be two. Approval of the extras will be part of the discussion with the instructor for approving the project. The extras, with the approval (or request) of the instructor, can be modified over the course of the term. —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. 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 taking on too many features.