Software Requirements Specification for MES-ERP

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

February 10, 2025

Contents

Revision History

Date	Version	Notes
10/11/2024	Revision 0	Initial draft of SRS

1 Purpose of the Project

1.1 User Business

The McMaster Engineering Society is looking to develop a finance and accounting system that will streamline the financial operations of 60 student groups. Due to recent struggles managing and keeping track of reimbursement requests, many students must wait large wait times during the process. The implementation of our solution will allow the MES to provide users with adequate tracking of reimbursement requests made as well as reduce the loss of requests and lower wait times.

1.2 Goals of the Project

Currently, the MES is losing track of reimbursement requests made resulting in large wait times and loss of money. We aim to provide a solution that effectively manages and organizes the large throughput of reimbursement requests sent. The effectiveness of our solution will be measured based on how well it reduces wait times as well as how many requests are lost. Additionally, we plan on implementing custom budget creation and tracking for all users. Through this, users will be able to effectively create and manage budgets for different categories and all transactions will be tracked in real time. This would allow users to gain financial oversight, ultimately preventing going over budget. Custom budgets should be categorized correctly nine times out of ten. The system should also reduce reimbursement turnover time for users, ultimately improving the claim process. Finally, the final solution should aim to be easy for users to understand and use. There should be an intuitive way to create reimbursement requests and if required, users will be provided extra instructions where needed.

2 Stakeholders

2.1 Client

The client for this project is the McMaster Engineering Society (MES). The MES financially supports 60 different student groups and numerous individuals throughout the year through a reimbursement model. The society is responsible for approving the final product and ensuring it meets the needs of

the student groups. The MES invests in this platform to streamline financial processes, reduce administrative overhead, and improve user experience for students and administrators. Roles and responsibilities: Maintain the server that will host the tool. Process and complete the reimbursement requests

2.2 Customer

The primary customers of this platform are the student groups and administrative personal who utilize the reimbursement system. Their roles and responsibilities include:

- **Expense Submission:** Student leaders and members will be submitting expense reports, requiring a user-friendly interface that simplifies the process.
- **Feedback Providers:** Customers will offer feedback on system usability and required features, which is critical for ensuring the platform meets their expectations.
- **User Adoption:** The willingness of these customers to adopt the new system will largely depend on its functionality and ease of use, impacting the overall success of the project.

2.3 Other Stakeholders

Other stakeholders include individuals and groups that may influence or be impacted by the product. Their roles are as follows:

- **Student Group Leaders:** Responsible for managing their group's finances, they will provide insights into the specific needs and challenges they face in the current reimbursement process.
- **MES Administrative Staff:** These individuals assist in processing reimbursements and will need training on the new system to ensure smooth operations.
- **McMaster Administion:** The Faculty of Engineering and CCGE faculty interact with clubs often. They may also want to ensure that our tool follows their principles and standards in terms of rules, flow, cohesiveness.

- **Club members:** Can also submit expense receipts, will want an easy experience.
- **MES IT:** Involved in the integration of the platform with existing systems, they will need detailed documentation and support for maintenance.

2.4 Hands-On Users of the Project

Hands-on users are the individuals who will directly interact with the platform. Their characteristics include:

- **Student Leaders:** Typically aged 18-25, they manage club finances, possess an intermediate amount of technological experience, and are familiar with digital forms and spreadsheets. They would prioritize an intuitive interface that simplifies expense reporting.
- **Administrators:** Members of the MES staff that are aged 18-25, these users manage the overall reimbursement process, have intermediate technological experience, and require efficient tools for tracking and reviewing submissions.
- **Financial Administrators:** Experienced professionals (aged 30+) with an advanced understanding of financial compliance, they need robust reporting features to analyze and audit transactions effectively.

2.5 Personas

The development of personas helps to understand user needs more concretely. Examples include:

- **Alex, the Student Leader:** 20 years old, manages a student club. Familiar with Google Forms and spreadsheets, Alex seeks efficiency in submitting reimbursement requests and values features that save time, such as automated data entry.
- **Jamie, the MES Administrator:** 21 years old, processes reimbursements for multiple student groups. Jamie values accuracy and compliance and prefers tools that provide a clear overview of submissions and their statuses.

- **James, the VP Finance:** 22 years old, Vice President of Finance for the MES. James values comprehensive financial oversight and strategic planning, and prefers tools that provide high-level summaries and detailed financial reports.
- **Jordan, the Club Member:** 19 years old, a member of a student club who occasionally submits reimbursement requests. Jordan values simplicity and ease of use, and prefers a straightforward interface for submitting and tracking requests.

2.6 Priorities Assigned to Users

Understanding user priorities is critical for product success:

- **Key Users:** Student leaders are crucial to the project's success. They must be able to submit expenses easily and effectively in order for this project to be successful. Their requirements will take precedence due to their direct interaction with the system.
- **Secondary Users:** MES Administrators who facilitate the reimbursement process are equally as important, but their needs will be prioritized lower than those of key users.
- **Unimportant Users:** Casual users with minimal interaction, such as infrequent users of the system, will have their requirements considered last, as they do not impact the product's long-term success.

