Hazard Analysis
McMaster Engineering Society Custom Financial Expense Reporting Platform

Team #12, Reimbursement Rangers
Adam Podolak
Evan Sturmey
Christian Petricca
Austin Bennett
Jacob Kish

Table 1: Revision History

Date	**Developer(s)**	**Change**
10/25/2024	All team members	Revision 0, including all sections and appendix

Contents

1 Introduction **1**

2 Scope and Purpose of Hazard Analysis **1**

3 System Boundaries and Components **1**

4 Critical Assumptions **2**

5 Failure Mode and Effect Analysis **2**

6 Safety and Security Requirements **6**

7 Roadmap **6**

7.1 Requirements During Capstone Timeline 6

7.2 Requirements For The Future 6

1 Introduction

This document provides an outline and anaylsis of hazards that may arise during the development of the MES finanical platform Hazards do not necessarily cause problems on their own but can lead to failures or errors when they interact with other system components in specific contexts.

2 Scope and Purpose of Hazard Analysis

In the scope of this project, a **hazard** refers to any potential source of failure that could result in undesirable outcomes, particularly concerning data security, system reliability, functionality, and user experience. Specifically, these hazards could impact our project by posing risks to users and the MES system in terms of:

- Compliance with legal and regulatory confidentiality agreements,
- Reliability in processing and tracking payments for audit accuracy,
- Ensuring timely and correct reimbursement to users and partner organizations.

By identifying and managing these hazards, we aim to protect the integrity and dependability of our system, ensuring a secure and seamless experience for all users.

3 System Boundaries and Components

- **User Information Database:** A secured storage location for user data that may need to be repeatedly referenced, such as contact or bank information.
- **Backup Database:** A separate database, holding similar data to the User Information DB, but backed up regularly via the main database. Used in the event old data is needed or there is a problem with the main db.
- **User Interface:** An umbrella term for the graphical layout of the software system. All interactions between user and system will be facilitated through the UI, which provides a visual outline of system usage.
- **Payment/Funding Request Module:** A partition of the system into functions and interfaces supporting reimbursement, payment, and intramural funding requests. A liason between users making requests and users with the ability to approve or deny.
- **Invoice Generation Module:** A partition of the system supporting the generation and submission of personalized invoices.

- **Notification Module:** A partition of the system supporting automated notifications for users to track statuses of payment/reimbursement requests, etc via email.
- **Authentication Module:** A partition of the system supporting login and authentication of privileges and actions for users, such as making, approving, or denying requests.
- **Stripe API:** External API for streamlining future integration with other MES services or even third parties.

4 Critical Assumptions

- **Reliability of Third-Party FinTech Services:** We assume that any external fintech or payment processor used in the system will not fail to meet legal compliance or breach user privacy agreements.
- **University IT Infrastructure:** We assume that McMaster University's IT infrastructure, which the platform relies on, will remain stable and secure.
- **Network Stability:** We assume that users will have consistent internet access when interacting with the system, as the platform relies on an online connection.
- **Audit Regulations:** We assume that the MES's audit and regulatory requirements will not drastically change during the system's lifecycle, allowing current compliance protocols to remain valid.
- **Compliance with Financial Institutions:** We assume that Canadian banks or other financial institutions used for reimbursement disbursements will maintain their API and integration standards for processing payments.
- **Student Information Accuracy:** We assume that student data provided by the university (for login and user verification) is accurate and up-to-date.
- **Data Hosting Provider Stability:** We assume that DigitalOcean (or another cloud service provider) will continue to offer reliable hosting services and maintain their SLAs for uptime and security.
- **Third-Party Software Licenses:** We assume that any third-party software (e.g., for PDF generation or data visualization) used in the system will remain licensed and supported during the platform's lifetime.

