Software Requirements Specification for Software Engineering: MES Finance Tracking Platform

Team #5, Money Making Mauraders
Zhenia Sigayev
Justin Ho
Thomas Wang
Michael Shi
Johnny Qu

October 10, 2025

Contents

iii

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Purpose of the Project - Thomas

1.1 User Business - Thomas

The McMaster Engineering Society (MES) seeks a more effective and efficient way to manage and track finances of its over 60 student clubs. Due to the current difficuties of the manual proces to manage and track reimbursements, the MES needs to streamline the process so reimbursements can occur regularly throughout the year and have them processed quicker to reduce processing times for students. The implementation of our solution will allow student clubs to submit reimbursement requests digitally and in an easier manner, allow MES reviewers to approve or reject requests efficiently, and provide transparency to all parties involved in the reimbursement process.

1.2 Goals of the Project - Thomas

The primary goals of the MES Finance Tracking Platform are to:

- Provide a user-friendly interface for club members to submit expense reimbursement requests.

- Implement a robust review and approval workflow for MES staff to manage reimbursement requests efficiently.

- Ensure transparency and traceability of all financial transactions related to club reimbursements.

- Maintain data security and privacy in compliance with McMaster University policies.

- Integrate seamlessly with existing MES systems and infrastructure.

2 System Overview

2.1 Architecture Diagram
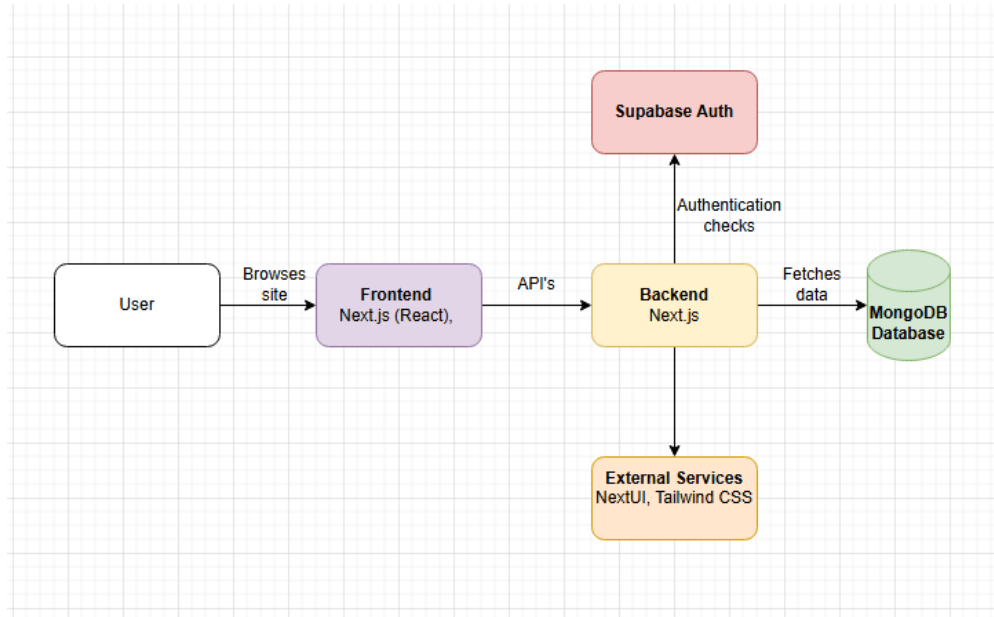
2.2 Technology Stack Overview

The system is a web-based application built with modern JavaScript frameworks and libraries, designed to support the needs of the MacEng Society website. The application provides user-facing web pages and administrative features for content management. It is structured as a client-server application where the frontend runs in a web browser and the backend interacts with a database.

2.3 Frontend (Client-Side)

- Developed using **React/Next.js**.

- Runs in modern web browsers such as Google Chrome, Microsoft Edge, Firefox, and Safari.

- Provides the user interface for interacting with content such as council information, events, and bookings.

- Fetches and displays data from the backend via RESTful or GraphQL APIs.

2.4   Backend (Server-Side)

- Executes on the **Node.js** runtime environment.

- Manages application logic, routing, and communication with the database.

- Uses **MongoDB Community Server** as the primary data storage system.

- Exposes APIs consumed by the frontend for dynamic content delivery.

2.5   Database

- Implemented using **MongoDB Community Server**.

- Stores member information, content, and other structured data required for the application.

2.6   Development Environment

- Source code is managed using **Git** and hosted in a repository such as GitHub.

- Developers use **Visual Studio Code (VSCode)** with mandatory extensions:

  - ESLint
  - Prettier

- Project dependencies are managed with **pnpm** or **yarn**.

- **Husky** and linting hooks enforce branch naming conventions, commit message standards, and code consistency.

