# Hazard Analysis
# 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** |
|---|---|---|
| Oct 2 | Johnny Qu | Complete document |

# Contents

# 1 Introduction

This document will serve as a hazard analysis of the MES Club Payment Tracking System. This system is a web application designed to streamline the reimbursement process for clubs under the McMaster Engineering Society. As a primarily software system, a hazard is not defined by physical injury, but rather, being related to data integrity, data security, and service availability.

# 2 Scope and Purpose of Hazard Analysis

Hazards in the MES Club Payment Tracking System can cause harm or damage in the following ways.

1. Financial Harm
2. Operational Harm
3. Privacy Harm
4. Reputational Harm

# 3 System Boundaries and Components

Our systems can be broken down into the following functional components

1. Authentication
2. Request Submission
3. Budget Tracking
4. Notifications
5. Administrative Tools
6. Data Storage

# 4 Critical Assumptions

The hazard analysis will be made with the following critical assumptions in mind;

1. The machines that will be used in deployment are physically secured. This means we assume that no one will have physical access to the machines to perform malicious actions (e.g. reading the hard drive).
2. The machines that will be used in deployment are stable and available. For example, they have a consistent and reliable network connection and stable power.

3. User devices have minimum requirements to run application (i.e. javascript enabled, adequate network connection).

## 5 Failure Mode and Effect Analysis

| Component | Failure Modes | Failure Effect | Causes of Failure | Risk Priority Number (Likelihood, Severity, Detection) | Recommended Actions |
|---|---|---|---|---|---|
| Authentication | User cannot log in | | User forgets authentication details User mistypes credentials on creation | 5 x 2 x 2 = 20 | Implement authentication reset processes |
| Authentication | Unauthorized Access | Sensitive/private information exposed Reputational damage | Security vulnerability | 1 x 5 x 5 = 25 | Enforce password strength Keep externally used packages up to date Lock account on discovery |
| Submission | Duplicate reimbursement submitted | Incorrect reports Club underspending | Multiple users submit for the same reimbursement | 3 x 3 x 2 = 18 | Clearly show all submitted Allow reimbursements to be deleted |

| | | | | | |
|---|---|---|---|---|---|
| Submission | Incorrect details submitted | Club overspending | Incorrect image to text parsing<br>Negligent user | 4 x 3 x 2 = 24 | Allow users to modify submissions pending review |
| Submission | Unable to upload receipts | Unable to provide adequate information to process a reimbursement | Backend logic failure | 3 x 4 x 1 = 12 | Show descriptive errors<br>Provide support contact |
| Budgeting | Reporting incorrect numbers | Club overspending<br>MES overspending | Rounding errors | 2 x 4 x 4 = 32 | Ensure all money formats are rounded up and to 2 decimal places |

| | | | | | |
|---|---|---|---|---|---|
| Database | Data loss | Incorrect Loss of users Loss of clubs Loss of reimbursement requests | Errors during deploy Bad database migration Accidental query execution | 3 x 5 x 3 = 45 | Regularly create database backups |
| Notification | Notification not visible | Users unaware of required actions to complete Users unaware of reimbursement status | Incorrect email address Notification reported as junk | 2 x 1 x 4 = 8 | Have multiple delivery channels, including an in app |
| Administrative Tools | Unwanted action performed | User access removed Incorrect | Negligent user | 3 x 3 x 2 = 18 | Implement reversible actions For irreversible actions, add confirmation before performing actions |
| Administrative Tools | Abuse of controls | MES Overspending Reputational damage | Collusion | 1 x 5 x 4 = 20 | Log all actions performed by administrators Allow administrators to audit each other |

# 6 Safety and Security Requirements

Newly discovered requirements

1. Database backups must be made on a weekly basis
2. Enforce role based access on the database to ensure data uploaded by a user on a club are only visible to users within the same club and admin-

istrators.

3. Passwords must not be stored in plain text
4. Administrative actions must be reversible
5. All administrative actions must be logged and visible to all other administrators
6. Administrative action logs must not be deletable.
7. Dollar amounts should always be rounded up and precise to 2 decimal places.

# 7 Roadmap

All the above requirements will be addressed within the timeline of the project.

If the project is to be expanded beyond the timeline, here are some future requirements to consider

1. Training system maintainers with automated recovery drills
2. Continuous penetration testing and security audits

# 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?

   (a) The deliverable helped us formalize many of the concerns we had been implicitly considering throughout design and development. It encouraged structured thinking about hazards rather than simply listing "things that could go wrong." Once we established the system components and boundaries, it became easier to identify credible hazards and match them with realistic mitigations. Collaboratively reviewing the document also helped align our team's understanding of safety and security concepts within a software context.

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

   (a) The FMEA was a little difficult to complete, as it challenged us to seriously consider the depth and breadth of potential system failures. While writing, it was also difficult to discern whether a thought would fall under "failure mode," "failure effect," or "cause of failure." To overcome these issues, we looked at previous examples and the lecture for support, which clarified how to separate symptoms from root causes. We also discussed ambiguous cases as a team to reach consensus and maintain consistency across the table.

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?

   (a) Before writing this, we had already identified general data security and authentication concerns—mainly the need for strong access control and proper password storage. However, this deliverable prompted us to recognize additional hazards we had not yet considered, such as administrative misuse of privileges, data corruption during database migrations, and the importance of maintaining audit logs and reversible administrative actions. These insights expanded our understanding of how operational and governance failures can pose just as much risk as purely technical ones.

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?

   (a) Data security: Many software systems collect and store sensitive user data, such as names, contact details, and financial information. If this data is exposed or mishandled, it can lead to identity theft, financial loss, and loss of trust in the organization. Data security risks are critical to consider because breaches can have long-term reputational and legal consequences.

   (b) Authentication and access control: Weak authentication mechanisms can allow unauthorized users to access restricted data or functions, potentially leading to data tampering, impersonation, or financial harm. Proper authentication design—including strong password policies, role-based permissions, and multi-factor authentication—is essential to ensure that users only perform actions appropriate to their role and authority.