5 Failure Mode and Effect Analysis

Table 2: FMEA for User Authentication and Access Control Design Function

Design function	**Failure Modes**	**Effects of Failure**	**Causes of Failure**	**Recommended Actions**	**Req.**
User Authentication and Access Control	Unauthorized user access	Unauthorized users gain access to sensitive data	a. Weak password policies/restrictions b. Lack of multi-factor authentication	a. Enforce users to create strong passwords b. Implement and enforce multi-factor authentication	a. SCR2 b. PRD2
	User accounts are compromised	Data theft, manipulation, or system sabotage	a. Weak or reused passwords	a. Enforce strong password policies	a. SCR1
	Improper privilege escalation by users	Users perform actions beyond their authorization	a. Missing or incorrect configuration of access controls	a. Use role-based access control	a. AR1

Table 3: FMEA for Data Privacy and Compliance Design Function

Design function	**Failure Modes**	**Effects of Failure**	**Causes of Failure**	**Recommended Actions**	**Req.**
Data Privacy and Compliance	Exposure of confidential personal financial information	Legal/financial penalties and damage to the organization's reputation	a. Poor security practices b. Lack of encryption of data in transit	a. Encrypt all data containing sensitive financial information at rest and in transit	a. PVR1
	Not complying with regulations relating to data protection	Repercussions from regulatory bodies, potential shutdown of organization's services	a. Lack of compliance measures b. Not adhering to legal requirements	a. Regular reviews and updates for policies regarding data compliance	a. LR1 b. LR2

Table 4: FMEA for Data Backup and Recovery Design Function

Design function	**Failure Modes**	**Effects of Failure**	**Causes of Failure**	**Recommended Actions**	**Req.**
Data Backup and Recovery	Scheduled data backup fails to execute or complete	Financial data is lost	a. Errors in the configuration of backup schedule automation b. General software errors or bugs	a. Regularly monitor backup processes b. Conduct regular integration testing throughout development	a. RFT1 b. MR1
	Data that is backed up is corrupted	Inability to retrieve data or restore system functionality promptly	a. Remote database failures	a. Store backed-up data in multiple, secure locations	a. IMM1

6 Safety and Security Requirements

- **SCR3.** The system shall implement multi-factor authentication (MHA) to prevent unauthorized access.
- **SCR4.** The system shall automatically log out users who have been inactive for 5 minutes.
- **SCR5.** The system shall ensure that users make strong passwords, including requiring a minimum length, use of special characters and a mix of uppercase and lowercase letters.
- **SCR6.** The system shall prompt users to alter their passwords every 12 months.

These additional safety and security requirements have been added to the SRS document.

7 Roadmap

7.1 Requirements During Capstone Timeline

The following safety and security requirements will be implemented as part of the capstone timeline. These requirements are either required for the core functionality of the system or are small enough in scope to be added in during the project timeline.

- **SCR1**
- **SCR2**
- **INR1**
- **PVR1**
- **PVR2**
- **ADR1**
- **ADR2**

7.2 Requirements For The Future

The following safety and security requirements will be implemented in the future, after the capstone timeline. These requirements are not required for the core functionality of the system and will take many resources to implement.

- **SCR3**
- **SCR4**

- **SCR5**
- **SCR6**
- **IMM1**

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?

   Christian: The Critical Assumptions were not too much of a challenge once I got clarification from the TA on what should be focused on. It was a good exercise in considering the scope of our work and what falls on what side of the fence when it comes to our responsibilities.

   Adam: For my section (FMEA tables) brainstorming the actual hazards/risks for the system went relatively well. There aren't any inherent "safety" risks that would endanger anyone's physical well-being, there really are only security risks that we have to worry about. Since we're dealing with financial information, these risks were quite obvious to identify and mitigate. Evidently, we need encryption, user authentication, etc. to ensure that we're upholding data privacy and financial information is stored securely.

   Austin: I really enjoyed how straight forwards this deliverable is as compared to the previous one. Adequately defining hazards in the context of our capstone is incredibly valuable as these will be considerations that influence every design detail of our project.

   Jacob: For section 3, there were a couple of obvious partitions, specifically the databases. It's logical and commonplace to think of databases as separate from the other software components, so with that there were a couple easy partitions to make right out of the gate.