2.7   External Dependencies

- **GitHub Actions** (via `.github/workflows`) for CI/CD and automated testing.

- **NextUI** and **TailwindCSS** for UI components and styling.

- Additional **Node.js** ecosystem libraries and packages installed via pnpm or yarn.

3   Stakeholders - Thomas

3.1   Direct Stakeholders

3.1.1   Club Leadership

- **Description**: Members of the executive team of clubs that fall under the MES.

- **Use Cases:**

  - Review and submit expense reimbursement requests on behalf of their club.
  - Track the status of submitted requests.
  - Review past reimbursements and budget accordingly.

3.1.2 Club General Members

- **Description**: General members of clubs that fall under the MES.

- **Use Cases:**

  - Submit expense reimbursement requests for expenses incurred on behalf of their club, pending review of executives.

3.1.3 McMaster University Finance

- **Description**: University staff responsible for auditing and overseeing the financial activities of student organizations.

- **Use Cases:**

  - Issue reimbursements to club members based on approved requests.
  - Ensure compliance with university financial policies and regulations.

3.2 Indirect Stakeholders

3.2.1 MES

- **Description**: Owns the platform and is responsible for the longterm sustainability, governance, and decisions regarding the platform.

- **Use Cases:**

  - Benefit from well-managed clubs and events funded through the MES.
  - Access information about club activities and financial transparency.

3.2.2 MES Student Developers

- **Description**: Students hired by the MES to maintain and improve the MES platform, including the finance tracking system.

- **Use Cases:**

  - Maintain and update the finance tracking platform as needed.
  - Ensure the platform remains secure, reliable, and user-friendly.

3.3   Personas - Thomas

- Wayne, Club President (Age 21): Looking for time-saving way to submit reimbursements

- Alice, Club Member (Age 20): Wants to easily submit expenses for club activities

- Frank, Club VP Finance (Age 21): Needs transparency into club spending and budgets

- Bob, MES Reviewer (Age 25): Needs efficient way to review and approve requests

- Eve, MES Admin (Age 30): Wants to ensure system is secure and compliant

- Charlie, MES Developer (Age 22): Needs maintainable and well-documented codebase

3.4   Priorities Assigned to Stakeholders - Thomas

- Key Users: Club Leadership; would like a easy way to submit and track reimbursements

- Secondary Users: MES Administrators; would like to ensure a secure and compliant way to manage and deliver reimbursements

- Tertiary Users: Casual Users; requirements are low priority

3.5   Stakeholder Participation - Thomas

- Gathering Requirements: Conduct interviews and surveys with club leaders and MES staff to understand key needs and an understanding of the current process and the desired changes.

- Usability Testing: Tangible session to validate and observe the usability of the platform with real users.

- Feedback and Iteration: Regularly gather feedback from stakeholders during development to ensure the platform meets their needs.

- Training and Support: Provide training sessions and support materials to help stakeholders effectively use the platform.

- Ongoing Engagement: Maintain open communication channels for continuous feedback and improvement after deployment.

4 Mandated Constraints

4.1 Solution Constraints

1. The system must be capable of managing expense reimbursement requests for up to 61 clubs or 7000 students.

2. The system must track and display the real-time status of each reimbursement request (e.g., submitted, under review, approved, rejected, reimbursed) to ensure transparency for submitters, reviewers, and administrators.

3. The system must allow users to attach digital receipts in PDF or image formats to their reimbursement requests.

4. The system must comply with data privacy regulations, ensuring that all personal and financial information is securely stored and transmitted.

5. The system must be able to import existing financial data from the current MES finance tracker to ensure continuity and data integrity.

6. The system must maintain consistent in accordance with the overall MES platform availability to support uninterrupted access.

4.2 Current MES Finance Tracker

- **MC X:**

  - **description:** The current MES finance tracker is not scalable and lacks proper traceability features.
  - **rationale:** The new system must be designed to integrate with existing MES infrastructure and address current limitations.

4.3 Cost Constraints

- **MC X:**

  - **description:** The amount spent by a team should not exceed $500.
  - **rationale:** The project must meet capstone budget constraints and be economically feasible for MES to support and maintain.

4.4 Off-the-Shelf Software

- **MC X:**

  - **description:** There are other financial tracking options (e.g., Quickbooks), but they may not match MES look-and-feel or budget.

- ○ **rationale:** Ensures the chosen solution integrates seamlessly with MES UX and licensing constraints.

4.5 Schedule Constraints

- **MC X:**

  - ○ **description:** The project must be completed by March 2026.
  - ○ **rationale:** Allows deployment one month before term end to avoid an end-of-term submission surge.