2.7 User Participation

User participation is vital for gathering accurate requirements:

- **Requirements Gathering:** Key users will participate in interviews and surveys to provide insights into their needs and expectations. We will need to document how a typical expense and reimbursement occours, from the time it is submitted into the current expese system to the final payout of the expense to the user. This will provide a clear overview of how the current system works, what the roadblocks are and how we can optimize the tool to the needs of the users.

- **Usability Testing:** Hands-on users will be invited to participate in usability testing sessions to ensure the platform meets their requirements.
- **Regular Check-Ins:** Scheduled meetings with key users will provide ongoing opportunities for feedback and validation throughout the development process.

2.8 Maintenance Users and Service Technicians

Maintenance users play a vital role in the long-term success of the platform:

- **MES IT Staff:** Responsible for system updates and troubleshooting, they will require detailed documentation on the platform's functionalities to effectively manage its operation. These users ensure the platform runs smoothly, addressing any technical issues that arise and implementing updates as necessary.

3 Mandated Constraints

3.1 Solution Constraints

1. The app must handle the throughput of the 60 student groups to the MES without trouble.
2. The app must inform student groups the status of their request.
3. The app must notify the appropriate parties of the MES for any new requests made.
4. The app must allow student groups to upload receipts.
5. The app must be able to read the receipts that are uploaded by the student groups.
6. The app must be up and running at all times, if not a message must be shown signifying why the app is offline.

3.2 Implementation Environment of the Current System

1. There is no current implement of any current system.
2. The new system will be hosted on a laptop provided by the MES, this laptop will have access to Windows 10, and will have above average specs (eg, 4gb of RAM, RTX 2080).

3.3 Partner or Collaborative Applications

1. There is no current implement of any current system.

3.4 Off-the-Shelf Software

1. We will use an API for scanning receipts, A specific API has not been chosen yet.

3.5 Anticipated Workplace Environment

1. The workplace environment will be on our specific dev machines, we will use Git (A version control software) to collaborate and share code with each other.

3.6 Schedule Constraints

1. The final demonstration will be April 2nd, which will be our final design constraint.

3.7 Budget Constraints

1. The maximum budget of this would be $750.

3.8 Enterprise Constraints

1. The product and code will be made entirely available to the MES.

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

- **MES**: McMaster Engineering Society. The student organization responsible for overseeing financial and academic matters for engineering students at McMaster University.
- **Audit Log**: A record of all actions performed within the system, used for tracking who accessed or modified data and when these actions took place.
- **Club Financial Manager**: A designated member of a club responsible for managing that club's financial records and submitting reimbursement requests on behalf of the club.
- **Budget Allocation**: The amount of funding designated for specific clubs or activities by the MES team and is used to track and control spending.
- **Vendor Contracts**: Agreements between MES and external suppliers for goods or services provided to the engineering society or its member clubs.
- **Approval Workflow**: The process that a reimbursement request goes through before getting final approval.
- **Financial Dashboard**: A feature of the system that provides an overview of the current financial standing, showing budget tracking, reimbursement status, and pending requests.
- **CI/CD**: Continuous Integration/Continuous Deployment. A software development practice that automatically tests and deploys code changes to ensure the system is always in a working state.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

- There is no current application built for this project.

- The project will be built primarily in JavaScript and React.
- There are 60 student groups to account for.
- The current process for submitting checks is manual, which takes far too long.
- The input should be generalized, with no differentiation between student groups.
- The application is intended to be created with future growth in mind.

5.2 Business Rules

- Users will be notified when their receipts are reimbursed.
- The site will notify the relevant parties MES when a request comes through.
- Users will be notified of an expected deadline for when the requests will be fulfilled.
- Users will be notified of any maintenance times the site will need.
- All financial records will be recorded and saved in the backend of the application

5.3 Assumptions

- Student groups and the MES are in contact and know the ground rules of what to expect from purchases.
- We are expecting the MES to provide us with a server to run our software.
- The system will be coded in JavaScript and NextJS.
- The system will be built from scratch.
- The MES is expected to provide us with data for test cases.

- Internet is expected to be provided, as well as a suitable system with enough RAM/storage for the app.
- Student groups will use the application for further validation testing.

6 The Scope of the Work

6.1 The Current Situation

Currently, the McMaster Engineering Society (MES) manages financial operations for over 60 student groups. The existing process for handling reimbursement requests is inefficient and manual. This is due to the lack of a consolidated platform to manage this reimbursement workflow which leads to delays, loss of reimbursement requests, and difficulty in tracking budgets and expenses.

6.2 The Context of the Work

The goal of the MES-ERP project is to streamline the financial processes of the MES by providing a consolidated platform that combines reimbursement tracking, budget management, and financial reporting. The scope of the work includes allowing users to submit and track reimbursement requests, manage budgets, and generate financial reports. In addition, the system will include an audit trail of all transactions to aid the MES during audits.

6.3 Work Partitioning

- **Reimbursement Submission Module:** A feature that allows users to submit receipts and financial documents for reimbursement.
- **Reimbursement Review and Approval:** A feature to allow MES administrators to review, approve, or reject reimbursement requests.
- **Audit and Compliance Module:** This feature will ensure that all transactions are logged and auditable, complying with MES financial policies.
- **Budget and User Dashboards:** A feature that provides users with an overview of their budgets, expenses, and pending requests.

