

# Software Requirements Specification for MES-ERP

Team #26, Ethical Pals

Sufyan Motala

Rachid Khneisser

Housam Alamour

Omar Muhammad

Taaha Atif

April 3, 2025

# Contents

<b>1</b>	<b>Purpose of the Project</b>	<b>vi</b>
1.1	User Business . . . . .	vi
1.2	Goals of the Project . . . . .	vi
<b>2</b>	<b>System Overview</b>	<b>vi</b>
<b>3</b>	<b>Stakeholders</b>	<b>viii</b>
3.1	Client . . . . .	viii
3.2	Customer . . . . .	ix
3.3	Other Stakeholders . . . . .	ix
3.4	Hands-On Users of the Project . . . . .	ix
3.5	Personas . . . . .	x
3.6	Priorities Assigned to Users . . . . .	x
3.7	User Participation . . . . .	x
3.8	Maintenance Users and Service Technicians . . . . .	x
<b>4</b>	<b>Mandated Constraints</b>	<b>xi</b>
4.1	Solution Constraints . . . . .	xi
4.2	Implementation Environment of the Current System . . . . .	xi
4.3	Partner or Collaborative Applications . . . . .	xi
4.4	Off-the-Shelf Software . . . . .	xi
4.5	Anticipated Workplace Environment . . . . .	xi
4.6	Schedule Constraints . . . . .	xii
4.7	Budget Constraints . . . . .	xii
4.8	Enterprise Constraints . . . . .	xii
<b>5</b>	<b>Naming Conventions and Terminology</b>	<b>xii</b>
5.1	Glossary of All Terms, Including Acronyms, Used by Stake- holders Involved in the Project . . . . .	xii
<b>6</b>	<b>Relevant Facts And Assumptions</b>	<b>xiii</b>
6.1	Relevant Facts . . . . .	xiii
6.2	Business Rules . . . . .	xiii
6.3	Assumptions . . . . .	xiii

<b>7</b>	<b>The Scope of the Work</b>	<b>xiii</b>
7.1	The Current Situation . . . . .	xiii
7.2	The Context of the Work . . . . .	xiv
7.3	Work Partitioning . . . . .	xiv
7.4	Specifying a Business Use Case (BUC) . . . . .	xiv
<b>8</b>	<b>Business Data Model and Data Dictionary</b>	<b>xv</b>
8.1	Business Data Model . . . . .	xv
8.2	Data Dictionary . . . . .	xvi
<b>9</b>	<b>The Scope of the Product</b>	<b>xvi</b>
9.1	Product Boundary . . . . .	xvi
9.2	Product Use Case Table . . . . .	xvii
9.3	Individual Product Use Cases (PUCs) . . . . .	xvii
<b>10</b>	<b>Functional Requirements</b>	<b>xvii</b>
10.1	Functional Requirements . . . . .	xvii
<b>11</b>	<b>Look and Feel Requirements</b>	<b>xix</b>
11.1	Appearance Requirements . . . . .	xix
11.2	Style Requirements . . . . .	xx
<b>12</b>	<b>Usability and Humanity Requirements</b>	<b>xx</b>
12.1	Ease of Use Requirements . . . . .	xx
12.2	Personalization and Internationalization Requirements . . . . .	xx
12.3	Learning Requirements . . . . .	xxi
12.4	Understandability and Politeness Requirements . . . . .	xxi
12.5	Accessibility Requirements . . . . .	xxi
<b>13</b>	<b>Performance Requirements</b>	<b>xxi</b>
13.1	Speed and Latency Requirements . . . . .	xxi
13.2	Safety-Critical Requirements . . . . .	xxi
13.3	Precision or Accuracy Requirements . . . . .	xxii
13.4	Robustness or Fault-Tolerance Requirements . . . . .	xxii
13.5	Capacity Requirements . . . . .	xxii
13.6	Scalability or Extensibility Requirements . . . . .	xxii
13.7	Longevity Requirements . . . . .	xxii

<b>14 Operational and Environmental Requirements</b>	<b>xxii</b>
14.1 Expected Physical Environment . . . . .	xxii
14.2 Wider Environment Requirements . . . . .	xxiii
14.3 Requirements for Interfacing with Adjacent Systems . . . . .	xxiii
14.4 Productization Requirements . . . . .	xxiii
14.5 Release Requirements . . . . .	xxiii
<b>15 Maintainability and Support Requirements</b>	<b>xxiii</b>
15.1 Maintenance Requirements . . . . .	xxiii
15.2 Supportability Requirements . . . . .	xxiv
15.3 Adaptability Requirements . . . . .	xxiv
<b>16 Security Requirements</b>	<b>xxiv</b>
16.1 Access Requirements . . . . .	xxiv
16.2 Integrity Requirements . . . . .	xxiv
16.3 Privacy Requirements . . . . .	xxiv
16.4 Audit Requirements . . . . .	xxv
16.5 Immunity Requirements . . . . .	xxv
<b>17 Cultural Requirements</b>	<b>xxv</b>
17.1 Cultural Requirements . . . . .	xxv
<b>18 Compliance Requirements</b>	<b>xxv</b>
18.1 Legal Requirements . . . . .	xxv
18.2 Standards Compliance Requirements . . . . .	xxvi
<b>19 Open Issues</b>	<b>xxvi</b>
<b>20 Off-the-Shelf Solutions</b>	<b>xxvii</b>
20.1 Ready-Made Products . . . . .	xxvii
20.2 Reusable Components . . . . .	xxvii
20.3 Products That Can Be Copied . . . . .	xxvii
<b>21 New Problems</b>	<b>xxvii</b>
21.1 Effects on the Current Environment . . . . .	xxvii
21.2 Effects on the Installed Systems . . . . .	xxvii
21.3 Potential User Problems . . . . .	xxviii
21.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	xxviii

