

# Module Guide for MES-ERP

Team #26, Ethical Pals

Sufyan Motala

Rachid Khneisser

Housam Alamour

Omar Muhammad

Taaha Atif

January 17, 2025

# 1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

## 2 Reference Material

This section records information for easy reference.

### 2.1 Abbreviations and Acronyms

symbol	description
AC	Anticipated Change
DAG	Directed Acyclic Graph
M	Module
MG	Module Guide
OS	Operating System
R	Requirement
SC	Scientific Computing
SRS	Software Requirements Specification
MES-ERP	Explanation of program name
UC	Unlikely Change
<a href="#">[etc. —SS]</a>	<a href="#">[... —SS]</a>

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Reference Material</b>	<b>ii</b>
2.1	Abbreviations and Acronyms . . . . .	ii
<b>3</b>	<b>Introduction</b>	<b>1</b>
<b>4</b>	<b>Anticipated and Unlikely Changes</b>	<b>2</b>
4.1	Anticipated Changes . . . . .	2
4.2	Unlikely Changes . . . . .	2
<b>5</b>	<b>Module Hierarchy</b>	<b>2</b>
<b>6</b>	<b>Connection Between Requirements and Design</b>	<b>3</b>
6.1	Design Decisions for Each Requirement . . . . .	3
<b>7</b>	<b>Module Decomposition</b>	<b>5</b>
7.1	Module Levels . . . . .	5
7.2	Use Relations Between Modules . . . . .	6
7.3	Hardware Hiding Modules (M1) . . . . .	7
7.4	Behaviour-Hiding Module . . . . .	7
7.4.1	Input Format Module (M??) . . . . .	8
7.4.2	Etc. . . . .	8
7.5	Software Decision Module . . . . .	8
7.5.1	Etc. . . . .	8
<b>8</b>	<b>Traceability Matrix</b>	<b>8</b>
8.1	Anticipated Changes to Modules . . . . .	8
8.2	Impact of Anticipated Changes . . . . .	8
<b>9</b>	<b>Use Hierarchy Between Modules</b>	<b>10</b>
<b>10</b>	<b>User Interfaces</b>	<b>12</b>
10.1	Dashboard . . . . .	12
10.2	Form . . . . .	12
10.3	Login . . . . .	13
10.4	Receipt Input . . . . .	14
10.5	Website Tutorial . . . . .	14
<b>11</b>	<b>Design of Communication Protocols</b>	<b>15</b>
<b>12</b>	<b>Timeline</b>	<b>15</b>

## List of Tables

1	Module Hierarchy . . . . .	3
2	Connection Between Requirements and Modules . . . . .	4
3	Traceability Matrix: Anticipated Changes to Modules . . . . .	9

## List of Figures

1	Module Hierarchy . . . . .	11
2	Dashboard View . . . . .	12
3	Form View . . . . .	12
4	Login View . . . . .	13
5	Receipt Scanning View . . . . .	14
6	Tutorial Page . . . . .	14

### 3 Introduction

Decomposing a system into modules is a commonly accepted approach to developing software. A module is a work assignment for a programmer or programming team (Parnas et al., 1984). We advocate a decomposition based on the principle of information hiding (Parnas, 1972). This principle supports design for change, because the “secrets” that each module hides represent likely future changes. Design for change is valuable in SC, where modifications are frequent, especially during initial development as the solution space is explored.

Our design follows the principles laid out by Parnas et al. (1984), as follows:

- System details that are likely to change independently are the secrets of separate modules.
- Each data structure is implemented in only one module, ensuring clarity and responsibility for managing that structure.
- Any program requiring information from a module’s data structures must access it via well-defined interfaces.

The MES-ERP financial platform’s design focuses on:

- Modularity: Each functional requirement is encapsulated in a distinct module to isolate changes and enhance maintainability.
- Scalability: Modules are designed to accommodate future enhancements, such as support for new input formats or integration with third-party systems.
- Robustness: Validation and error-handling mechanisms ensure data integrity across all modules.

After completing the Software Requirements Specification (SRS), the Module Guide (MG) was developed (Parnas et al., 1984). The MG specifies the modular structure of the system and is intended to allow both designers and maintainers to easily identify and work on the system’s components.