   Evan: For section 6, I found it easy to brainstorm additional safety and security requirements. Going through the hazard analysis was useful in finding areas that we might have missed and potential risks we may have overlooked. It shows how important each step of documentation truly is.

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

   Christian: The main pain point would have been making sure what constitutes a "critical" assumption, which the TA was a huge help with.

Many assumptions could be made about the system but deciding which were the ones exclusively outside of our scope of responsibility was the main task of this section and it became easier after coming up with the first couple.

Adam: One of the biggest pain points I had while writing my section was formatting the tables using latex. I had to import extra packages and spend a lot of time playing around with the spacing of things to ensure it was legible and formatted decently. Unfortunately, This took more time than the actual engineering work of coming up with the hazards. It would be nice if we could make the tables in a Word document and then insert a screenshot into the Latex file, but I can see how this would defeat the purpose of traceability.

Jacob: For section 3, the difficulty came from how to partition the system. Since it is all software, there are no concrete components to separate. So, it was necessary to reason what logical partitions might be, and decide if a partition was too large or small to make sense.

Evan: The biggest difficulty during this deliverable was the dependency on the SRS document. We were asked to make use of our requirements from our SRS and to make additional requirements. I found this to be challenging since at the time of making this document, we have not received feedback about our SRS.

Austin: There are countless kinds of hazards, some of which may or may not be relevant to the scope of our project, its quite easy to list every kind of risk you can think of. Rereading our problem statements and clearly delinating what we're really trying to accomplish helped distiguish the kinds of hazards we need to focus on.

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?

   Team: Before this deliverable, our team mainly though about briefly on potential integration risks like connecting to third party APIs, usability risks through ensuring users are successfuly able to navigate the platform, and security risks ensuring user data is safely stored in databases. However while doing the hazard analysis we developed additional safety and security risks, that will ensure we deliver a robust plaform to MES.

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?

   Christian: The biggest known risk computers pose to computer users is data privacy. Anything a computer is trusted with is something it can pose a risk towards as well. While people don't use computers to do everything, most people trust computers with sensitive personal information that they

would not trust people with, and in this respect it poses a threat. The other threat it poses is reliance to conduct tasks that may have previously been undertaken by people. As soon as a technology replaces a person in completing a task rather than becoming another, more convenient, option, it poses a serious threat to the ability of a person to say, handle their finances remotely at a time when they may have not anticipated needing to visit a physical branch, or relevant financial entities.

Jacob: One big risk is data security. We will be handling sensitive personal information from users, so it is critical that this data stays well-secured and out of the hands of any unauthorized users to protect people's privacy and banking assets. Secondly, large system downtimes or failures is a huge risk. A long outage has the potential to seriously slow and cripple financial operations that may lead to funding and payment issues, or losing track of debts and payments.

Adam: Another type of risk, and the one we will be mostly dealing with, is a security risk. More specifically, security risks to personal financial information/data. This also includes unauthorized access to user accounts, which could jeopardize data confidentiality and integrity. This is important to consider because this could lead to legal and financial repercussions. Financial data breaches and leaks can also lead to the organization losing its reputation. Another form of risk is operational risk. This relates to the stability and performance of the system. This is important to consider because operational performance can sometimes lead to other failures that lead to other risks, for example, the system fails a data backup, or a bug fails to encrypt data, and data is lost or exposed (security risk).

Evan: One type of risk for software products is integration risks. This arises when products require interfacing with other systems or applications. Compatibility issues can arise which would affect the performance and customer satisfaction of the product. Another type of risk is compliance risk. It is important to ensure that software products do not violate any laws or regulations. If a product were to violate any standards, it could lead to fines, legal penalties and ruin the reputation of the brand.

Austin: Less commonly, ethical risk, there are always some kind of inherent biases that flow into the development of software and a specific example can be gender bias risks. Consequently, this can result in reputational, social or legal implications due to lack of ethical considerations. Antoher risk is cultural risks to software, this largely impacts usability of software as underlying cultural differences can lead to uninteded use of software, for example the meaning beind symbols can be different based on culture.