6.4 Specifying a Business Use Case (BUC)

BUC 1 - Submitting a Reimbursement Request:

- **Precondition:** The user must have access to the system and necessary documentation for submission.
- **Trigger:** A student group needs reimbursement for an expense.
- **Steps:**
  1. The user logs into the MES-ERP system.
  2. The user navigates to the 'Submit Reimbursement' section.
  3. The user fills out the required fields, attaches relevant receipts or documents, and submits the form.
- **Postconditions:** The reimbursement request is logged in the system and sent for review by the MES administrators.

BUC 2 - Review Reimbursement Requests:

- **Precondition:** A request has been submitted by a student group.
- **Trigger:** MES administrators receive a notification of a new reimbursement request.
- **Steps:**
  1. The MES administrator logs into the system.
  2. The MES administrator reviews the reimbursement request and attached documents.
  3. The MES administrator updates the status of the request in the system with either an approval or rejection with feedback.
- **Postconditions:** The reimbursement request is either approved and sent for payment or rejected with feedback provided to the user.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

The MES-ERP system will handle various types of financial and user-related data, organized into distinct entities to support the core functionalities of the platform. Below is an overview of the key entities and their relationships:

- **User Entity:**
  - **Attributes:** UserID, Name, Email, Group, Role (Student Group Leader, Administrator, Financial Officer), Login Credentials, Phone Number, Reimbursement Choice
  - **Relationship:** A user can submit many reimbursement requests and manage budgets.
- **Reimbursement Request Entity:**
  - **Attributes:** RequestID, Timestamp, Email Address, SubmitterID (links to UserID), Identification, Full name, Phone Number, Email, Role, Budget Line, Approved Individual, Team name, Conference or competition name/type, Head Delegate, Amount, Status (Submitted, Approved, Rejected, Paid), Attached Receipts, Reimbursement Choice, Payment timeframe
  - **Relationship:** Each reimbursement request is submitted by one user but can be reviewed by multiple administrators.
- **Budget Entity:**
  - **Attributes:** BudgetID, GroupID (identifies the student group), Total Amount Allocated, Amount Spent, Category (e.g., Event, Travel, Supplies), Additional group information
  - **Relationship:** A student group manages one budget per financial year, and each budget can have multiple expense reports linked to it.
- **Group Entity:**
  - **Attributes:** ID, Name, Created at, Created by

  - **Relationship:** A group will identify a bunch of individuals to a specific group which they take part of

- **Permission Entity:**
  - **Attributes:** ID, Name
  - **Relationship:** The permission entity will allow a user to have different roles on the site based off their permissions

- **Role Permission Entity:**
  - **Attributes:** ID, Permission ID
  - **Relationship:** This links role IDs to permission IDs

- **Role Entity:**
  - **Attributes:** ID, Name
  - **Relationship:** This creates roles for a user.

7.2 Data Dictionary

Insert your content here.

8 The Scope of the Product

8.1 Product Boundary

The product boundary defines the scope of the McMaster Engineering Society (MES) Custom Financial Expense Reporting Platform. The product boundary distinguishes between functionalities to be automated by the platform and those that will remain manual or handled by other existing systems.

- **Automated Functions:**
  - Streamlining reimbursement requests for student groups.
  - Saving repetitive information.
  - Sending SMS and automated emails for updates and notifications to users.

  - Processing invoices, receipts, payment notices, and remittance advice effectively.

- **Manual Functions:**
  - Initial data entry for historical records and previous expense submissions.
  - User training and support for navigating the new platform.
  - Handling of any non-digital submissions or documents.
  - Submitting reimbursement receipts.
  - Handling payment requests.
  - Creating roles and groups.
  - Managing users and groups.
  - Creating budget lines.

8.2 Product Use Case Table

The Product Use Case Table summarizes the specific functionalities (Product Use Cases, or PUCs) that the platform will support. This table identifies the main actors involved and outlines the input and output data associated with each use case.

Table 1: Product Use Case Summary Table

PUC No	**PUC Name**	**Actor/s**	**Input & Output**
1	Submit Reimbursement Request	Student Leader	Reimbursement Details (in)
2	Review Reimbursement Requests	Administrator	Reimbursement Request (in)
3	Track Payment Requests	Administrator	Payment Request (in), Paym
4	Process Funding Applications	Student Leader	Funding Application (in), Ap
5	Save Repetitive Information	Student Leader	Repetitive Info (in), Confirm
6	Generate Reports	Financial Auditor	Report Criteria (in), Genera
7	Send Notifications	System	User Notification Preferences
8	Manage Audit Trails	Financial Auditor	Audit Criteria (in), Audit Tr

8.3 Individual Product Use Cases (PUCs)

This section details the individual product use cases listed in the Product Use Case Table. Each use case outlines the scenario, the actors involved, and the interactions within the system.

8.3.1 PUC 1: Submit Reimbursement Request

Actors: Student Leader
Scenario: A student leader submits an expense reimbursement request through the platform.