The potential readers of this document include:

- New Project Members: Provides an overview of the system’s structure, allowing new members to quickly identify relevant modules.
- Maintainers: Ensures maintainers can understand and update the system while minimizing the impact on unrelated modules.
- Designers: Allows designers to verify consistency, feasibility, and flexibility of the system’s architecture.

## 4 Anticipated and Unlikely Changes

This section lists possible changes to the system, categorized as either anticipated or unlikely. These changes reflect both user-driven requirements and system-level constraints.

### 4.1 Anticipated Changes

Anticipated changes are those expected to occur due to user feedback, evolving requirements, or regular system updates. These changes are addressed by hiding information in the appropriate modules.

#### **AC1:** Input Data Format

- Impacted Modules: Input Format Module, Data Validation Module
- Examples: Adding support for new formats (e.g., images, PDFs) or OCR for scanned receipts.

#### **AC2:** Notification Methods

- Impacted Modules: Notification System
- Examples: Adding SMS or push notification support.

#### **AC3:** Reimbursement Workflow

- Impacted Modules: Reimbursement Submission, Approval Workflow
- Examples: Modifying approval rules or automating incomplete submission rejections.

### 4.2 Unlikely Changes

Unlikely changes are less probable due to their disruptive nature or misalignment with project goals.

#### **UC1:** Technology Stack

- Impacted Modules: All
- Examples: Switching from TypeScript/Next.js to another framework.

## 5 Module Hierarchy

Modules are organized to encapsulate secrets and ensure clear responsibilities. The hierarchy reflects relationships between modules and their dependencies, as shown in Table 1.

**M1: Hardware-Hiding Module:** Abstracts database and file system interactions.

**M2: Behaviour-Hiding Module:** Implements workflows for submissions, reviews, and dashboards.

**M3: Software Decision Module:** Encapsulates shared logic for validation, testing, and CI/CD.

Level 1	Level 2
Hardware-Hiding Module	Database Interaction Layer
	Reimbursement Submission
	Reimbursement Review and Approval
	Budget Dashboard
Behaviour-Hiding Module	Notification System
	User Management
	Financial Reporting
	GUI Module
Software Decision Module	CI/CD Integration
	Data Validation Module
	Test Automation Framework

Table 1: Module Hierarchy

## 6 Connection Between Requirements and Design

The design of the MES-ERP system has been structured to ensure that the requirements outlined in the Software Requirements Specification (SRS) are met comprehensively. Table 2 illustrates the connection between the requirements and the modules designed to fulfill them.

### 6.1 Design Decisions for Each Requirement

- **R1: Secure Authentication**

- **Module:** User Authentication & Profile Management Module
- **Design Decision:** Implements secure login with session timeouts, role-based access control, and account lockout mechanisms.



Table 2: Connection Between Requirements and Modules

Requirement (R)	Modules
R1: Secure Authentication	User Authentication & Profile Management Module
R2: Expense Submission & Tracking	Expense Submission & Tracking Module
R3: Budget Validation	Budget and Funding Management Module
R4: Approval Workflow	Approval Workflow and Review Module
R5: Notifications	Notifications & Communication Module
R6: Compliance	Policy & Compliance Management Module
R7: Reporting	Reporting and Analytics Module
R8: Administrative Tools	Administrator and Configuration Panel Module

- **R2: Expense Submission & Tracking**

- **Module:** Expense Submission & Tracking Module
- **Design Decision:** Provides forms for submission, receipt uploads, and tracking expense status.

- **R3: Budget Validation**

- **Module:** Budget and Funding Management Module
- **Design Decision:** Ensures expense requests do not exceed departmental budgets and integrates with the financial system.

- **R4: Approval Workflow**

- **Module:** Approval Workflow and Review Module
- **Design Decision:** Implements dynamic routing for approvals and notifications for pending actions.

- **R5: Notifications**

- **Module:** Notifications & Communication Module
- **Design Decision:** Sends alerts via email, SMS, and dashboard notifications for system events.