4.6 Workflow Constraints

- **MC X:**

  - ○ **description:** Capstone group will collaborate with MES developers using staging branches and incremental merges.
  - ○ **rationale:** Minimizes disruption and enables gradual adoption.

4.7 Enterprise Constraints

- **MC X:**

  - ○ **description:** Development must use technologies with favourable licensing terms for non-profits.
  - ○ **rationale:** Avoids licensing costs or obligations that could hinder MES future plans.

5 Terminology, Acronyms, and Technologies

5.1 Terminology

- **Audit Log**: A chronological record of all changes made to the system, including who made the change and when it was made.

- **Budget Allocation**: The total amount of funds allocated to a club or team for a specific period, typically a fiscal year.

- **Expense Claim**: A request for reimbursement submitted by a club or team member for expenses incurred on behalf of the club or team.

- **Receipt**: A digital or physical document that serves as proof of purchase for an expense.

- **Reimbursement**: The process of repaying a club or team member for approved expenses.

- **Reviewer**: An MES staff member or administrator responsible for reviewing and approving or rejecting expense claims.

- **Submitter**: A club or team member who submits an expense claim for reimbursement.

- **MES Student Developer**: Students that work with the MES to update, refine, and update code bases (will not be actively making changes to this application once completed)

- **MES Administrator**: An MES staff member with elevated permissions to manage the finance tracking platform, including user roles and system settings. They are the only members apart of the MES staff team that will interact with the application.

5.2   Acronyms

- **MES:** McMaster Engineering Society.

  - **description:** The student-led organization representing engineering students at McMaster University.

- **SLA:** Service Level Agreement.

  - **description:** A formal agreement between a service provider and a client that outlines the expected level of service.

- **CI:** Content Integration.

  - **description:** Content integration refers to validating the correctness of the software changes, typically running unit or integration tests.

- **CD:** Content Deployment.

  - **description:** Content deployment refers to deploying those changes to your product / platform (e.g., a website) for users to interact with.

- **CTA:** Call-to-Action.

  - **description:** An element of a web-page (e.g., a button or link) that encourages users to take a specific action.

- **UX:** User Experience.

  - **description:** The overall experience a user has when interacting with a product, including usability and design.

- **UI:** User Interface.

  - **description:** The visual and interactive elements users interact with (buttons, menus, forms).

- **OCR:** Optical Character Recognition.

  - **description:** Technology that converts images of text into editable and searchable data.

- **WCAG:** Web Content Accessibility Guidelines.

  - **description:** Guidelines from W3C to make web content more accessible to people with disabilities.

5.3   Technologies

- **Front-end:** The client-side part of the application that users interact with directly, typically built using HTML, CSS, and JavaScript frameworks.

- **Back-end:** The server-side part of the application that handles business logic, database interactions, and authentication, often built using languages like Python, Ruby, or Node.js.

- **Full-Stack:** A combination of both front-end and back-end development, where a developer is proficient in working on all layers of the application.

- **Testing:** The process of evaluating the functionality and performance of the application to ensure it meets the specified requirements and is free of defects.

- **Database:** A structured collection of data that is stored and managed to facilitate efficient retrieval and manipulation.

- **Cloud Services:** Online platforms that provide computing resources, storage, and services over the internet, such as AWS, Azure, or Google Cloud.

- **Version Control:** A system that tracks changes to code and allows multiple developers to collaborate on a project, with Git being the most popular version control system.

6   Relevant Facts And Assumptions

6.1   Relevant Facts

There exist a set of relevant facts related to the project that are not considered hard constraints or goals. They have been collected below.

- The MES has existing information about clubs stored within the MES monorepo. This data exists in a MongoDB database instance in non-relational, document form.

  - Their current applications restrict access based on role according to the data stored in this database.

- Internal documentation is stored in a seperate repo @ https://github.com/McMaster-Engineering-Society/Documentation (Private Repository)

- The MES has existing development practices. For the purposes of this project, they are generally considered *suggestions*.

  - A Kanban board has been setup for MES stakeholders to track our progress.
  - Current testing standards consist of informally performing manual end-to-end tests. If proof is required, video captures of testing are standard.
  - Work will be based on a staging branch in the monorepo specifically for our project.
  - Pre-existing styled React front-end components exist, and we may use them.

6.2 Business rules

These rules are imposed on our group, without necessarily directly being imposed as requirements or constraints on the project. Restrictions on group member interaction with existing user data, as an example, falls under this category.

- Code and data outside of that which we directly write for the project should not be modified without permission from a MES representative. This includes the aforementioned pre-styled components, any other MES projects, as well as existing user data.

- To maintain user privacy, group members will not view production database data. An exception is only allowed in exceptional circumstances under the explicit permission of the project supervisor, who is associated with the MES.