21.5 Follow-Up Problems . . . . .	xxviii
<b>22 Tasks</b>	<b>xxviii</b>
22.1 Project Planning . . . . .	xxviii
22.2 Planning of the Development Phases . . . . .	xxviii
22.3 Requirement Priority . . . . .	xxix
<b>23 Migration to the New Product</b>	<b>xxx</b>
23.1 Requirements for Migration to the New Product . . . . .	xxx
23.2 Data That Has to be Modified or Translated for the New System	xxx
<b>24 Costs</b>	<b>xxx</b>
24.1 Labor Costs . . . . .	xxx
24.2 Monetary Costs . . . . .	xxx
24.3 Additional Considerations . . . . .	xxx
<b>25 Likely Changes</b>	<b>xxx</b>
<b>26 Ideas for Solution</b>	<b>xxx</b>

## 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 in managing and keeping track of reimbursement requests, many students must endure long wait times during the process. The implementation of our solution will allow the MES to provide users with adequate tracking of reimbursement requests, 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 long wait times and lost money. We aim to provide a solution that effectively manages and organizes the large volume of reimbursement requests sent. The effectiveness of our solution will be measured based on how well it reduces wait times and 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 will allow users to gain financial oversight, ultimately preventing them from going over budget. Custom budgets should be categorized correctly nine times out of ten. The system should also reduce reimbursement turnover time for users, improving the claim process. Finally, the solution should 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 with extra instructions where needed.

# **2 System Overview**

In this section, the system architecture and general flow of how users will interact with the system are detailed.

