

Problem Statement and Goals

Software Engineering

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

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...

1 Problem Statement

The engineering department currently uses spreadsheets and manual processes to track club finances. This approach often results in errors from lost receipts, duplicated claims, and large amount of claims at the end of the year. A centralized financial tracking software system is needed to streamline and automate this process.

The software will take clear inputs such as budgets, expenses, transactions, receipts, and any other financial information. The outputs will include reports, summaries, and visual dashboards. This system will be kept simple so it is easy to use and maintain for future student groups.

Stakeholders include student group leadership teams who manage budget, administrative staff who monitor spending and issue reimbursements, and faculty advisors. The environment the software will run on is the department's existing digital infrastructure with the software being in the form of a web platform

The importance of the problem lies in improving financial tracking efficiency, reducing the reporting time and support better financial decision making for student activities. The goals of the project are measurable: minimize errors in

transaction logging and receipt submissions, encourage continuous submissions over the school year rather than a large quantity at the end of the year, and provide clear visualization that makes budgets and spending easy to interpret.

1.1 Problem

- MES reimbursement process is manual and slow to complete
- expense submissions can come from multiple places making them difficult to access
- Hard to keep track of payments and receipts

1.2 Inputs and Outputs

Inputs:

- **User Information**
 - Authentication credentials
 - User role (club member, treasurer, MES admin)
 - Club affiliation
- **Financial Data**
 - Reimbursement amount
 - Transaction date
 - Expense category/purpose
 - Payment method used
- **Documentation**
 - Digital receipts (PDF, images, emails)
 - Physical receipt scans/photos
 - Supporting documentation (event details, approval emails)
- **Budget Information**
 - Club budget allocations
 - Funding category limits

Outputs:

- **User Interface**
 - Personalized dashboards by role
 - Reimbursement request history

- Status tracking for pending requests
- **Reporting**
 - Financial summaries by club/category
 - Budget utilization metrics
 - Transaction logs for auditing
 - Exportable reports (CSV, PDF)
- **Notifications**
 - Status change alerts (submitted, approved, rejected)
 - Reminder notifications for pending actions
 - Budget threshold warnings
- **Administrative Tools**
 - Approval workflows
 - Batch processing capabilities
 - System configuration options

[Characterize the problem in terms of 'high level' inputs and outputs. Use abstraction so that you can avoid details. —SS]

1.3 Stakeholders

- Group leadership teams in charge of managing MES budget
- Administrative staff in charge of monitoring spending provide reimbursements
- Faculty advisors who determine what expenses are eligible for reimbursement

1.4 Environment

[Hardware and Software Environment —SS] Software and Technology Stack:

- **Languages:**
 - TypeScript (primary)
 - CSS
- **Runtime:**
 - Node.js
- **Database:**

- MongoDB Community Server
- **Core Frameworks:**
 - Next.js (React-based framework for server-side rendering, routing, static site generation)
 - React (UI library)
 - Tailwind CSS (utility-first CSS framework)
 - NextUI (UI component library)
- **Major Libraries:**
 - @mui/material (Material UI for React)
 - @tanstack/react-query (Data fetching, state management)
 - axios (HTTP client)
 - next-auth (authentication)
- **Other Configuration/Plugin Files:**
 - next-sitemap (sitemap generation)
 - pnpm/yarn (dependency management)

2 Goals

3 Stretch Goals

4 Extras

[For CAS 741: State whether the project is a research project. This designation, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

[For SE Capstone: List your extras. Potential extras include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. (The full list is on the course outline and in Lecture 02.) Normally the number of extras will be two. Approval of the extras will be part of the discussion with the instructor for approving the project. The extras, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?