6.3 Assumptions

- **Development Period:** The development period will be up to the completion of the capstone 4G06 course. Further development may be considered at the discretion of the project supervisor and pending further agreement with development group members.

- **Communication with MES:** It is assumed that the MES will remain open to communication for the purposes of the project during the development period.

- **Developmental Discretion:** Development choices, including changes to any plan documents, are completely up to the discretion of the development group.

7 The Scope of the Work

7.1 The Current Situation

The MES currently uses spreadsheets and manual processes to track club finances. This approach results in a large amount of work for MES financial staff due to lost receipts, duplicated claims, or other manual errors. A centralized financial tracking platform is needed to streamline and automate this process for the MES.

7.2 The Context of the Work

The MES Information Technology (IT) team currently has systems and data related to clubs associated with their organization. The project however, will not directly interface with most of the other systems. The work will be conducted within the MES's monorepo, on an individual staging branch, and production databases and code will be owned by the MES.

7.3 Work Partitioning

Work on the project is to be partitioned such that development can proceed in an efficient manner while maintaining traceability and accountability for project supervisors. As the project involves a client body (the MES), requirement gathering and communications are included as part of the work.

- **Initial Requirements Gathering and Communications**

  - **Collection of initial information:** Directly elicit initial requirements, situation context, and problem context from the client body. Details about desired project output, existing systems, as well as expectations in terms of client supervision/input during project work.

  - **Formal requirement and planning documentation:** Create various documents, including this one, detailing the formal requirements, situation information, and development plan for the project.

- **Proof of Concept (PoC) Development**

  - **Local Wireframe Full-stack App Development:** Create a local implementation including all major system components to be required for the project. Capability of developing the full system will be demonstrated.

  - **Receipt Scanning Proof of Concept:** Develop a robust system for extracting finacial data from receipts as a PoC for other physical mediums of financial data. Testing and related documentation will be required to prove system effectiveness.

  - **User Authentication Proof of Concept:** Develop a system to authenticate users and restrict access to system data depending on user role.

– **Integration with MES Monorepo:** Integration of minimum PoC with existing MES monorepo to prove ability to fulfill project under ownership of the MES.

– **PoC Demonstration:** Demonstrate PoC to client and supervisors. Collect feedback and iterate development plans for fulfillment of the rest of project goals and functional requirements.

- **Integration and Requirement Fulfillment**

  – **Full Development Plan:** Create plan and work delegation for fulfillment of rest of functional requirements and goals of the project.

  – **Fulfillment Development:** Fulfill requirements according to plan through software development and iteration, including maintaining iterative communication with client and supervisors.

In addition to the mentioned work, communication with the client and testing of the application will be maintained during the entire development process.

7.4   Specifying a Business Use Case (BUC) - Thomas

- BUC 1: Uploading Receipts

  – **Precondition:** User has proper club roles to grant permission for receipt uploads (any member of a club, general or executive member, can upload receipt)

  – **Trigger:** User has valid receipts that require reimbursement

  – **Steps:**

    1. User logs into MES Financial Tracker
    2. User navigates to "Submit Reimbursement" page
    3. User uploads receipt and clicks "Upload Receipt" button

  – **Postcondition:** Receipt is uploaded and stored in database

- BUC 2: Submitting Reimbursement Request

  – **Precondition:** User has proper club roles to grant permission for reimbursement requests (only executive members of a club are allowed to officially submit a created reimbursement request)

  – **Trigger:** User has recieved notification of receipt upload success and is authorizing expense claim submission

  – **Steps:**

    1. User logs into MES Financial Tracker
    2. User navigates to "Submit Reimbursement" page
    3. User fills out reimbursement form and uploads receipts

4. User approves of request and clicks "Submit Request" button

- **Postcondition:** Reimbursement request is submitted and stored in database, MES admin is notified of submission

- BUC 3: Reviewing Reimbursement Request

  - **Precondition:** User is an MES admin with proper roles to review reimbursement requests
  - **Trigger:** User recieves notification of new reimbursement request submission
  - **Steps:**
    1. MES admin logs into MES Financial Tracker
    2. MES admin navigates to "Review Reimbursements" page
    3. MES admin reviews reimbursement request details and uploaded receipts
    4. MES admin approves or rejects request and clicks "Submit Review" button
  - **Postcondition:** Reimbursement request status is updated in database, submitter is notified of review outcome

8 Business Data Model and Data Dictionary - Thomas

8.1 Business Data Model - Thomas

- MES Admin User Entity:

  - **Description:** An individual associated with McMaster that verifies, validates, and administers the reimbursements
  - **Attributes:** Name, Email, roleID, Phone Number
  - **Relationship:** MES Admin may approve, reject, request additional information, and administer reimbursements

- Club Executive + General User Entity:

