Hazard Analysis
MES-ERP

Team #26, Ethical Pals
Sufyan Motala
Rachid Khneisser
Housam Alamour
Omar Muhammad
Taaha Atif

Contents

List of Tables

List of Figures

Table 1: Revision History

Date	Version	Developer(s)	Change
October 15, 2024	0.1	Omar Muhammad	Introduction, scope and system boundaries
October 18, 2024	0.2	Housam Alamour	Critical Assumptions, started on FMEA
October 25, 2024	0.3	Sufyan Motala	Safety and Security Requirements and reflection
April 3, 2025	1.0	Sufyan Motala	Revised based on rubric feedback: Added missing elements (ToC, Lists), refined intro/scope, added labeled SSRs, completed FMEA table, improved hazard identification specificity.

1 Introduction

A hazard is defined as a condition or event that can result in harm, failure, or an undesirable outcome in a system. For engineers, hazards are risks that can affect the safety of a system, its functionality, or its operational integrity. For this project, MES-ERP, a possible solution discussed was a web based app for phones, laptops, and computers. Specifically, in our project, a hazard is identified as any potential event or condition that could lead to financial loss, data breach, unauthorized access, incorrect processing of requests, system downtime, or significant user frustration, thereby compromising the system's intended function of streamlining MES financial operations. ~~can be identified as something that would degrade the operational integrity of the system, cause a crash in our system, expose a user to information that should not be available to them, or grant a user access to something they shouldn't have access to.~~ This document outlines the scope, system components, assumptions, identified hazards, analysis (using FMEA), and resulting safety requirements.

2 Scope and Purpose of Hazard Analysis

The scope of the hazard analysis ~~will include any potential risk that will degrade operational integrity, cause crashes, or give users unauthorized access/information to system functionalities.~~ encompasses potential failures within the MES-ERP web application, including its frontend interface, backend processing logic, database interactions, authentication mechanisms, and integration points (like notification services). We will analyze hazards related to data input/output, processing accuracy, security vulnerabilities, access control failures, system availability, and usability issues that could lead to incorrect financial outcomes or data exposure. The losses incurred from ~~the listed scope~~ these hazards would include unauthorized access to sensitive financial data (like reimbursement amounts, budget details), exposure to restricted information, incorrect payment processing, or ~~exposure to restricted information. This risk could lead to a~~ failure to maintain accurate data ~~of the reimbursement requests~~. For example, if the backend of the app ~~is edited without proper authorization~~ incorrectly calculates budget totals, it will cause the McMaster Engineering Society to have false information and could ~~make them miss out on reimbursement requests~~ lead to overspending, which lowers the integrity of the app. A final loss would be degraded system functionality or complete system failure if the app cannot handle ~~various inputs from users~~ unexpected data formats (e.g., in receipt uploads) or high traffic loads. If these risks occur frequently, the user base of the application will not be happy and we will risk them not wanting to use the app. By conducting this hazard analysis, we intend to ~~limit~~ identify and propose mitigations for ~~all~~ these risks, ~~by limiting the risks we ensure~~ to ensure the users of the application will remain content with it and will continue to use the app.

3 System Boundaries and Components

- **Component: Database (Supabase/PostgreSQL)**
  - Hazards:
    1. Unauthorized access leading to data breaches (e.g., exposing financial details of clubs).
    2. Data corruption or loss during operations or updates (e.g., failed budget update transaction).
    3. Insufficient backups causing irreversible data loss.
    4. Inconsistent data due to race conditions during concurrent updates.
    5. Performance degradation under high query load.
- **Component: Front End (Next.js/React)**
  - Hazards:
    1. Unhandled inputs (e.g., invalid characters in forms, large file uploads) leading to system crashes or incorrect data submission.
    2. Poor user experience ~~with performance speed.~~ due to slow page loads or unresponsive UI elements.
    3. Poor user experience from a lack of clear error messages or system feedback.
    4. Browser compatibility issues causing incorrect display of content or non-functional features.
    5. ~~Browser compatibility issues causing crashes.~~
    6. Cross-Site Scripting (XSS) vulnerabilities if user input is not properly sanitized before display.
    7. Failure to correctly reflect backend status changes in real-time (e.g., request approval).
- **Component: Back End (Next.js API Routes / Supabase Functions)**
  - Hazards:
    1. System crashes due to unhandled exceptions or inputs (e.g., unexpected API request format).
    2. Failure to process data correctly, leading to incorrect outputs (e.g., miscalculation of budget totals, incorrect status updates).
    3. Poor security implementation exposing APIs to unauthorized access or data manipulation.
    4. Logic errors in approval workflows leading to incorrect routing or status changes.
    5. Failure to properly integrate with external services (SendGrid, Twilio) resulting in missed notifications.