- The student logs into the system.
- They navigate to the reimbursement section and fill out the required fields (e.g., amount, purpose, and attaching receipts).
- Upon submission, the request is sent for review to the administrator.
- Confirmation of submission is displayed on the screen.

8.3.2 PUC 2: Review Reimbursement Requests

Actors: Administrator
Scenario: An administrator reviews submitted reimbursement requests.

- The administrator logs into the system.
- They access the list of pending reimbursement requests.
- The administrator reviews the details and either approves or rejects the request.
- An automated notification is sent to the student leader regarding the decision.

8.3.3 PUC 3: Track Payment Requests

Actors: Administrator
Scenario: The administrator tracks the status of payment requests.

- The administrator logs into the system.

- They navigate to the payment request section.
- The status of each request is displayed (e.g., pending, processed, completed).

8.3.4 PUC 4: Process Funding Applications

Actors: Student Leader
Scenario: A student leader processes an application for intramural funding.

- The student accesses the funding application section.
- They fill out the required information and submit it for approval.
- The system sends a notification confirming the application submission.

8.3.5 PUC 5: Save Repetitive Information

Actors: Student Leader
Scenario: The student saves repetitive financial information for future use.

- The student navigates to the settings section.
- They enter and save their personal information.
- The information is stored for future reimbursement requests.

8.3.6 PUC 6: Generate Reports

Actors: Financial Auditor
Scenario: A financial auditor generates financial reports for review.

- The auditor selects criteria for the report (e.g., date range, type of expenses).
- The system generates the report and displays it for download.

8.3.7 PUC 7: Send Notifications

Actors: System
Scenario: The system sends notifications to users based on their preferences.

- Upon important updates (e.g., approval of reimbursement), the system triggers notifications.
- Notifications are sent via SMS or email as specified by the user.

8.3.8 PUC 8: Manage Audit Trails

Actors: Financial Auditor
Scenario: The financial auditor reviews audit trails to ensure compliance.

- The auditor accesses the audit trail section in the platform.
- They filter results based on date and type of activity.
- A detailed report is generated to review all activities related to financial transactions.

9 Functional Requirements

9.1 Functional Requirements

- **Requirement Number**: 001
- **Description**: The system must allow club financial managers to submit reimbursement requests.
- **Rationale**: Ensures that clubs can request refunds for approved expenses.
- **Fit Criterion**: A club financial manager must be able to submit a reimbursement request successfully, which should appear in the MES staff's pending approval list.
- **Priority**: High
- **Originator**: Club Financial Manager

- **Requirement Number**: 002
- **Description**: The system must allow MES staff to approve or reject reimbursement requests.
- **Rationale**: Ensures proper management of reimbursement requests.
- **Fit Criterion**: A reimbursement request must be updated with the approval or rejection status when MES staff submits their decision.
- **Priority**: High
- **Originator**: MES Finance Team

- **Requirement Number**: 003
- **Description**: The system must send a notification to MES financial managers when their reimbursement requests are approved or rejected.
- **Rationale**: This ensures that club financial managers are informed of the status of their reimbursement requests.
- **Fit Criterion**: A notification must be sent within a short amount of time after approval or rejection of a reimbursement request to the person who requested it. In case of approval, this will include some form of receipt for the user confirming their reimbursement.
- **Priority**: Medium
- **Originator**: MES Finance Team

- **Requirement Number**: 004
- **Description**: The system must allow administrators to access all financial records, audit logs, and user activity.
- **Rationale**: Administrators need full access for auditing and tracking user activity.
- **Fit Criterion**: Administrators must be able to view any financial records or user actions.

- **Priority**: High
- **Originator**: System Administrators

10 Look and Feel Requirements

10.1 Appearance Requirements

1. Animations should be kept minimal and used only when necessary to avoid distractions.
   Rationale: Minimal animation improves the user experience through feedback, without being very distracting, hence keeping the focus on the functionality of the platform.

10.2 Style Requirements

1. The system must not have elements that depict violence, nudity or language that is discriminatory, vulgar, or derogatory.
   Rationale: The system should not offensive or inappropriate to users.

2. There will be no visually unpleasing elements such as colour combinations that strain the eyes.
   Rationale: There should not be any elements that make the application harder to read.

3. The platform's tone should remain professional and formal.
   Rationale: A professional tone builds trust with users by emphasizing the seriousness of the platform, particularly given its purpose in handling financial workflows.

4. Form fields should be simple and clear, with error states highlighted in red to help users easily identify and correct mistakes.
   Rationale: Well-designed form fields minimize confusion during data entry, while clear error messages guide users through resolving issues.

5. Clear and intuitive icons should be used for navigation and key actions, such as submitting expenses or reviewing approvals.
   Rationale: Icons provide immediate visual cues that reduces the need for text explanations.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

1. The solution's UI elements such as buttons and forms will adhere to consistent styling in order to allow for an easy to understand user experience.
   Rationale: Maintaining consistency allows user to easily understand the system while interacting with the UI.

2. The solution will have an organized main menu that will allow users to navigate quickly between numerous tabs.
   Rationale: Having a platform that is easy to use requires a good navigation menu in order to reduce time taken for users when finding the tab they wish to visit.