- **R6: Compliance**

- **Module:** Policy & Compliance Management Module
- **Design Decision:** Validates expense requests against predefined policies to ensure compliance with regulations.
- **R7: Reporting**
  - **Module:** Reporting and Analytics Module
  - **Design Decision:** Generates detailed reports in PDF/CSV format for expense tracking and system usage.
- **R8: Administrative Tools**
  - **Module:** Administrator and Configuration Panel Module
  - **Design Decision:** Enables administrators to manage user roles, configure approval workflows, and access system logs.

## 7 Module Decomposition

The MES-ERP system is designed following the principle of *information hiding*, which ensures that each module encapsulates decisions likely to change independently. The modules are organized into a hierarchy, with higher-level modules relying on lower-level ones for functionality. This hierarchy forms a directed acyclic graph (DAG). Figure ?? illustrates the module decomposition.

### 7.1 Module Levels

- **Hardware-Hiding Modules**
  - **Database Module:** Provides a foundational interface for storing and retrieving data. It serves as the core data storage system, used by nearly all other modules.
- **Behavior-Hiding Modules**
  - **User Authentication & Profile Management Module:** Manages secure login, user sessions, and profiles.
  - **Expense Submission & Tracking Module:** Handles expense submission, receipt uploads, and tracking.
  - **Policy & Compliance Management Module:** Ensures submitted expenses comply with organizational policies.
- **Software Decision Modules**
  - **Approval Workflow and Review Module:** Manages dynamic workflows for expense approvals.

- **Budget and Funding Management Module:** Validates budgets and tracks funding.
- **Reporting and Analytics Module:** Provides reporting tools and analytics for system data.
- **Presentation Layer Modules**
  - **Graphical User Interface (GUI) Module:** Interacts with users by displaying data and capturing inputs.
  - **Notifications & Communication Module:** Sends alerts and notifications to users via email, SMS, or the dashboard.

## 7.2 Use Relations Between Modules

- **Database Module**
  - **Used By:** User Authentication & Profile Management, Expense Submission & Tracking, Budget and Funding Management, Approval Workflow and Review, Notifications & Communication, Reporting and Analytics, Policy & Compliance Management.
  - **Purpose:** Acts as the central data storage and retrieval system.
- **User Authentication & Profile Management Module**
  - **Used By:** Approval Workflow and Review, Notifications & Communication, GUI Module.
  - **Purpose:** Provides secure access to the system and maintains user roles and profiles.
- **Expense Submission & Tracking Module**
  - **Used By:** Approval Workflow and Review, Budget and Funding Management, Notifications & Communication, Reporting and Analytics.
  - **Purpose:** Handles expense submissions, categorization, and tracking.
- **Budget and Funding Management Module**
  - **Used By:** Expense Submission & Tracking, Approval Workflow and Review, Notifications & Communication.
  - **Purpose:** Validates and updates budgets associated with submitted expenses.
- **Approval Workflow and Review Module**
  - **Used By:** Notifications & Communication, GUI Module.

- **Purpose:** Implements dynamic routing and rules for approving expenses.
- **Notifications & Communication Module**
  - **Used By:** GUI Module, all other modules requiring alerts or updates.
  - **Purpose:** Sends notifications and alerts via email, SMS, or dashboard.
- **Reporting and Analytics Module**
  - **Used By:** GUI Module, Administrator and Configuration Panel.
  - **Purpose:** Provides analytics and reporting tools for system data.
- **Graphical User Interface (GUI) Module**
  - **Used By:** End users interacting with the system.
  - **Relies On:** Backend modules to fetch and display data.
  - **Purpose:** Displays system data and handles user interactions.

### 7.3 Hardware Hiding Modules (M1)

**Secrets:** The data structure and algorithm used to implement the virtual hardware.

**Services:** Serves as a virtual hardware used by the rest of the system. This module provides the interface between the hardware and the software. So, the system can use it to display outputs or to accept inputs.

**Implemented By:** OS

### 7.4 Behaviour-Hiding Module

**Secrets:** The contents of the required behaviours.

**Services:** Includes programs that provide externally visible behaviour of the system as specified in the software requirements specification (SRS) documents. This module serves as a communication layer between the hardware-hiding module and the software decision module. The programs in this module will need to change if there are changes in the SRS.