- **Component: Hardware/Server (Assumed McMaster/MES provided)**
  - Hazards:
    1. Power failures or hardware malfunctions causing system unavailability.
    2. Weak server side processing power causing increased latency during peak usage.
    3. No ~~potential~~ backup server or failover mechanism in case of emergency.
    4. Insufficient server capacity (CPU, RAM, storage) for handling peak loads or data growth.
- **Component: Authentication System (Supabase Auth)**
  - Hazards:
    1. Unauthorized access due to weak authentication protocols or compromised credentials.
    2. Mismanagement of user roles (RBAC) leading to incorrect access control (privilege escalation or denial of necessary access).
    3. Session hijacking vulnerabilities.
    4. Failure to properly revoke access for users who leave MES/clubs.
- **Component: Reimbursement System (Input/Output)**
  - Hazards:
    1. Incorrect data submission (e.g. incorrect amounts, invalid receipts, OCR errors) causing delays or rejection of requests.
    2. Output discrepancies (e.g. incorrect approvals or incorrect reimbursement amounts) due to calculation or logic errors in the backend.
    3. Failure to notify the appropriate parties (clubs or administrators) regarding the status of the request, leading to confusion and delays.
    4. Data tampering during the approval process, allowing unauthorized changes to reimbursement requests.
    5. Lack of exception handling, leading to duplicate requests or other unintended circumstances (e.g., submitting the same request twice quickly).
    6. Inaccurate budget tracking due to errors in associating expenses with the correct budget lines.

4 Critical Assumptions

1. **Assumption 1: Reliable Internet Connectivity**
   The system assumes that all users~~, including student leaders, administrators, and MES staff,~~ will have access to reliable and stable internet connections when interacting with the platform. This assumption is important because the platform is designed to work through a web-based interface ~~. This will require~~ requiring real-time data processing. Reliable internet connectivity is necessary for users to access features like submitting reimbursement requests, tracking the status of payments, and viewing financial reports without interruption. If the connection is unstable, users might experience delays or errors during data submission, which may lead to incomplete reimbursement requests or user frustration. Also, poor connectivity could ~~stop~~ delay the delivery of notifications, meaning users ~~will~~ may not get critical ~~notifications and~~ updates promptly. Addressing potential connectivity issues early through ~~notifications in the interface~~ clear UI feedback, error-handling when ~~there is no connection~~ requests fail, and potentially ~~providing offline access options could mitigate~~ local caching mechanisms could mitigate some risks associated with this assumption.

2. **Assumption 2: Server Availability and Performance**
   It is assumed that the ~~McMaster Engineering Society (MES) will provide a dedicated~~ server hosting the application ~~with sufficient resources, including CPU power, memory, and storage, to support the platform's operations.~~ has adequate resources (CPU, RAM, storage, network bandwidth) to handle the expected workload. The server must handle ~~the anticipated workload,~~ real-time updates, and concurrent access by ~~all administrator of the reimbursement program of the MES as well as administrators of the clubs during peak periods~~ the expected number of users (approx. 60+ groups plus MES staff). ~~The server's storage capacity should be enough to store financial data for at least 3 years.~~ Sufficient storage must be available for financial records, receipts, and audit logs, potentially spanning multiple years as required by MES policy. This assumption is important because if the server cannot handle the project load, this could result in slower response times, data processing delays, or system crashes. This would directly impact user experience and the efficiency of financial operations. If ~~these are not met or if future demand necessitates~~ server resources are insufficient, ~~then~~ there may be a need ~~to upgrade~~ for upgrades, potentially increasing ~~project~~ operational costs ~~and delaying deployment~~. Regular performance monitoring should be planned to ensure server capacity aligns with user demand over time.

3. **Assumption 3: User Compliance with Data Entry Standards**
   It is assumed that all users of the platform~~, including student group leaders and MES staff,~~ will adhere to ~~the established data entry standards~~ reasonable data entry practices when submitting information into the system. This includes entering accurate details for ~~each~~ reimbursement request~~s,~~

~~such as the amount, purpose, and necessary attachments like receipts~~. Sticking to these standards ensures that the data processed by the platform is accurate and complete, minimizing errors during financial reviews and report generation. If users do not comply ~~with these standards~~, there is a risk of data entry errors (e.g., incorrect amounts, mismatched categories, unreadable receipts) leading to rejected requests, delays in processing, and additional administrative workload to correct mistakes. Proper user training and clear guidelines within the application should ~~be implemented so all users comply~~ guide users. Additionally, the platform should include input validation and error-checking features to reduce the likelihood of incorrect data entries.