11.2 Personalization and Internationalization Requirements

1. Users will have the ability to create an account and update their information to their liking.
   Rationale: This will allow users to save time for any future reimbursement requests.

11.3 Learning Requirements

1. Users will be provided instructions if they go to a tutorial page.
   Rationale: Walking the user through the reimbursement request process would effectively and efficiently teach the user on how to create requests in the future, saving time and minimizing confusion.

11.4 Understandability and Politeness Requirements

1. Any error messages on the platform will provide information regarding why the error is showing.
   Rationale: Having detailed error messages will help users to quickly troubleshoot the problem and fix their mistakes.

11.5 Accessibility Requirements

1. The product shall have an option for adjusting colour themes and will avoid colour combinations that are not supported for those with colour blindness.
   Rationale: Allowing customizable colour themes would accomodate for users who are colour blind.

12 Performance Requirements

12.1 Speed and Latency Requirements

The platform must ensure a responsive experience for users to maintain workflow efficiency and user satisfaction. Specific speed and latency requirements include:

- Any interface between a user and the automated system shall have a response time that ensures a smooth user experience.
- The response shall be fast enough to avoid interrupting the user's flow of thought during the submission and review processes.
- The system shall process and save repetitive information quickly to enhance user experience.
- The product shall update status parameters in a timely manner to reflect changes accurately.
- The system shall generate reports promptly to facilitate timely decision-making by financial auditors.

12.2 Safety-Critical Requirements

Given the financial nature of the platform and its handling of sensitive information, safety-critical requirements focus on data security and user privacy. These requirements include:

- The platform shall comply with relevant data protection and user privacy standards.

- The system shall ensure that no sensitive user data is exposed during the submission or review of expense reports.
- User authentication processes must prevent unauthorized access to financial information.
- Regular security audits must be performed to ensure compliance with established safety standards.

12.3 Precision or Accuracy Requirements

To ensure accurate financial reporting and user trust, precision requirements for monetary transactions and data entries are essential. These requirements include:

- All monetary amounts shall be accurate to a standard level of precision.
- The system shall ensure the accuracy of expense calculations to prevent discrepancies in reporting.
- The product shall validate and display error messages for any inaccurate entries immediately upon submission.

12.4 Robustness or Fault-Tolerance Requirements

The platform must demonstrate robustness to ensure continuous operation even during unexpected failures or high usage. Specific robustness requirements include:

- The product shall continue to operate in local mode whenever it loses its connection to the central server, allowing users to save entries temporarily.
- The system shall provide emergency operation capabilities in case of power or network failures, ensuring that critical processes can still be completed.
- Automatic backups of all financial data shall occur regularly to prevent data loss.

12.5 Capacity Requirements

The platform must accommodate a significant number of users and transactions, ensuring it can handle the expected load. These requirements include:

- The product shall cater to a large number of simultaneous users during peak usage hours, ensuring that the system remains responsive.
- The platform must handle a high volume of expense report submissions without degradation in performance.
- The database shall be capable of storing financial records for an extended period, accommodating historical data for auditing purposes.

12.6 Scalability or Extensibility Requirements

To meet future growth and increased demand, the platform must be designed for scalability. Key requirements include:

- The product shall be capable of scaling to accommodate a growing number of users and transactions over time.
- The platform must handle increasing transaction volumes as the organization grows.
- The system architecture should be modular to facilitate the addition of new features without requiring significant redesign.

12.7 Longevity Requirements

The longevity of the platform is crucial for ensuring a return on investment and sustainability. Requirements include:

- The product shall be expected to operate effectively for an extended period within the defined maintenance budget.
- All components of the system must be built with durability and support in mind, with the expectation of routine maintenance.

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

1. The system will be in an office room.
2. The system will be located in the McMaster Campus.
3. The system will be useable in any natural environment for a laptop or desktop.

13.2 Wider Environment Requirements

1. The system will make the MES limit the use of Google Forms/Excel for reimbursements and adopt the application for these requests.

13.3 Requirements for Interfacing with Adjacent Systems

1. The app will be useable on a desktop and laptop.
2. The app will be running on Windows, Mac and Linux.
3. The app will work on the last 4 versions of Chrome, Firefox and Edge.

13.4 Productization Requirements

1. N/A (this product is not for sale)

13.5 Release Requirements

1. The product will follow an agile software development lifestyle.
2. The product will only have maintenance is a crucial bug is found.
3. Products must be tested before each release.
4. A bug found will be made into a ticket which will be given a deadline to solve by, and then a new release build will be made.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

1. If any changes to the reimbursement process is made in the future or if the number of student groups increase, the software will be easily extensible to allow for future development.
   Rationale: Keeping the program extensible will make the program easy to maintain and add features later on in the future.

2. Any general maintenance issues should be resolvable by hired software developers after being provided all necessary documentation.
   Rationale: The system should be well documented to allow for easy understanding of the software.

14.2 Supportability Requirements

1. The solution will be supported on all platforms (Windows, Linux, Mac).
   Rationale: Having a wide range of platforms will allow users from different operating systems to easily use the software.