**Implemented By:** –

#### 7.4.1 Input Format Module (M??)

**Secrets:** The format and structure of the input data.

**Services:** Converts the input data into the data structure used by the input parameters module.

**Implemented By:** [Your Program Name Here]

**Type of Module:** [Record, Library, Abstract Object, or Abstract Data Type] [Information to include for leaf modules in the decomposition by secrets tree.]

#### 7.4.2 Etc.

### 7.5 Software Decision Module

**Secrets:** The design decision based on mathematical theorems, physical facts, or programming considerations. The secrets of this module are *not* described in the SRS.

**Services:** Includes data structure and algorithms used in the system that do not provide direct interaction with the user.

**Implemented By:** –

#### 7.5.1 Etc.

## 8 Traceability Matrix

This section outlines the anticipated changes to the MES-ERP system and identifies the modules that would be impacted by each change. The traceability matrix maps these anticipated changes to the affected modules, ensuring alignment with the principle of *information hiding*.

### 8.1 Anticipated Changes to Modules

Table 3 provides a traceability matrix that lists the anticipated changes and their associated modules.

### 8.2 Impact of Anticipated Changes

- **AC1: Changes in Authentication Protocols**
  - **Affected Module:** User Authentication & Profile Management Module
  - **Impact:** Updates to the `authenticate()` function to support new protocols (e.g., multi-factor authentication) and modifications to constants like session timeouts.

Table 3: Traceability Matrix: Anticipated Changes to Modules

Anticipated Change (AC)	Affected Modules
AC1: Changes in Authentication Protocols	User Authentication & Profile Management Module
AC2: New Expense Categories	Expense Submission & Tracking Module
AC3: Budget Limits Adjustment	Budget and Funding Management Module
AC4: Workflow Modifications	Approval Workflow and Review Module
AC5: New Notification Methods	Notifications & Communication Module
AC6: Policy Updates	Policy & Compliance Management Module
AC7: Report Formats or Types	Reporting and Analytics Module
AC8: Additional Admin Features	Administrator and Configuration Panel Module

- **Mitigation:** Encapsulation of authentication logic minimizes impact on other modules.

- **AC2: New Expense Categories**

- **Affected Module:** Expense Submission & Tracking Module
- **Impact:** Updates to the `EXPENSE_CATEGORIES` constant and adjustments to the `categorizeExpense()` function.
- **Mitigation:** Modular handling of categories allows isolated changes without affecting other modules.

- **AC3: Budget Limits Adjustment**

- **Affected Module:** Budget and Funding Management Module
- **Impact:** Modifications to the `MAX_BUDGET` constant and related validation logic.
- **Mitigation:** Centralized budget handling simplifies updates.

- **AC4: Workflow Modifications**

- **Affected Module:** Approval Workflow and Review Module
- **Impact:** Updates to `routeRequest()` and `updateStatus()` functions for new workflow rules.
- **Mitigation:** Dynamic routing logic supports flexible workflow updates.

- **AC5: New Notification Methods**

- **Affected Module:** Notifications & Communication Module

- **Impact:** Addition of new constants (e.g., `NOTIFICATION_TYPES`) and updates to notification functions.
  - **Mitigation:** Modular notification handling allows easy integration of new methods.
- **AC6: Policy Updates**
    - **Affected Module:** Policy & Compliance Management Module
    - **Impact:** Updates to `validateRequest()` to handle new policy rules and thresholds.
    - **Mitigation:** Encapsulation of policy logic ensures minimal impact on unrelated modules.
  - **AC7: Report Formats or Types**
    - **Affected Module:** Reporting and Analytics Module
    - **Impact:** Modifications to `generateReport()` to support new formats or metrics.
    - **Mitigation:** Flexible report generation logic minimizes system-wide impact.
  - **AC8: Additional Admin Features**
    - **Affected Module:** Administrator and Configuration Panel Module
    - **Impact:** New access programs (e.g., `addFeature()`, `updateRolePermissions()`).
    - **Mitigation:** Well-defined admin tools enable straightforward updates.