4. **Assumption 4: User Access to Compatible Devices and Browsers**
   It is assumed that all users~~, including student group leaders, administrators, group members and MES staff,~~ will have access to devices (desktops, laptops) and browsers (recent versions of Chrome, Firefox, Edge, Safari) that are compatible with the ~~new~~ platform. The platform is designed to function optimally on modern operating systems~~, as well as on commonly used browsers~~. This assumption is critical because the system's user interface and performance rely on up-to-date browser features and operating system support to ensure a smooth user experience. If users attempt to access the platform on outdated devices or unsupported browsers, they may experience issues such as reduced functionality, slower response times, or display errors. To mitigate this risk, it will be essential to clearly communicate the platform's minimum system requirements to users ~~and provide support for upgrading or accessing compatible devices where possible~~.

5. **Assumption 5: Security of Underlying Infrastructure**
   It is assumed that the underlying infrastructure provided by Supabase (database, authentication, storage) and any third-party services (SendGrid, Twilio) maintain adequate security measures to protect against common external threats (e.g., DDoS attacks, unauthorized infrastructure access). The project team is responsible for configuring these services securely but relies on the provider's security posture for the infrastructure itself.

5 Failure Mode and Effect Analysis (FMEA)

This section performs a Failure Mode and Effect Analysis (FMEA) to systematically identify potential failures within the MES-ERP system, their causes, effects, and severity, and to propose mitigation strategies. The Risk Priority Number (RPN) is calculated as Severity (S) x Occurrence (O) x Detection (D), where scales are typically 1-10 (1=low, 10=high).

Table 2: FMEA for MES-ERP

Failure Mode	**Potential Cause(s)**	**Potential Effect(s)**	**S**	**O**	**D**	**RPN**	**Recommended Actions / Mitigations**
Unauthorized access to financial data	Weak passwords, compromised credentials, RBAC misconfiguration, session hijacking	Data breach, privacy violation, financial fraud, loss of trust. (Ref: SSR-2)	9	3	4	108	Enforce strong passwords, MFA (future), regular security audits, secure session management, implement strict RBAC checks on all sensitive endpoints.
Incorrect reimbursement amount processed	OCR error, User data entry error, Backend calculation bug	Financial loss for MES or user, incorrect budget tracking, user dissatisfaction.	7	4	5	140	Implement front-end validation for amounts, require user confirmation of OCR results, add backend validation rules, unit tests for calculation logic.
System downtime during peak usage	Insufficient server resources (CPU/RAM), Database connection pool exhaustion, Unoptimized code/queries	Users unable to submit/approve requests, operational delays, frustration.	6	5	3	90	Load testing (Test 7, Test 9), server resource monitoring, database connection pooling tuning, code optimization (caching, indexing).

Continued on next page

Continued on next page

Failure Mode	Potential Cause(s)	Potential Effect(s)	S	O	D	RPN	Recommended Actions / Mitigations
Failure to send status notification (Email/SMS)	Incorrect user contact info, SendGrid/Twilio API failure or misconfiguration, Network issues	User unaware of request status, delays in process, confusion. (Ref: SSR-1, indirect)	5	4	4	80	Validate contact info on input, implement retry mechanisms for API calls, robust error logging for notification service, provide in-app status tracking as primary method.
Data corruption in budget/expense tables	Race conditions during concurrent updates, Software bug during save, Database hardware/software failure	Incorrect financial reporting, inability to track budget accurately, audit failures. (Ref: SSR-3)	8	2	5	80	Use database transactions for updates, implement locking mechanisms where needed, regular automated backups, data integrity checks.
Privilege Escalation	Flaw in RBAC logic, Bug allowing users to modify their own roles/permissions	User gains unauthorized access to sensitive data or functions (e.g., approving own requests, viewing all budgets). (Ref: SSR-2)	10	2	4	80	Rigorous testing of permission checks (Test 8), code reviews focused on access control logic, limit user ability to self-modify roles, regular permission audits by admin.
OCR Receipt Processing Error	Poor image quality, Non-standard receipt format, OCR engine limitations	Incorrect amount extracted, requires manual correction, slows down submission process.	4	6	3	72	Provide clear guidelines for receipt images, allow manual override of OCR amount, potentially explore alternative OCR services if accuracy is consistently low.

6 Safety and Security Requirements

SSR-1: There must be comprehensive logging and monitoring of access ~~to reimbursement information,~~ and modifications to critical financial data (reimbursement requests, budget allocations), including tracking actions such as data ac-

cess, modification, and deletion with associated user IDs and timestamps.
Rationale: Monitoring access helps in identifying and responding to unauthorized access attempts or suspicious activities, providing an audit trail ~~that enhances data security.~~ essential for financial accountability and forensic analysis if issues arise.