2. The service shall only run on MES authorized devices. These devices are expected to have at least 4GB ram, a modern CPU, and access to the internet.
   Rationale: The MES requires the solution to be ran on authorized devices due to security reasons.

14.3 Adaptability Requirements

1. Expected to run above Windows 10 or ideally be updated to latest windows/mac software.

2. Expected to be designed for possible expansion (increase in users etc)

3. Expected to run on provided MES laptops and to support all platforms (Mac, Windows, Linux)

15 Security Requirements

15.1 Access Requirements

1. Administrators must have full access to all system functionalities, including creating, modifying, and deleting records for all users, managing user roles, and viewing audit logs and financial reports.

2. MES staff must have access to financial records, reimbursement requests, audit logs, and the ability to approve or reject reimbursement requests.

3. Club financial managers must have access to their respective club's financial records, submission of reimbursement requests, and viewing of their club's budget tracking, but no access to financial data from other clubs.

4. Only administrators and MES staff must be able to access sensitive data such as vendor contracts, budget allocations, and event financials.

5. Club financial managers must not have access to sensitive financial information beyond their own club's data.

6. Administrators and MES staff must be able to modify financial records, approve budgets, and manage payments, while club financial managers can only submit reimbursement requests.

7. All access to information and functions, such as reimbursement approval and budget tracking, must be restricted according to user roles to protect the confidentiality of the financial data.

15.2 Integrity Requirements

1. The system must ensure that all financial data, such as budgets, expense reports, and reimbursement requests, remain accurate and have not been tampered with. This includes preventing inaccurate or unauthorized modifications to any financial records. This is to ensure that submitted data is valid and consistent with the system's requirements.

2. The system shall prevent internal misuse, such as incorrect data entry, through validation rules and error handling mechanisms.

3. To maintain data integrity, the system must perform automated backups of all important financial data. In the event of system failure, data corruption, or accidental deletion, these backups should allow the system to recover with minimal data loss. Recovery processes should be tested regularly to ensure reliability.

4. The system shall maintain detailed logs of all actions taken within the platform, such as data modifications, approvals, or deletions. These logs will be immutable and can be used to restore data or identify the source of an error if data becomes corrupted or tampered with.

15.3 Privacy Requirements

1. The system must comply with all relevant privacy laws and regulations to ensure that personal data is handled in accordance with these laws.

2. The system must obtain consent from users before collecting personal information. Users must have the ability to revoke data collection consent at any time, and the system must stop processing or storing the data on the user.

3. Users must be able to view, edit, or request corrections to their personal data stored in the system.

4. The system must notify users of any changes to its information or privacy policy. Users should have the opportunity to review and give feedback or consent to the changes before any further processing of their data.

5. The system must protect all personal information with encryption or other security measures to ensure it cannot be accessed, modified, or deleted by unauthorized users.

15.4 Audit Requirements

1. The system must retain detailed records of all user transactions, including financial submissions, approvals, modifications, and deletions, ensuring that all actions can be traced back to the responsible user.

2. The system must record login attempts, both successful and unsuccessful, to provide a complete history of user access to the system.

3. Access to the audit logs must be restricted to authorized users, such as administrators and auditors, ensuring confidentiality of the audit data.

4. The system must ensure that audit data is protected from unauthorized access or tampering, using security measures such as encryption and access control.

15.5 Immunity Requirements

1. The system must ensure that all software components and libraries used in the product are regularly updated with the latest security patches to reduce the risk of infection by known vulnerabilities.

16 Cultural Requirements

16.1 Cultural Requirements

1. The application must be available in English.
   Rationale: English is the primary language spoken by citizens in Ontario and thus shall be the primary method of communication.

2. The application shall use British spelling.
   Rationale: As the application is for use within McMaster University, the predominant spelling method is using British spelling.

3. The system must be designed to be inclusive and usable by people from diverse cultural, linguistic, and social backgrounds. This includes ensuring gender-neutral language and avoiding culturally biased terms.
   Rationale: MES is a diverse community composed of students and staff from various cultural and linguistic backgrounds. The platform must reflect this diversity by avoiding assumptions about users' cultural norms or preferences.

17 Compliance Requirements

17.1 Legal Requirements

The development of the McMaster Engineering Society (MES) Custom Financial Expense Reporting Platform must adhere to various legal requirements to ensure compliance with applicable laws and regulations.

- **Data Protection Compliance:** Personal information shall be implemented in a manner that complies with the Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada. This includes ensuring that user data is collected, stored, and processed securely, with consent obtained from users.
- **Financial Regulations:** The platform must comply with the Financial Transactions and Reports Analysis Centre of Canada (FINTRAC) regulations to prevent money laundering and the financing of terrorist activities.
- **Accessibility Compliance:** The product shall meet the requirements set forth by the Accessibility for Ontarians with Disabilities Act (AODA), ensuring that all users, regardless of their abilities, can access and utilize the platform effectively.
- **Fit Criterion:** A legal opinion must be obtained confirming that the product does not violate any applicable laws or regulations.