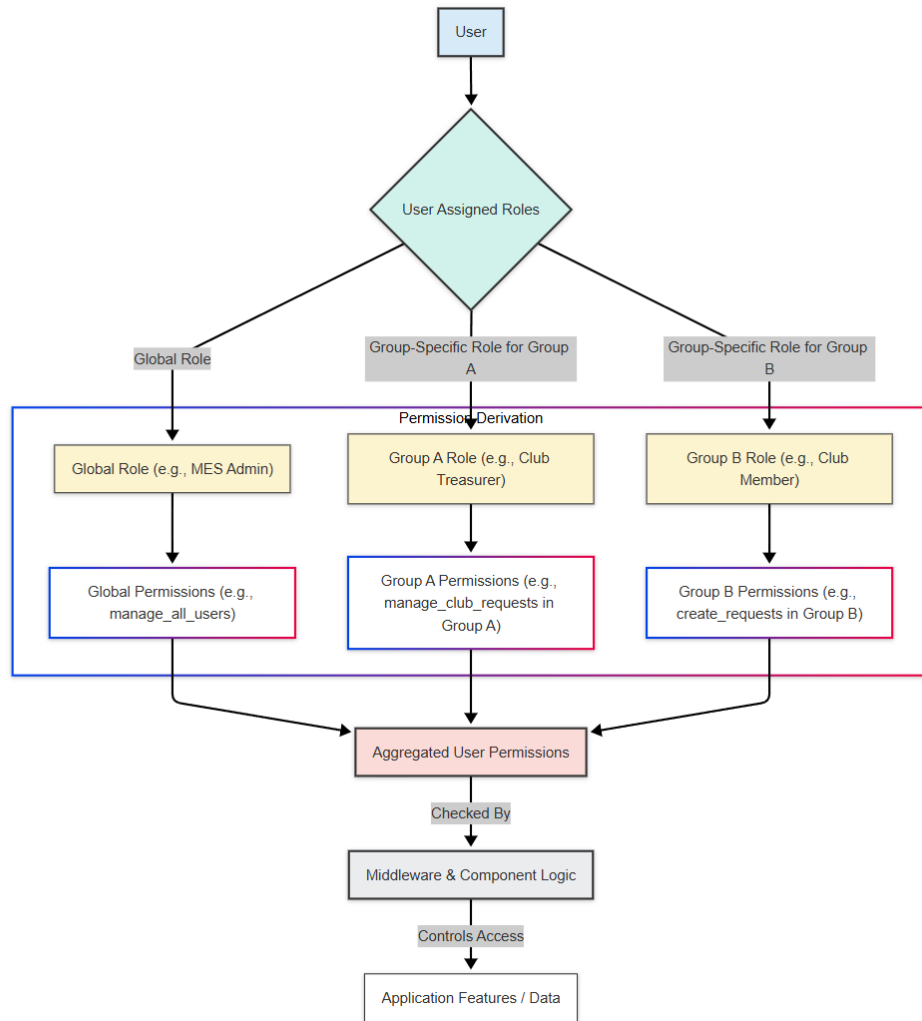


Figure 1: System process from different users



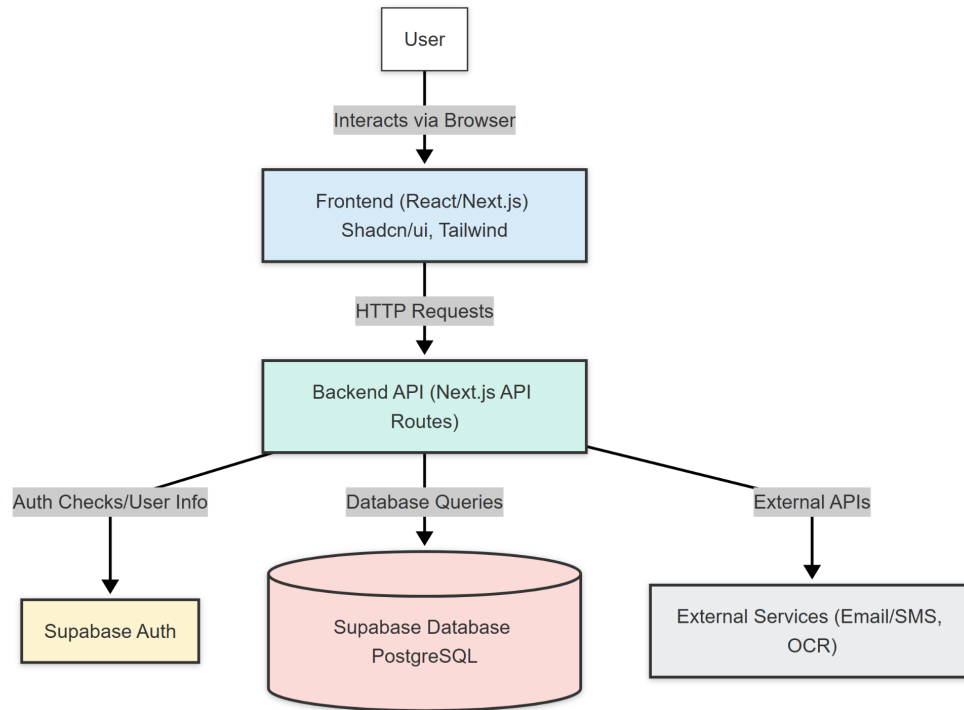


Figure 2: System Architecture

## 3 Stakeholders

### 3.1 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

## 3.2 Customer

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

- **Expense Submission:** Student leaders and members submit expense reports via a user-friendly interface.
- **Feedback Providers:** Provide feedback on system usability and required features.
- **User Adoption:** Their willingness to adopt the system affects overall success.

## 3.3 Other Stakeholders

Other stakeholders include individuals and groups that may influence or be impacted by the product:

- **Student Group Leaders:** Provide insights into the specific needs and challenges faced in the current reimbursement process.
- **MES Administrative Staff:** Assist in processing reimbursements and will need training on the new system to ensure smooth operations.
- **McMaster Administration:** May want to ensure that the tool follows relevant standards and policies.
- **Club Members:** Occasionally submit expense receipts and will want an easy experience.
- **MES IT:** Involved in integration with existing systems and maintenance.

## 3.4 Hands-On Users of the Project

- **Student Leaders:** Typically aged 18-25, manage club finances, need an intuitive interface.
- **Administrators:** MES staff aged 18-25, require tools for efficient review/approval workflows.

- **Financial Administrators:** Experienced professionals (30+), need advanced financial compliance and reporting features.

### 3.5 Personas

- **Alex, the Student Leader (20):** Seeks time-saving features for reimbursements.
- **Jamie, the MES Administrator (21):** Values accuracy and clear approval workflows.
- **James, the VP Finance (22):** Prefers tools providing high-level summaries and robust reports.
- **Jordan, the Club Member (19):** Submits occasional claims, wants straightforward UI.

### 3.6 Priorities Assigned to Users

- **Key Users:** Student leaders; they must easily submit expenses.
- **Secondary Users:** MES administrators; essential for approvals but secondary to consistent submissions.
- **Unimportant Users:** Casual users; their requirements are considered last.

### 3.7 User Participation

- **Requirements Gathering:** Interviews and surveys with key users to document the reimbursement process.
- **Usability Testing:** Hands-on sessions to validate design choices.
- **Regular Check-Ins:** Meetings to ensure alignment with user expectations.

### 3.8 Maintenance Users and Service Technicians

- **MES IT Staff:** Responsible for system updates, troubleshooting, and maintenance documentation.

## **4 Mandated Constraints**

### **4.1 Solution Constraints**

1. The system must manage reimbursement requests for 60 student groups efficiently.
2. The system must track and display the status of each request to the submitting group.
3. The system must allow uploading and storage of receipt images or files.
4. The system must be online continuously or display a clear message when offline.