SSR-2: There must be role based access control ~~to restrict~~ enforced consistently across the application (UI and API) to restrict access to ~~reimbursement and~~ financial data and functionalities based on defined user roles and group memberships.
Rationale: Only authorized personnel~~, such as administrators and designated financial officers,~~ should have access to sensitive ~~reimbursement~~ information and actions. This control ensures confidentiality ~~and limits~~ , prevents unauthorized modifications, and limits the scope of potential damage from compromised accounts. ~~access to sensitive financial information.~~

SSR-3: There must be data encryption for sensitive information such as user credentials, ~~financial data, and reimbursement requests.~~ API keys, and potentially sensitive financial details (like bank account numbers if collected for direct deposit), both in transit (using HTTPS) and at rest (using database-level encryption where appropriate).
Rationale: Encryption protects sensitive data from unauthorized access ~~and ensures that data is secure during transmission and storage.~~ if the underlying storage or network is compromised.

7 Roadmap

7.1 Immediate Implementation (During Capstone)

The following safety and security requirements will be implemented during the capstone project to ensure a functional and secure platform:

1. **Input Validation and Error Handling**
   - Ensure that user inputs such as reimbursement amounts ~~and receipts~~ are validated on both client and server sides to prevent submission errors ~~and system crashes~~ , common injection attacks, and data inconsistencies.
2. **Authentication and Access Control (SSR-2)**
   - Implement Supabase Auth login system with role-based access control (RBAC) to restrict ~~sensitive financial data and operations~~ access based on assigned permissions.
3. **Data Encryption and Security (SSR-3)**

- Encrypt sensitive data (like passwords via Supabase Auth, potentially API keys) and ensure secure connections ~~between client and server~~ (HTTPS) to protect against unauthorized access.

4. **Audit Logs (SSR-1)**
   - Set up audit logs in the database to track user actions such as submissions, approvals, and modifications to key financial records.
5. **Server Monitoring and Reliability**
   - Implement basic application-level server monitoring and ensure regular database backups (via Supabase) to ~~ensure system availability and~~ prevent data loss.

7.2 Deferred for Future Implementation

The following will be considered for future updates beyond the capstone project timeline:

1. **Multi-factor Authentication (MFA)**
   - Future versions ~~will consider adding~~ should implement MFA via Supabase Auth to add additional account security, especially for admin roles.
2. **Advanced Fraud Detection**
   - Potentially implement machine learning algorithms or rule-based systems to detect potentially fraudulent ~~reimbursement requests~~ patterns (e.g., duplicate submissions, unusual amounts).
3. **Formal Security Penetration Testing**
   - Conduct professional penetration testing to identify vulnerabilities not caught by automated scans or internal reviews.

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?
   Sufyan: What went well for me during this deliverable is that we were thorough enough in our SRS that I did not have much extra needed to be added to section 6 of the Hazard Analysis.
   Omar: What went well for me during this deliverable was the clarity towards the answers we needed to do. It was very clear what a hazard analysis document should include, that plus the help from the TA made it very easy to know all the potential hazards are project may include.
   Taaha: what went well while writing the deliverable was that all the questions were understandable and were able to be completed with minimal help from TA.
   Rachid: What went well for me this deliverable was that the hazard analysis questions were extremely clear. The SRS allowed me to have a strong understanding of the requirements of the project which allowed me to easily answer all the questions. **Housam**: I found it helpful to look back at the SRS, which provided a solid foundation for defining clear, relevant assumptions related to the project.

2. What pain points did you experience during this deliverable, and how did you resolve them?
   Sufyan: I did not have any pain points during this deliverable.
   Omar: A pain point I had for this deliverable was thinking of the scope of the hazard document, I was not too sure if I needed to include certain elements (building safety) for our project, but after some clarification with the TA it was easy to know.
   Taaha: No pain points during deliverable.
   Rachid: I had no pain points working on this deliverable. **Housam**: Setting boundaries for assumptions without repeating parts or being redundant was sometimes hard. I resolved this by reviewing examples and communicating with the team to keep each assumption focused.

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: We did not give much thought to the confidentiality of the data nor how roles should have varying levels of access. We thought we had that all covered until our meeting with the TA where this discussion was brought to light.

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?
   Team: One is data security risk. Sensitive data especially financial information in our case should not be leaked to the wrong people. There could be legal consequences if this were to happen. Another is system downtime. If the system is down, users will not be able to submit reimbursement requests, delays in financial processing will occur, and this will lead to user dissatisfaction.