## 9 Use Hierarchy Between Modules

In this section, the uses hierarchy between modules is provided. [Parnas \(1978\)](#) said of two programs A and B that A *uses* B if correct execution of B may be necessary for A to complete the task described in its specification. That is, A *uses* B if there exist situations in which the correct functioning of A depends upon the availability of a correct implementation of B. Figure ?? illustrates the use relation between the modules. It can be seen that the graph is a directed acyclic graph (DAG). Each level of the hierarchy offers a testable and usable subset of the system, and modules in the higher level of the hierarchy are essentially simpler because they use modules from the lower levels.

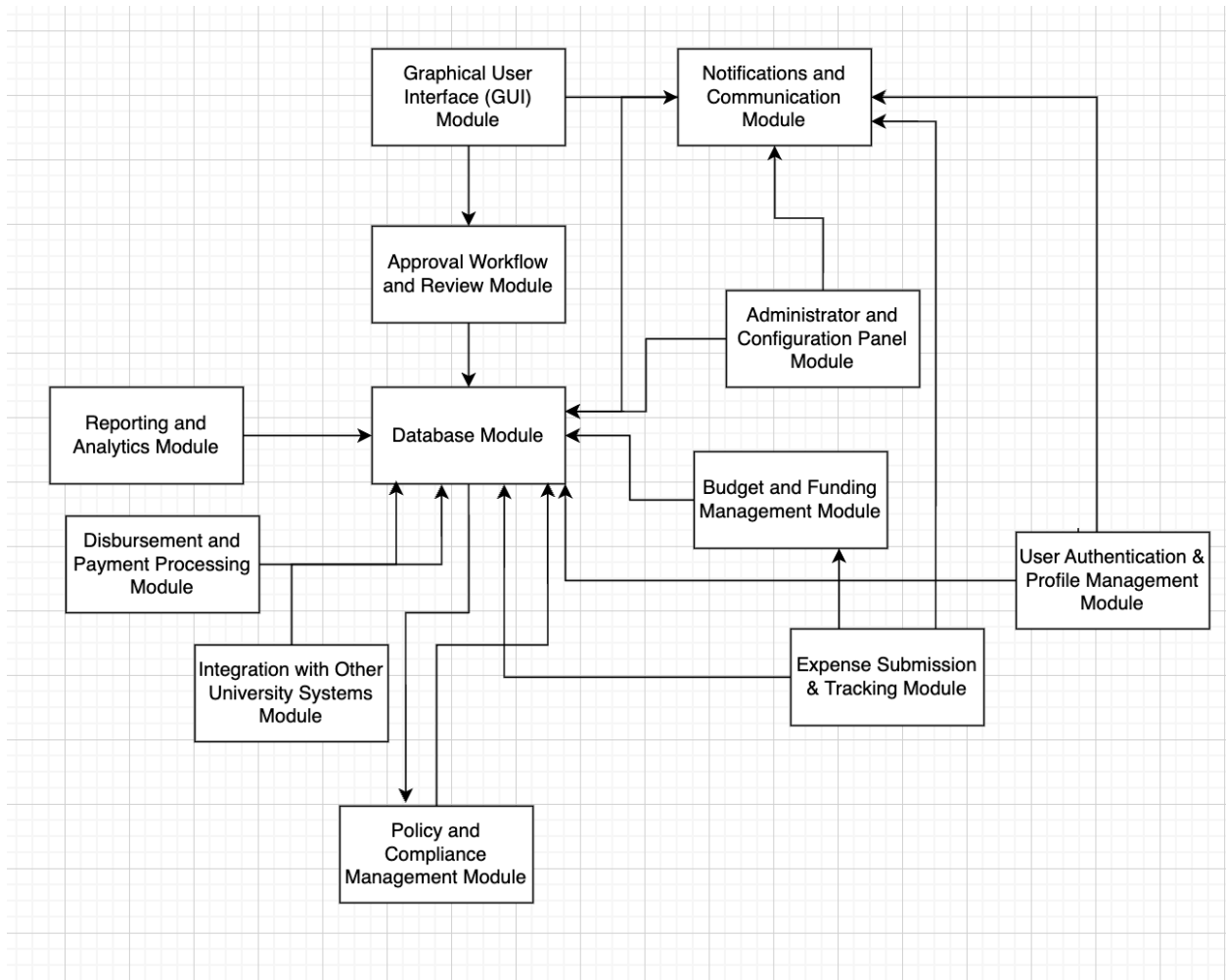
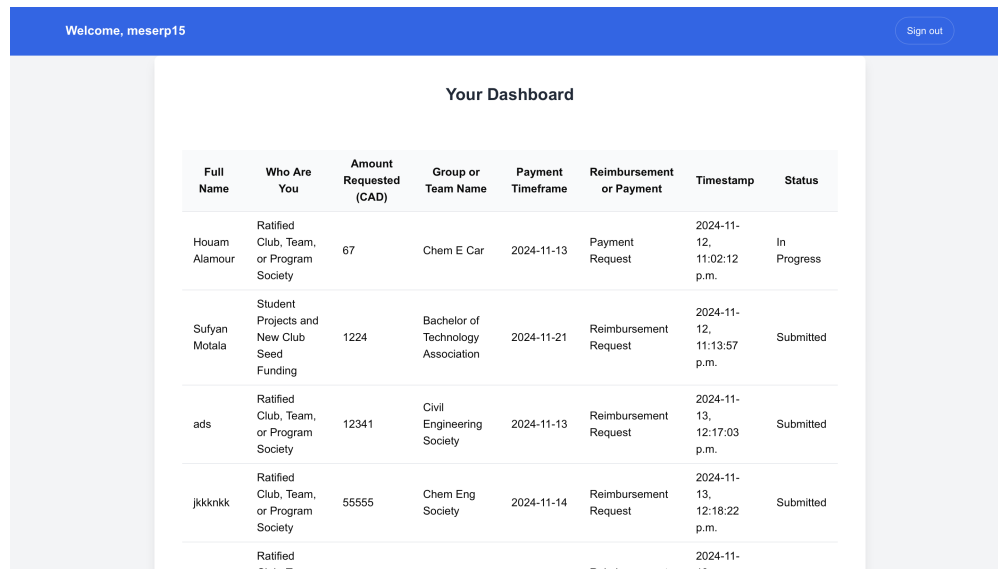