### **4.2 Implementation Environment of the Current System**

1. There is no existing system in place.
2. The new system will be hosted on an MES-provided laptop or server running Windows 10 or higher, with at least 4 GB RAM and a modern GPU (e.g., RTX 2080).

### **4.3 Partner or Collaborative Applications**

1. N/A (No partner or collaborative application exists for this project).

### **4.4 Off-the-Shelf Software**

1. An external API may be used for scanning receipts; the final choice is pending.

### **4.5 Anticipated Workplace Environment**

1. Development will occur on local machines with Git for version control and collaboration.

## 4.6 Schedule Constraints

1. Final demonstration is set for April 2nd, serving as the primary deadline for core features.

## 4.7 Budget Constraints

1. The budget is capped at \$750 for the entire project.

## 4.8 Enterprise Constraints

1. All developed code and documentation will be owned by the MES.

# 5 Naming Conventions and Terminology

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

- **MES:** McMaster Engineering Society.
- **Audit Log:** A record of actions performed within the system, noting the user, action, and time.
- **Club Financial Manager:** A designated club member overseeing financial records and reimbursement submissions.
- **Budget Allocation:** The amount of funding designated for specific clubs or activities, used to track and control spending.
- **Vendor Contracts:** Agreements with external suppliers for goods or services.
- **Approval Workflow:** The process a reimbursement request undergoes from submission to final approval.
- **Financial Dashboard:** A system feature providing an overview of budgets, reimbursements, and pending actions.
- **CI/CD:** Continuous Integration/Continuous Deployment for automated testing and code deployment.

## **6 Relevant Facts And Assumptions**

### **6.1 Relevant Facts**

- No existing application is currently handling these reimbursements.
- The project targets 60 student groups at McMaster University.
- The system design must allow for future growth and additional features.

### **6.2 Business Rules**

- Users receive notifications when their submitted receipts are reimbursed.
- Users are notified in advance of any scheduled maintenance windows.
- All financial records are stored on the backend for reference and auditing.

### **6.3 Assumptions**

- The MES provides the server hardware for hosting the system.
- The MES supplies any data required for testing and validation.
- Reliable internet connectivity is assumed for normal usage and notifications.

## **7 The Scope of the Work**

### **7.1 The Current Situation**

Currently, the MES manages reimbursements through manual methods such as Google Forms, PDFs, and spreadsheets, causing delays, lost requests, and difficulty tracking finances.

## 7.2 The Context of the Work

The MES-ERP project aims to streamline these financial processes by enabling a single platform for reimbursement requests, budget oversight, and comprehensive reporting. This includes an audit trail for transaction histories.

## 7.3 Work Partitioning

- **Reimbursement Submission Module:** Allows users to upload receipts and request reimbursements.
- **Reimbursement Review and Approval:** Administrators approve or reject pending requests.
- **Audit and Compliance Module:** Maintains an audit log for all actions and financial records.
- **Budget/User Dashboards:** Provides real-time tracking of budgets, spending, and request statuses.

## 7.4 Specifying a Business Use Case (BUC)

### BUC 1: Submitting a Reimbursement Request

- **Precondition:** User has valid credentials and receipts.
- **Trigger:** A club needs reimbursement for an authorized expense.
- **Steps:**
  1. User logs into MES-ERP.
  2. User navigates to ‘Submit Reimbursement.’
  3. User completes required fields and uploads receipts.
  4. User submits the request.

- **Postcondition:** Request is placed in the approval queue.

### BUC 2: Reviewing a Reimbursement Request

- **Precondition:** A request has been submitted.

- **Trigger:** An MES administrator checks the pending requests or receives a notification.
- **Steps:**
  1. Administrator logs in.
  2. Administrator navigates to pending requests.
  3. Administrator reviews details and receipts.
  4. Administrator approves or rejects the request with optional feedback.
- **Postcondition:** Request status is updated for the submitter to view.

## 8 Business Data Model and Data Dictionary

### 8.1 Business Data Model

The MES-ERP system manages financial and user-related data across several entities:

- **User Entity:**
  - **Attributes:** UserID, Name, Email, Group, Role, Phone Number
  - **Relationships:** A user may submit multiple reimbursement requests; roles determine permissions.
- **Reimbursement Request Entity:**
  - **Attributes:** RequestID, Timestamp, SubmitterID, Amount, Status, Attached Receipts
  - **Relationships:** Each request is linked to a single user; administrators can review or modify the status.
- **Budget Entity:**
  - **Attributes:** BudgetID, GroupID, TotalAmountAllocated, AmountSpent, Category
  - **Relationships:** A group manages one budget, with possible multiple spending entries or reimbursement references.



- **Group Entity:**
  - **Attributes:** GroupID, GroupName, CreatedAt
  - **Relationships:** A group can have multiple associated users and budgets.
- **Role Entity:**
  - **Attributes:** RoleID, RoleName
  - **Relationships:** Determines which system actions are allowed (e.g., submit, approve).