  - **Description:** Members apart of a MES club who wishes to submit items for reimbursements
  - **Attributes:** Name, macID, Email, clubID, roleID, Phone Number
  - **Relationship:** A club member may submit multiple reimbursement requests. Specified club executive team role grants permission to finalize reimbursement request on behalf of the club

- Club Entity:

  - **Description:** Club Entity that consists of all club members/those who will be creating reimbursement requests

- **Attributes:** clubID, clubName
- **Relationship:** A club entity that submits request on behalf of the club. Members within club entity can be added or removed (only for people with club admin team role)

- Reimbursement Request Entity:

  - **Description:** Reimbursement request that has been approved by club leadership and submitted to MES admin team for review
  - **Attributes:** clubID, requestID, Timestamp, amountRequested, requesetStatus, attachments(recipets)
  - **Relationship:** A request is linked to a particular club, MES admin team can review the reimbursement and change its status between approved, rejected, pending

- Club Finance Entity:

  - **Description:** Data displayed for club's financial allocation from MES
  - **Attributes:** clubID, totalAmountAllocated, totalAmountSpent, pendingAmount
  - **Relationship:** A display of the total resources available for a club

- Role Entity:

  - **Description:** List of all roles that can be assigned to an individual
  - **Attributes:** roleID, roleName
  - **Relationship:** Every person at McMaster has an account, no default role assigned. Roles will be later assigned

8.2 Data Dictionary - Thomas

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

9 Functional Requirements

9.1 Functional Requirements [FRQ]

1. The system shall allow MES clubs and teams to submit expense claims.

2. The system shall provide MES reviewers with tools to efficiently review, approve, or reject the reimbursement requests.

3. The system shall track the status of each expense claim (e.g., submitted, under review, approved, rejected, reimbursed).

4. The system shall permanently store and retain digital receipt submissions.

5. The system shall maintain an audit trail that records who submitted, reviewed, approved, or denied each expense claim.

6. The system shall enable access to club expense submissions to the submitters, and other club members with a role greater or equal to that of the submitter, or MES administrators and approvers.

10   Non-Functional Requirements

The following 7 sections outline the non-functional requirements for the MES Finance Tracking Platform.

11   Look and Feel Requirements

11.1   Appearance Requirements [APP]

1. The interface should use the McMaster Engineering Society's branding colors and logo.

2. The design should maintain a clean, modern layout using NextUI and TailwindCSS for consistency.

3. The application should use responsive design principles to adapt to desktop, tablet, and mobile displays.

4. Buttons, icons, and navigation components should follow accessible contrast ratios (WCAG 2.1 AA).

11.2   Style Requirements [STY]

1. Typography should match MES branding guidelines, using sans-serif fonts for readability.

2. UI components must adhere to consistent padding, spacing, and rounded-corner styles.

3. Notification banners and status indicators (e.g., "Submitted," "Pending," "Approved") should use color cues for clarity.

4. The application should provide a visually appealing balance between white space and content density.

12 Usability and Humanity Requirements

12.1 Ease of Use and Learning Requirements [EUL]

1. A straightforward optional overview or tutorial should be provided to first-time users.

12.2 Personalization and Internationalization Requirements [PaI]

1. The platform will be in English only.

12.3 Understandability and Politeness Requirements [UaP]

1. The platform will not use offensive language, imagery, symbols, or media of any kind.

2. All errors and warnings will be communicated in a clear, concise, and polite manner.

12.4 Accessibility Requirements [ABL]

1. Front-end navigation and interactions must comply with WCAG standards to ensure accessibility for users with disabilities.

2. The platform will allow users to select between light and dark mode themes.

3. The front-end will use semantic tags (¡header¿, ¡main¿, ¡nav¿, ¡footer¿, ¡article¿, ¡button¿, etc.) for structure.

4. All images must have meaningful alternative text (alt).

5. All functionality must be navigable from a keyboard (i.e., tab to move through content).

13 Performance Requirements

13.1 Speed and Latency Requirements [SaL]

1. API response times must average below 1 second for standard database queries and data submissions.

2. Page load times should not exceed 3 seconds under normal operating conditions on modern browsers.

3. File uploads (e.g., receipts or invoices) must complete within 5 seconds for files under 10 MB.

13.2   Safety-Critical Requirements [SAF]

1. The system must ensure that no reimbursement or financial transaction is executed without MES reviewer approval.

2. All monetary data must be validated before submission to prevent corruption or duplication.

13.3   Precision or Accuracy Requirements [PRE]

1. Financial transactions must maintain accuracy up to two decimal places.

2. All calculations involving funding allocations or reimbursements must be precise and consistent with MES accounting standards.

13.4   Robustness or Fault-Tolerance Requirements [FLT]

1. The system should handle unexpected backend or network failures gracefully, displaying meaningful error messages to users.