17.2 Standards Compliance Requirements

To ensure the quality and reliability of the platform, it must comply with various industry standards. These standards are critical for avoiding delays in deployment and ensuring user trust.

- **Information Security Standards:** The product shall comply with the ISO/IEC 27001 standard for information security management systems, ensuring a systematic approach to managing sensitive company and customer information.
- **Software Development Standards:** The development process shall follow Agile methodologies, specifically adhering to the Agile Manifesto principles, which emphasize flexibility and collaboration.

- **Fit Criterion:** The appropriate standard-keeper (e.g., certification bodies or industry organizations) must certify that the platform adheres to the specified standards throughout its development and operation.

18 Open Issues

- **Issue Number**: 001
- **Cross-reference**: Affects Functional Requirement 003.
- **Summary**: The team is deciding if notifications should be sent through email, SMS, or through in-app alerts.
- **Stakeholders Involved**: Project Team, MES IT Team.
- **Action**: Perform a survey with stakeholders to determine the preferred notification method.
- **Resolution**: Expected decision within one week.

- **Issue Number**: 002
- **Cross-reference**: Affects Functional Requirement 004.
- **Summary**: It is not clear how long audit logs should be stored in the system.
- **Stakeholders Involved**: Project Team, MES IT Team.
- **Action**: Stakeholders will be meeting to determine data retention policies.
- **Resolution**: Expected decision within two weeks.

- **Issue Number**: 003
- **Cross-reference**: Affects Functional Requirement 002.
- **Summary**: It is not clear whether some requests should get multiple approvals from multiple people. For example, a reimbursement request showing high levels of spending needing to be reimbursed.

- **Stakeholders Involved**: Project Team, MES Finance Team.
- **Action**: Stakeholders will be meeting to define thresholds and cases for multiple approvals.
- **Resolution**: Expected decision within two weeks.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

1. There is no pre-existing solution for this problem, this problem is MES specific and has never been done before.

19.2 Reusable Components

1. The main library we will use is ReactJS, there will be no other components that can be used/reused.

19.3 Products That Can Be Copied

1. Any budgeting apps UI can be copied and taken over for the front end of the application. With that frontend along with some changes it can be applied to our app.

20 New Problems

20.1 Effects on the Current Environment

The introduction of the McMaster Engineering Society (MES) Custom Financial Expense Reporting Platform will have several effects on the current implementation environment:

- **Integration with Existing Processes:** The new platform will replace the cumbersome process involving Google Forms, PDFs, and spreadsheets. This transition may initially disrupt the workflow for users accustomed to the old system, necessitating training and adaptation.

- **Data Migration:** Historical data from existing systems must be migrated to the new platform, which could lead to data integrity issues if not managed correctly.
- **Resistance to Change:** Existing users may exhibit reluctance to adopt the new platform, impacting initial usage rates and acceptance.
- **Preservation of Critical Operations:** It is crucial that the new product does not interfere with ongoing financial operations, such as payment processing or compliance reporting.

The goal is to identify potential conflicts early to minimize disruption during implementation.

20.2 Effects on the Installed Systems

The new platform will interface with existing systems at MES, and understanding these interactions is critical:

- **Compatibility:** Ensure that the platform is compatible with existing hardware and software, minimizing the risk of conflicts or failures during operation.
- **Legacy System Considerations:** Users may need access to legacy systems during the transition phase, requiring well-defined procedures for accessing both systems without confusion.

A model depicting these interfaces, along with data dictionary definitions, will help clarify these relationships.

20.3 Potential User Problems

The transition to a new system may lead to potential user problems, which should be anticipated:

- **Loss of Familiarity:** Existing users may struggle with the new interface, leading to frustration and decreased productivity.
- **User Error:** Inexperienced users might make errors during the transition, especially in submitting requests or processing approvals, which could impact financial accuracy.

- **Support Needs:** Increased demand for technical support during the initial rollout phase may overwhelm the support team, leading to delays in assistance.

Identifying these potential issues early allows for proactive measures to be taken.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Several limitations in the anticipated implementation environment may affect the success of the new platform:

- **Server Capacity:** The current server infrastructure may not support the expected load during peak usage times. Upgrading server capacity may be necessary to accommodate the increased demand.
- **Power Requirements:** If the new platform relies on additional hardware or services, ensuring that power capabilities meet these needs is critical to avoid downtime.
- **User Bandwidth:** Users may require sufficient internet bandwidth to utilize the platform effectively, which may not be available in all cases.

Addressing these limitations early in the development process will facilitate a smoother implementation.

20.5 Follow-Up Problems

As the project progresses, several follow-up problems may emerge:

- **Increased Demand:** If the new platform proves popular, there may be an unexpected surge in users, leading to scalability issues.
- **Compliance Challenges:** New legal regulations may emerge after implementation, necessitating additional features or modifications to ensure compliance.

- **Resource Allocation:** Ensuring that the necessary resources, including personnel for support and development, are available will be critical to the ongoing success of the platform.

Identifying these potential follow-up problems allows for strategic planning and resource allocation to mitigate risks.