## 8.2 Data Dictionary

N/A (No additional data dictionary provided at this time).

# 9 The Scope of the Product

## 9.1 Product Boundary

- **Automated Functions:**
  - Reimbursement request submission
  - Notification of updates and statuses
  - Approval/rejection workflow
  - Basic budget tracking
- **Manual Functions:**
  - Legacy data entry for older records
  - User training and support
  - Handling of non-digital (physical) receipts

Table 1: Product Use Case Summary Table

PUC No	PUC Name	Actor/s	Input & Output
1	Submit Reimbursement Request	Student Leader	(in) Reimbursement details(out) Confirmation message
2	Review Reimbursement Requests	Administrator	(in) Request details (out) Approved/Rejected status
3	Track Payment Requests	Administrator	(in) Payment info (out) Payment status
4	Process Funding Applications	Student Leader	(in) Funding application (out) Approval status
5	Save Repetitive Information	Student Leader	(in) User data (out) Data saved confirmation
6	Send Notifications	System	(in) Event triggers (out) Emails/SMS notifications
7	Manage Audit Trails	Financial Auditor	(in) Search/Filter criteria (out) Audit trail report

## 9.2 Product Use Case Table

## 9.3 Individual Product Use Cases (PUCs)

**PUC 1: Submit Reimbursement Request Actors:** Student Leader  
**Scenario:** A student leader submits an expense reimbursement request with attached receipts.

**PUC 2: Review Reimbursement Requests Actors:** Administrator  
**Scenario:** An administrator reviews and decides on a submitted request.

**PUC 3: Track Payment Requests Actors:** Administrator **Scenario:** An administrator verifies the status of paid or pending reimbursements.

**PUC 4: Process Funding Applications Actors:** Student Leader  
**Scenario:** A student leader submits an application for intramural or special funding.

**PUC 5: Save Repetitive Information Actors:** Student Leader **Scenario:** A user stores frequently used data (address, contact info) to simplify future submissions.

**PUC 6: Send Notifications Actors:** System **Scenario:** Automated notifications when request statuses change or approvals occur.

**PUC 7: Manage Audit Trails Actors:** Financial Auditor **Scenario:** Auditor reviews logs to ensure compliance or investigate anomalies.

# 10 Functional Requirements

## 10.1 Functional Requirements

- **Requirement Number:** 001
- **Description:** The system must allow club financial managers (or authorized users) to submit reimbursement requests with attached re-

ceipts.

- **Rationale:** Without a submission feature, expenses cannot be recorded, delaying financial processes and risking lost receipts.
- **Fit Criterion:**
  - A user with the “Club Financial Manager” role can upload receipts and complete a reimbursement form.
  - The new request must appear in the system’s pending queue at least 95% of the time without errors.
- **Priority:** High
- **Originator:** Club Financial Manager
- **Requirement Number:** 002
- **Description:** The system must allow MES staff to approve or reject reimbursement requests, updating the status accordingly.
- **Rationale:** Administrators require an approval workflow to finalize reimbursements; otherwise, requests cannot be processed or escalated.
- **Fit Criterion:**
  - An administrator can change a request’s status to Approved or Rejected.
  - The change must reflect in the database and be visible to the submitter.
- **Priority:** High
- **Originator:** MES Finance Team
- **Requirement Number:** 003
- **Description:** The system must notify the request submitter whenever there is a change in the status of the reimbursement request.

- **Rationale:** Submitters must be informed of any updates to address additional requirements or confirm approvals.
- **Fit Criterion:**
  - A notification is sent (email or user-chosen method) within 30 seconds of status change.
  - 90% of notifications should reach users successfully under normal conditions.
- **Priority:** Medium
- **Originator:** MES Finance Team
- **Requirement Number:** 004
- **Description:** The system must allow administrators to access all financial records, audit logs, and user actions for oversight.
- **Rationale:** Full access is needed for auditing, compliance, and potential investigations.
- **Fit Criterion:**
  - An administrator can retrieve data on any submitted reimbursement or user action without permission errors.
  - The system must maintain at least one year's worth of logs for each financial record.
- **Priority:** High
- **Originator:** System Administrators

## 11 Look and Feel Requirements

### 11.1 Appearance Requirements

1. Animations must be minimal and only used to provide necessary feedback.  
**Rationale:** Minimizing animations reduces distractions and maintains clarity of purpose.

## 11.2 Style Requirements

1. The system must not include any imagery or language that is discriminatory, vulgar, or otherwise inappropriate.

**Rationale:** Ensures a professional and inclusive environment.

2. The platform should avoid high-contrast color combinations that may strain the eyes.

**Rationale:** Preserves readability and user comfort.

3. Overall tone shall be professional and formal.

**Rationale:** Reinforces confidence in a financial system handling sensitive data.

4. Form fields shall be clearly labeled, with errors highlighted in red.

**Rationale:** Clear labels and error indicators reduce user confusion.

5. Icons for navigation or key actions shall be intuitive and consistent.

**Rationale:** Simplifies learning curve and speeds up task completion.

## 12 Usability and Humanity Requirements

### 12.1 Ease of Use Requirements

1. The system's UI elements (buttons, forms, menus) shall maintain consistent styling and positioning to promote ease of learning.