2. Data must be saved incrementally to prevent loss during unexpected session timeouts or browser crashes.

3. In the event of a failed API request, the system should automatically retry up to three times before reporting failure.

13.5   Capacity Requirements [CAP]

1. The system must support at least 500 concurrent users without noticeable degradation in performance.

2. The database must handle storage for at least 10,000 user records and 100,000 expense claims.

13.6   Scalability or Extensibility Requirements [EXT]

1. The system must be designed to allow horizontal scaling (e.g., deploying additional backend instances during high usage).

2. New features such as additional modules (e.g., club budgeting or analytics dashboards) should be easily integrable without major architectural changes.

13.7 Longevity Requirements [LNG]

1. The system is expected to remain operational and maintainable for at least 5 years with regular updates.

2. Future technology stack upgrades (React, Node.js, MongoDB versions) should not require a full system rewrite.

14 Operational and Environmental Requirements

14.1 Expected Physical Environment [PHY]

1. The system will be accessed through desktop and laptop computers using modern browsers (Chrome, Edge, Firefox, Safari).

2. Users may also access the site via tablets or smartphones with stable internet connectivity.

14.2 Wider Environment Requirements [ENV]

1. The application will be hosted on cloud infrastructure (e.g., Vercel or AWS) with automatic scaling and uptime monitoring.

2. The system must operate effectively under typical university network conditions, including campus Wi-Fi and VPN connections.

14.3 Requirements for Interfacing with Adjacent Systems [INF]

1. The system may integrate with McMaster's Single Sign-On (SSO) or authentication system in the future.

2. Potential interfaces include email APIs (e.g., SendGrid) for notifications and payment APIs for reimbursement processing.

14.4 Productization Requirements [PRO]

1. The application should be packaged for deployment using Docker containers to simplify installation and environment configuration.

2. Versioning of releases must follow semantic versioning conventions (e.g., v1.0.0).

14.5   Release Requirements [REL]

1. New versions of the application must be deployed through GitHub Actions CI/CD pipelines.

2. Each release must undergo code review and automated testing before deployment to production.

15   Maintainability and Support Requirements

15.1   Maintenance Requirements [MNT]

1. The codebase must adhere to consistent formatting and linting rules enforced by ESLint and Prettier.

2. All major system dependencies must be updated at least once per academic year to maintain compatibility and security.

15.2   Supportability Requirements [SUP]

1. The system should include detailed developer documentation covering setup, deployment, and troubleshooting steps.

2. MES technical staff or designated maintainers should have access to logs and monitoring dashboards for issue diagnosis.

15.3   Adaptability Requirements [ADP]

1. The application should support easy modification to accommodate policy or procedural changes in MES financial processes.

2. The UI should allow for additional modules or new forms without major rework to existing code.

16   Security Requirements

16.1   Access Requirements [ACS]

1. Only authenticated users may access the system.

2. Role-based access control (RBAC) must be implemented to distinguish between club members, reviewers, and administrators.

16.2 Integrity Requirements [INT]

1. All form submissions must be validated both client-side and server-side to prevent tampering.

2. Database operations must be protected against injection and cross-site scripting (XSS) attacks.

16.3 Privacy Requirements [PRI]

1. Personal information such as names and emails must be stored securely in compliance with McMaster University privacy policies.

2. Sensitive financial data must be encrypted at rest and in transit using TLS 1.2 or higher.

16.4 Audit Requirements [AUD]

1. Every key action (claim submission, approval, rejection, fund disbursement) must be logged with timestamp, user ID, and action details.

2. Audit logs must be retained for a minimum of 3 years for accountability and financial review.

16.5 Immunity Requirements [IMM]

1. The system must be protected from common web vulnerabilities (e.g., SQL injection, CSRF, XSS).

2. Rate limiting must be enforced to prevent denial-of-service (DoS) attacks.

17 Cultural Requirements

17.1 Cultural Requirements [CUL]

1. The system must use inclusive and gender-neutral language throughout all interfaces.

2. The design should align with the professional and academic tone of McMaster University and the Engineering Society.

18   Compliance Requirements [CMP]

18.1   Legal and University Policy Alignment

1. All data handling in this system must follow McMaster University's Data and Information Classification Policy (https://secretariat.mcmaster.ca/app/uploads/Data-and-Information-Class pdf). All data stored, processed, or transmitted by the application must be classified according to that policy prior to collection.

2. Personal identifiers (names, emails, student IDs) and financial records must be treated at or above the confidentiality level required by the University (i.e., treated as *confidential* or *higher* per the policy) and handled accordingly.

3. A Data Protection Impact Assessment (DPIA) or equivalent privacy review must be completed before production deployment.

