

Hazard Analysis Software Engineering

Team #12, Streamliners

Mahad Ahmed

Abyan Jaigirdar

Perna Prabhu

Farhan Rahman

Ali Zia

Table 1: Revision History

| Date | Developer(s) | Change |
|-------------|---------------------|---|
| 2025-10-02 | Farhan | Critical assumptions made for the project |
| Date2 | Name(s) | Description of changes |
| ... | ... | ... |

Contents

| | | |
|---|--------------------------------------|---|
| 1 | Introduction | 1 |
| 2 | Scope and Purpose of Hazard Analysis | 1 |
| 3 | System Boundaries and Components | 1 |
| 4 | Critical Assumptions | 1 |
| 5 | Failure Mode and Effect Analysis | 2 |
| 6 | Safety and Security Requirements | 2 |
| 7 | Roadmap | 2 |

[You are free to modify this template. —SS]

1 Introduction

[You can include your definition of what a hazard is here. —SS]

2 Scope and Purpose of Hazard Analysis

[You should say what **loss** could be incurred because of the hazards. —SS]

3 System Boundaries and Components

[Dividing the system into components will help you brainstorm the hazards. You shouldn't do a full design of the components, just get a feel for the major ones. For projects that involve hardware, the components will typically include each individual piece of hardware. If your software will have a database, or an important library, these are also potential components. —SS]

4 Critical Assumptions

The following assumptions have been identified as critical to the safe and reliable operation of the MacSync platform.

- **Reliable Internet Access:** It is assumed that both attendees and organizers will have access to stable internet connections during registration, payment, and check-in. While temporary connectivity loss may occur, the system must handle these cases gracefully.
- **Third-Party Service Availability:** The platform depends on external services such as payment processors (e.g., Stripe, PayPal) and hosting infrastructure. It is assumed these services provide high availability, but the system will still account for outages or delays to prevent complete operational failure.
- **Device Compatibility:** It is assumed that attendees will primarily use modern smartphones and organizers will have access to laptops or mobile devices capable of running the dashboard. Reliance on outdated devices or unsupported browsers must be minimized through compatibility testing.
- **User Data Accuracy:** The system assumes that users provide correct information (e.g., dietary restrictions, accessibility needs, payment details). However, hazards tied to incorrect or incomplete inputs will be addressed by validation checks.

- **Organizational Oversight:** It is assumed that event organizers will actively monitor the system for anomalies (e.g., payment disputes, capacity errors, failed notifications). This system will assist and automate many of the tasks and centralize information, but human oversight is still necessary to manage unexpected situations.
- **Security Measures:** It is assumed that standard security practices (encrypted storage, secure authentication, and role-based access control) will be implemented and maintained. Failure to enforce these could expose sensitive student data or enable fraudulent event access.

5 Failure Mode and Effect Analysis

[Include your FMEA table here. This is the most important part of this document. —SS] [The safety requirements in the table do not have to have the prefix SR. The most important thing is to show traceability to your SRS. You might trace to requirements you have already written, or you might need to add new requirements. —SS] [If no safety requirement can be devised, other mitigation strategies can be entered in the table, including strategies involving providing additional documentation, and/or test cases. —SS]

6 Safety and Security Requirements

[Newly discovered requirements. These should also be added to the SRS. (A rationale design process how and why to fake it.) —SS]

7 Roadmap

[Which safety requirements will be implemented as part of the capstone timeline? Which requirements will be implemented in the future? —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?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?
4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?