**Rationale:** Consistency helps new users quickly understand navigation and reduces errors.

2. The main menu shall allow navigation to primary features in three clicks or fewer.

**Rationale:** Quick navigation improves efficiency and user satisfaction.

### 12.2 Personalization and Internationalization Requirements

1. N/A (English only; no additional personalization or localization required at this stage).

## 12.3 Learning Requirements

1. The system shall provide a brief on-screen tutorial or help option for new users on their first reimbursement submission.

**Rationale:** Reduces user confusion and the likelihood of errors during early usage.

## 12.4 Understandability and Politeness Requirements

1. All error messages shall include a description of the issue and a suggestion for fixing it.

**Rationale:** Clear guidance on errors saves user time and reduces frustration.

## 12.5 Accessibility Requirements

1. The product shall offer a high-contrast mode for visually impaired users.

**Rationale:** Ensures the platform is accessible to those with visual impairments or color blindness.

# 13 Performance Requirements

## 13.1 Speed and Latency Requirements

- The system shall respond to user actions (page loads, form submissions) within 2 seconds for 90% of requests under normal load.
- Repetitive information entry or retrieval shall not exceed 1 second in typical usage scenarios.

## 13.2 Safety-Critical Requirements

- N/A (No immediate life-critical functionality; focus is on data security and accuracy).

### **13.3 Precision or Accuracy Requirements**

- All monetary values must be stored and displayed with two decimal places.
- The system shall validate numerical fields to prevent data entry errors (e.g., non-numeric amounts).

### **13.4 Robustness or Fault-Tolerance Requirements**

- The system shall back up financial records daily to prevent loss from system failures or corruption.

### **13.5 Capacity Requirements**

- The system shall support at least 200 concurrent users during peak usage without a noticeable slowdown.
- The database must store up to five years of financial transactions without performance degradation.

### **13.6 Scalability or Extensibility Requirements**

- The system shall allow for adding new user roles or additional student groups without a major redesign of the database schema.

### **13.7 Longevity Requirements**

- The product shall be maintainable for at least five years under a defined maintenance budget.

## **14 Operational and Environmental Requirements**

### **14.1 Expected Physical Environment**

1. The system will be used primarily in indoor office settings, with stable power and networking.

## 14.2 Wider Environment Requirements

1. The system is intended to replace manual methods (Google Forms, spreadsheets) for MES reimbursements.

## 14.3 Requirements for Interfacing with Adjacent Systems

1. The system must be accessible from Windows, Mac, and Linux devices, supporting at least the last four versions of major browsers (Chrome, Firefox, Edge).

## 14.4 Productization Requirements

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

## 14.5 Release Requirements

1. The product shall be tested for regressions before each official release.
2. Critical bugs must be addressed and resolved before a new version is deployed to production.

# 15 Maintainability and Support Requirements

## 15.1 Maintenance Requirements

1. The system must allow for future reimbursement process changes or an increase in student groups without needing a complete redesign.

**Rationale:** Ensures the software remains adaptable as MES operations evolve.

2. The codebase and technical documentation must enable a new developer to address general maintenance issues within one week of onboarding.

**Rationale:** Proper documentation and clear structure shorten learning curves for support staff.



## 15.2 Supportability Requirements

1. The system must run on MES-authorized devices meeting a minimum of Windows 10, 4 GB RAM, and stable internet connectivity.

**Rationale:** Standard hardware and networking reduce support complexity.

## 15.3 Adaptability Requirements

1. The system must handle scaling from 60 clubs to at least 100 clubs if the MES expands.

**Rationale:** Accommodates future growth without major refactoring.

# 16 Security Requirements

## 16.1 Access Requirements

1. Administrators must have full privileges to view, modify, or delete all financial records, user roles, and audit logs.

**Rationale:** Necessary for oversight, auditing, and compliance checks.

2. Club financial managers shall only access or modify data relevant to their own club.

**Rationale:** Protects privacy and data integrity across different clubs.

## 16.2 Integrity Requirements

1. All modifications to financial records must be logged with user ID, timestamp, and action details.

**Rationale:** Ensures traceability of changes for audit and compliance.

## 16.3 Privacy Requirements

1. The system must adhere to PIPEDA, ensuring personal data is handled securely and only for authorized purposes.

**Rationale:** Protects user data and aligns with legal standards.

2. Users should be able to request corrections or removal of their personal data if allowed by MES policies.

**Rationale:** Respects user rights and privacy.

## 16.4 Audit Requirements

1. The system must generate exportable logs detailing each significant financial action, from request submission to final approval.