18.2   Storage, Encryption, and Infrastructure [SEI]

1. Sensitive and confidential data must be stored only on approved infrastructure that meets McMaster security controls (e.g., approved cloud providers or on-prem systems with equivalent protections).

2. All network traffic carrying user or financial data must use TLS 1.2+; sensitive fields (e.g., banking details, full student numbers) must be encrypted at rest with strong algorithms and key management.

3. Backups containing sensitive data must be encrypted and retained only per the retention policy.

18.3   Access Control, Audit, and Logging [AUL]

1. Enforce role-based access control (RBAC) with least privilege; administrative access must require multi-factor authentication where available.

2. Maintain tamper-resistant audit logs for all key actions (submission, approval, rejection, data exports). Audit logs must be protected, monitored, and retained in accordance with university and MES retention rules (minimum three years for finance/audit logs as stated elsewhere in this SRS).

18.4   Retention, Disposal, and Legal Holds [RDL]

1. Implement data retention and secure disposal processes that conform to McMaster retention schedules and MES auditing requirements. Support legal holds to prevent deletion when required.

18.5 Standards Compliance Requirements [STD]

1. The software must conform to standard web development best practices (HTML5, CSS3, ECMAScript 6+).

2. Accessibility and usability must meet WCAG 2.1 AA compliance.

3. Version control, branching, and commit message standards must comply with MES development guidelines.

19 Open Issues

- Clubs portal login and administration system is not yet defined, however, it it being developed by MES developers: https://github.com/McMaster-Engineering-Society/MES-Website-App-Router/pull/791.

- The existing data is stored in Google Sheets and thus not easy to iterate over. The data will need to be manually migrated to the new system in a manner which has not yet been discussed.

20 Off-the-Shelf Solutions

20.1 Ready-Made Products

Two previous groups attempted to create the MES Finance Tracker system, however, both of their attempts were never integrated into the MES' platform.

- MES-ERP: https://github.com/Housam2020/MES-ERP

- Finance-platform: https://github.com/adampodolak/finance-platform

20.2 Reusable Components

- Third-party OCR software specializing in receipt scanning can be integrated. For example, Amazon's Textract.

- There exist reusable React.js components for the front-end that are proided by the MES to conform to thier look and feel requirements.

20.3 Products That Can Be Copied

- QuickBooks: https://quickbooks.intuit.com/ca/

- Expensify: https://www.expensify.com/

- Zoho Expense: https://www.zoho.com/expense/

- Wave: https://www.waveapps.com/

21 Likely Changes

- The MES will adopt a new database solution (e.g., Postgres) to store audit log trails and other financial data.

- The MES may want to add additional features such as budgeting tools or analytics dashboards in the future.

- The authentication process might need to support more roles if university administrators want to be involved.

22 Unlikely Changes

- Integration with external payment gateways (e.g., Stripe) for direct reimbursements is unlikely in the near term due to complexity and security concerns. Additionally, the MES uses E-Transfer which is free.

- Integration with third parties such as Plaid for bank account linking is unlikely due to privacy and security concerns.

- The core technology stack (React, Node.js, MongoDB) is unlikely to change significantly give the MES platform is built with these technologies.

23 Ideas for Solution

- Implementing a role-aware finance portal on the existing MES Next.js stack with dedicated dashboards for club members, treasurers, and administrators using reusable components (NextUI, Tailwind) that satisfy accessibility requirements.

- Building the reimbursement workflow as Next.js API routes on Node.js with MongoDB collections for clubs, requests, budgets, and attachments to enforce audit trails and status tracking.

- Establishing a GitHub Actions pipeline for linting, testing with Jest/RTL, and deploying to MES staging/production branches to maintain development workflow commitments.

- Using cloud storage (e.g., AWS S3) for receipt uploads with secure access controls and integrating OCR services (e.g., Amazon Textract) for automated data extraction.

- Designing the database schema to support scalability for future features like budgeting tools or analytics dashboards.

- Ensuring all sensitive data is encrypted in transit (TLS) and at rest, complying with McMaster University privacy policies.