21 Tasks

21.1 Project Planning

1. The lifestyle we intend to use and develop this software is an Agile software development lifestyle.

21.2 Planning of the Development Phases

Step	Description
1. Requirements Gathering	Stakeholders and the team collaborate to define high-level requirements and user stories. This is an ongoing process, with requirements being added or modified during the development cycle.
2. Sprint Planning	The team selects the user stories or tasks to be completed during the upcoming sprint. A sprint typically lasts 1 to 4 weeks.
3. Design and Architecture	Before starting the implementation, the team defines a general approach for the architecture and design of the system. The design may evolve as new requirements are discovered.
4. Development	The team works on the selected tasks, developing the functionality and features according to the user stories. Developers and testers collaborate closely.
5. Testing	Continuous testing is done throughout the sprint. Unit tests, integration tests, and functional tests are performed to ensure that the software meets the requirements.

6. Review and Retrospective	At the end of the sprint, the team holds a sprint review to showcase the completed work. A retrospective is also held to discuss improvements for the next sprint.
7. Release Planning	The team decides when to release new features to production. This can happen after each sprint or after several sprints, depending on the project's needs.
8. Continuous Feedback	The team gathers feedback from stakeholders and users to refine the product in subsequent sprints, ensuring the software evolves according to user needs.

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

N/A - There are no current requirements to migrate data from the old product to the new product since no existing data will be imported into the new MES-ERP system. Historical reimbursement data from previous systems will be considered as legacy information and will remain in the old system, which will be referenced when necessary.

22.2 Data That Has to be Modified or Translated for the New System

N/A - No existing data will be modified or translated. Should it become necessary to import historical data in the future, this process will be considered during a potential extension of the system, as detailed in Section 24.

23 Costs

The cost estimation for the McMaster Engineering Society Custom Financial Expense Reporting Platform is based on factors such as functional and non-functional requirements, the difficulty of the business use cases, and the expected development effort.

23.1 Labor Costs

The total estimated labor cost is based on the following:

- **Development Hours**: The estimated time required for developing the platform is between 400 to 500 hours. This estimate accounts for the implementation of core features such as reimbursement submissions, approval workflows, and financial tracking and integration with third-party services for sms notifications, and UI/UX design.
- **Testing Hours**: Testing the platform, including unit testing and integration testing, is expected to take 50 to 70 hours. This estimate is based on functional requirements and the need to ensure stability across many use cases.

23.2 Monetary Costs

While the development and testing are measured in hours, the following are expected to have real monetary costs:

- **Server Operating Costs**: The system will be hosted on a local server provided by MES. Although there is no cloud storage cost, running the server will have electricity and maintenance costs. These are expected to be minimal and can be estimated at approximately $10 to $20 per month.

23.3 Additional Considerations

- **Maintenance Costs**: After development, maintenance will be required for system updates, bug fixes, and adding new features. This will likely require an extra 5-10 hours per month.
- **Backup Plan**: An extra 10-20 hours is set aside for unexpected issues or changes in requirements during development.

24 Waiting Room

1. A mobile app version of the platform should be developed for easier on-the-go access to financial tracking and submissions.

Rationale: A dedicated mobile app would improve accessibility for users who need to submit expenses or view budgets while away from their desktops. Although this would be beneficial, it is not critical for the platform's core functionality as responsive design already ensures usability on mobile devices.

2. Implement machine learning-based expense categorization to automatically categorize expenses based on historical data.
   Rationale: Automated categorization of expenses would reduce manual entry and improve accuracy, providing an intelligent system that learns from user inputs. This is a technically complex feature that could be revisited after the platform's initial release.

3. Provide users with customizable dashboards where they can select and prioritize the financial metrics most relevant to them.
   Rationale: Customizable dashboards allow users to personalize their experience and focus on the data that matters most to them. While this is an attractive feature, it can be delayed until after the core dashboard functionalities are developed.

4. Enable multi-language functionality, allowing users to switch between different languages for a more inclusive experience.
   Rationale: MES is a diverse community with users from various linguistic backgrounds. Multi-language support would enhance accessibility and usability for all users. However, this feature is not critical for the platform's initial release and can be considered in future updates.

5. The system should include functionality to import historical reimbursement data from the previous system, allowing all past transactions and records to be accessible within the new MES-ERP platform.
   Rationale: Historical data migration will allow users to reference past financial transactions within the new system, ensuring continuity and easy access to older records. This feature, while not critical to the initial deployment, adds value by consolidating all financial data into a single platform which reduces the need for referencing multiple systems.

25 Ideas for Solution

A solution for this will be a web-based finance and accounting system using JavaScript, NextJS, React, and Git to address the MES's challenges in managing reimbursement requests for 60 student groups. This system will streamline financial operations by allowing students to submit reimbursement requests with modular receipt uploads, real-time status tracking, and automated notifications. It will also feature live ledger tracking, custom budget creation, and support for multiple approval levels, ensuring transparency and reducing delays in request processing. This solution will greatly increase the efficiency of the old system the MES uses (excel files). Typescript will be used as the primary programming language. NextJS will serve as the framework for both front-end and back-end. React will also be used to create a responsive and interactive experience for the app. Git will manage version control, allowing the development team to share code, track changes, and collaborate efficiently without overwriting each other's work.

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?