**Rationale:** Facilitates investigations and historical record-keeping for compliance.

## 16.5 Immunity Requirements

1. All third-party libraries must be reviewed monthly for updates or security patches.

**Rationale:** Minimizes exposure to known vulnerabilities.

# 17 Cultural Requirements

## 17.1 Cultural Requirements

1. The interface language shall be English, using British spelling if conflict arises.

**Rationale:** Aligns with McMaster University's regional norms.

2. Inclusive language shall be used to avoid cultural bias or assumptions about users.

**Rationale:** Reflects the diversity of MES's student population.

# 18 Compliance Requirements

## 18.1 Legal Requirements

- **Data Protection Compliance:** Must comply with PIPEDA for data collection, storage, and consent.
- **Accessibility Compliance:** Follow AODA guidelines for an inclusive user experience.

## 18.2 Standards Compliance Requirements

- **Information Security Standards:** Align with ISO/IEC 27001 best practices for managing sensitive information.
- **Software Development Standards:** Adhere to Agile principles for iterative development and stakeholder feedback.

## 19 Open Issues

- **Issue Number:** 001
- **Cross-reference:** FR-003 (Notifications)
- **Summary:** Determining whether to send notifications via email, SMS, or in-app alerts.
- **Stakeholders Involved:** Project Team, MES IT Team
- **Action:** Conduct a survey or meeting to finalize the preferred method.
- **Resolution:** Expected within one week.
  
- **Issue Number:** 002
- **Cross-reference:** FR-004 (Audit Logs)
- **Summary:** Retention period for audit logs is unclear.
- **Stakeholders Involved:** Project Team, MES IT Team
- **Action:** Decide data retention policy in alignment with MES financial guidelines.
- **Resolution:** Expected within two weeks.
  
- **Issue Number:** 003
- **Cross-reference:** FR-002 (Approval Workflow)

- **Summary:** Thresholds for multi-level approvals (e.g., high-value requests) are undefined.
- **Stakeholders Involved:** Project Team, MES Finance Team
- **Action:** Define rules and thresholds for multi-level approvals.
- **Resolution:** Expected within two weeks.

## 20 Off-the-Shelf Solutions

### 20.1 Ready-Made Products

1. N/A (No existing product meets MES-specific needs).

### 20.2 Reusable Components

1. Existing UI libraries and receipt parsing APIs can be integrated.

### 20.3 Products That Can Be Copied

1. Some budgeting app UIs can inspire design but will require custom adaptation to MES workflows.

## 21 New Problems

### 21.1 Effects on the Current Environment

- Training might be needed as users transition away from manual systems.
- Historical data remains in spreadsheets unless explicitly imported.

### 21.2 Effects on the Installed Systems

- Must ensure hardware/software compatibility with existing MES infrastructure.
- Temporary hybrid usage (old and new) may occur during transition.

### 21.3 Potential User Problems

- Resistance to change from familiar manual processes.
- Errors if new users find the interface confusing.

### 21.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

- Server capacity may become an issue at peak load, requiring upgrades.
- Network outages or limited bandwidth could hinder usage.

### 21.5 Follow-Up Problems

- Need for additional training if advanced features or expansions are introduced.
- Possible legal/policy changes affecting how finances are tracked.

## 22 Tasks

### 22.1 Project Planning

An Agile approach (e.g., Scrum) will be used to iteratively refine requirements and deliverables.

### 22.2 Planning of the Development Phases

Step	Description
1. Requirements Gathering	Ongoing collaboration with stakeholders to define user stories and acceptance criteria.
2. Sprint Planning	Select and prioritize tasks from the backlog for each 1–4 week sprint.

<b>3. Design and Architecture</b>	Establish initial data models and system design, refining as needed.
<b>4. Development</b>	Implement features, coordinating with testers for immediate feedback.
<b>5. Testing</b>	Conduct unit, integration, and user acceptance tests within each sprint.
<b>6. Review and Retrospective</b>	Evaluate completed work, gather stakeholder feedback, and identify improvements for the next sprint.
<b>7. Release Planning</b>	Decide on feature packages and when to deploy them to production.
<b>8. Continuous Feedback</b>	Stakeholder input guides future sprints, ensuring alignment with evolving needs.

## 22.3 Requirement Priority

The following outlines the priority levels for each category of requirements in this SRS, along with brief rationales. These priorities guide the order in which features should be addressed during development when resources (time, budget) are constrained.

- Functional Requirements (Highest Priority)**  
**Rationale:** Core functionalities like request submission, approval workflows, and admin oversight (FR-001, FR-002, FR-003, and FR-004) form the backbone of the system. Without these features, the MES cannot handle reimbursements effectively.
- Security and Privacy Requirements (High Priority)**  
**Rationale:** Financial data is sensitive. Ensuring controlled access, encryption, and adherence to legal standards (e.g., PIPEDA) is critical to user trust and regulatory compliance. Security failures can severely undermine the system’s viability.
- Compliance Requirements (High Priority)**  
**Rationale:** Meeting legal (PIPEDA, AODA) and industry standards (ISO/IEC 27001) is non-negotiable for a financial platform. Non-compliance can lead to legal repercussions and reputational damage.
- Performance Requirements (Medium–High Priority)**  
**Rationale:** The system must reliably manage peak loads and handle

concurrent users. While performance optimization can occur iteratively, unacceptable lag or downtime would undermine user confidence and adoption.