24 Requirements Traceability

	APP-1	APP-2	APP-3	APP-4	STY-1	STY-2	STY-3	STY-4	EUL-1	ABL-1
FRQ-1							X			
FRQ-2		X			X	X	X			
FRQ-3		X			X	X	X			
FRQ-4							X			
FRQ-5	X	X			X	X	X			
FRQ-6		X			X	X	X			

	ABL-2	ABL-3	ABL-4	ABL-5	SAF-1	SAF-2	PRE-1	PRE-2	FLT-1	FLT-2
FRQ-1							X	X		
FRQ-2					X	X				
FRQ-3					X	X	X			
FRQ-4			X			X				X
FRQ-5					X	X	X			
FRQ-6					X	X	X			

	FLT-3	CAP-1	CAP-2	EXT-1	EXT-2	LNG-1	LNG-2	PHY-1	PHY-2	ENV-1
FRQ-1	X		X							
FRQ-2	X		X				X			
FRQ-3	X	X	X				X			
FRQ-4	X		X							
FRQ-5	X		X				X			
FRQ-6	X		X				X			

	ENV-2	INF-1	INF-2	PRO-1	PRO-2	REL-1	REL-2	MNT-1	MNT-2	SUP-1
FRQ-1									X	
FRQ-2							X		X	
FRQ-3			X				X		X	
FRQ-4									X	
FRQ-5							X		X	
FRQ-6							X		X	

	SUP-2	ADP-1	ADP-2	ACS-1	ACS-2	INT-1	INT-2	PRI-1	PRI-2	AUD-1
FRQ-1						X		X		X
FRQ-2			X		X	X		X		X
FRQ-3			X		X	X		X		X
FRQ-4						X	X	X	X	X
FRQ-5	X		X		X	X		X		X
FRQ-6			X		X	X		X		X

	AUD-2	IMM-1	IMM-2	CUL-1	CUL-2	SEI-1	SEI-2	SEI-3	AUL-1	AUL-2
FRQ-1						X				X
FRQ-2	X					X			X	X
FRQ-3	X					X			X	X
FRQ-4						X	X	X		X
FRQ-5	X					X			X	X
FRQ-6	X					X			X	X

	RDL-1	STD-1	STD-2	STD-3
FRQ-1				
FRQ-2	X			X
FRQ-3	X			X
FRQ-4	X			
FRQ-5	X			X
FRQ-6	X			X

As discussed in **?**, financial tracking systems require careful requirements analysis.

Appendix — Reflection

24.1 What went well while writing this deliverable?

- Zhenia Sigayev

  - Splitting up the work was once again an easy process for this deliverable. I believe that everyone was aligned on the project after previous meetings with the MES developers and our own internal meetings.

  - Due to thorough prior meetings with the MES, the requirements were planned ahead of time and straightforward to write.

- Justin Ho

  - TODO.

- Thomas Wang

  - TODO.

- Michael Shi

  - TODO.

- Johnny Qu

  - TODO.

24.2 How many of your requirements were inspired by speaking to your clients or their proxies?

- Nearly all of our requirements arose from speaking with our largest client, the MES. Additionally, our group members have been part of clubs that are overseen by the MES, which made reimbursement requirements gathering much easier. For example, it is apparent from the MES and club side that an approval workflow is necessary for both parties. Furthermore, our discussions showed the importance of strong security requirements.

24.3 What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project?

- Zhenia Sigayev

  - Align on MES reimbursement rules and compliance so workflows, validations, and audits reflect stakeholder expectations.

- ○ Understand NextAuth SSO and granular RBAC that fits privacy mandates and the MES monorepo.
- ○ Evaluate OCR options like Textract or custom CV, and package them in resilient async flows so manual entry stays optional.

- Justin Ho
  - ○ TODO.

- Thomas Wang
  - ○ TODO.

- Michael Shi
  - ○ TODO.

- Johnny Qu
  - ○ TODO.

24.4 What parts of this deliverable has each team member contributed to?

- Zhenia Sigayev

  - ○ Created the inital skeleton of the SRS document by integrating SRS-Volere into the normal SRS document.
  - ○ The entire "Mandated Constraints" section which includes subsections: Solution Constraints, Current MES Finance Tracker, Cost Constraints, Off-the-Shelf Software, Schedule Constraints, Workflow Constraints, and Enterprise Constraints.
  - ○ All of "Terminology, Acronyms, and Technologies" section.
  - ○ The subsection "The Current Situation".
  - ○ The subsections Functional Requirements, Ease of Use and Learning Requirements, Learning Requirements, Understandability and Politeness Requirements, Accessibility Requirements.
  - ○ Completed sections Reusable Components, Off-the-Shelf Solutions, Products That Can Be Copied, Likely Changes, Unlikely Changes, Ideas for Solution.
  - ○ Part of "What went well while writing this deliverable?", "What went well while writing this deliverable?"
  - ○ Completed the subsection "How many of your requirements were inspired by speaking to your clients or their proxies?"
  - ○ Delegated various last-mile remaining sections to all team members.

- Justin Ho

  - TODO.

- Thomas Wang

  - TODO.

- Michael Shi

  - TODO.

- Johnny Qu

  - TODO.

Appendix — References

[1] "Data and Information Classification Policy," McMaster University Secretariat,
https://secretariat.mcmaster.ca/app/uploads/Data-and-Information-Classification-
Policy.pdf (accessed Oct. 10, 2025).