- **Usability and Accessibility Requirements (Medium Priority)**  
**Rationale:** A user-friendly interface and features like high-contrast mode encourage adoption and reduce errors. Though essential for overall success, these can be refined after the core reimbursement logic is in place.
- **Look and Feel Requirements (Medium Priority)**  
**Rationale:** Maintaining a clean, professional interface supports a positive user experience, especially important for financial transactions. However, basic functionality and security must still take precedence.
- **Maintainability and Support Requirements (Medium Priority)**  
**Rationale:** Ensuring the codebase is well-documented and easily extensible will save time and costs in the long run, especially if MES processes change or the platform expands to serve more groups. While it may not block initial launch, neglecting maintainability increases technical debt.
- **Operational and Environmental Requirements (Medium Priority)**  
**Rationale:** The system's environment (offices, MES-provided hardware) and adjacency to other processes (e.g., budget committees, cloud backups) must be stable to avoid downtime. This stability underpins successful day-to-day usage.
- **Cultural Requirements (Low–Medium Priority)**  
**Rationale:** Respectful, inclusive language and British spelling (as required by McMaster conventions) are important but do not hinder core financial workflows. These can be verified or adjusted without blocking core features.
- **Longevity Requirements (Medium Priority)**  
**Rationale:** Building the system to last at least five years is valuable for ROI and operational continuity; however, it can be balanced with immediate functional needs and iterative improvements.

## **23 Migration to the New Product**

### **23.1 Requirements for Migration to the New Product**

N/A — No official migration of legacy data is currently mandated. Historical files remain in their existing formats unless otherwise requested.

### **23.2 Data That Has to be Modified or Translated for the New System**

N/A — No translation of old data is required for the first release. Further data integration can be explored later.

## **24 Costs**

### **24.1 Labor Costs**

- **Development Hours:** Estimated 400–500 hours to implement key features (reimbursement flow, approvals, budgeting).
- **Testing Hours:** Estimated 50–70 hours for comprehensive testing (unit, integration, acceptance).

### **24.2 Monetary Costs**

- **Server Operating Costs:** \$10–\$20 monthly for electricity and basic upkeep of MES's local server.

### **24.3 Additional Considerations**

- **Maintenance Costs:** 5–10 hours monthly for bug fixes and feature updates.
- **Backup Budget:** 10–20 hours set aside for unforeseen issues or requirement changes.



## 25 Likely Changes

Potential future additions include:

1. A mobile app for instant expense submission.
2. Machine learning-based receipt categorization to minimize manual data entry.
3. Customizable dashboards for different user roles.
4. Multi-language support to serve a broader community.
5. Historical data import from prior systems.

## 26 Ideas for Solution

A web-based finance and accounting solution using JavaScript, NextJS, React, and Git is proposed to handle the MES's reimbursement challenges:

- Streamlines submission with digital receipts and real-time status tracking.
- Offers automated notifications for both submitters and administrators.
- Includes budget oversight and multiple approval levels to prevent lost requests.

TypeScript will serve as the main language. NextJS handles front-end and back-end, with React providing a responsive, interactive UI. Git supports version control for collaborative development.

## References

Hunter, C. (n.d.). *Software Requirements Specification (SRS)* [PDF].

GitHub. [https:](https://github.com/cer-hunter/OAR-CAS741/blob/main/docs/SRS/SRS.pdf)

[//github.com/cer-hunter/OAR-CAS741/blob/main/docs/SRS/SRS.pdf](https://github.com/cer-hunter/OAR-CAS741/blob/main/docs/SRS/SRS.pdf)

McMaster Engineering Society. (n.d.). *Home*. <https://macengsociety.ca/>

## 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?**

As a team, we must acquire technical and domain-specific knowledge, as well as relevant soft skills. Key areas include:

- **Domain-Specific Knowledge:** Understanding the MES's financial workflows, budgeting cycles, and typical expense categories.
- **Frontend Development:** Building intuitive and responsive user interfaces with React.
- **Backend and Database Management:** Managing data securely and in real time, possibly using NextJS and Supabase.
- **Project Management:** Using Agile processes and Git for efficient collaboration and consistent code quality.

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? Which will the team pursue, and why?**

- **Domain-Specific Knowledge**
  - Interview MES finance staff and student leaders to understand current workflows.
  - Examine existing reimbursement documents and spreadsheets.
  - *Chosen Approach:* Direct interviews to obtain real-world context and pain points.
- **Frontend Development**
  - Complete online tutorials or short courses in React.
  - Build a small prototype and gather feedback from typical users.

- *Chosen Approach:* Prototype-based learning to identify usability issues early.

- **Backend and Database Management**

- Study official Supabase and PostgreSQL documentation.
- Implement a proof-of-concept backend to test data flow and security.
- *Chosen Approach:* Hands-on experimentation to confirm performance and reliability before scaling.

- **Project Management**

- Use GitHub Projects, Kanban boards, or Scrum boards for sprint planning.
- Conduct frequent reviews and retrospectives with stakeholders.
- *Chosen Approach:* An Agile framework enabling iterative feedback, ensuring the final product meets evolving needs.