Figure 1: Module Hierarchy



## 10 User Interfaces

### 10.1 Dashboard

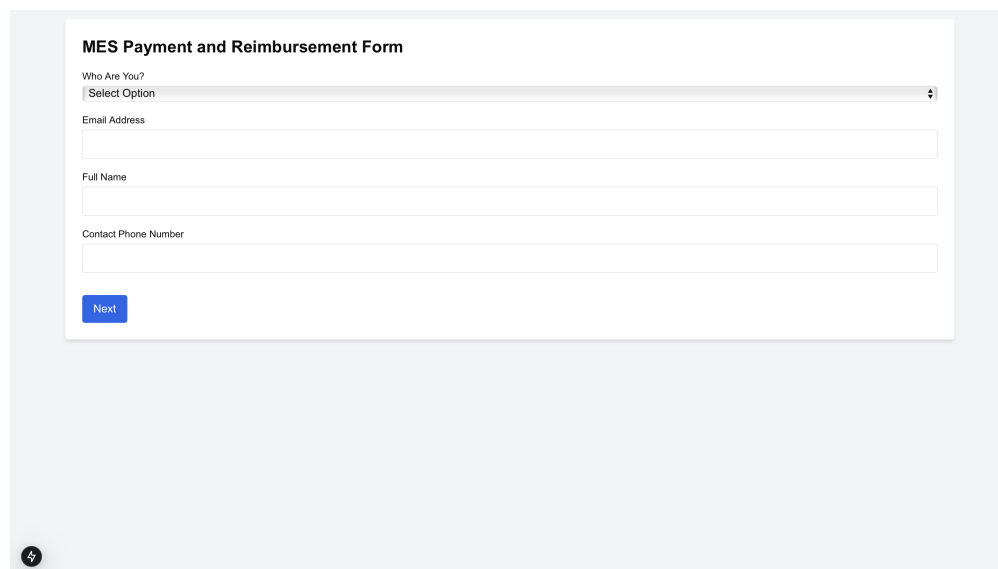


The screenshot shows a web dashboard with a blue header bar containing 'Welcome, meserp15' and a 'Sign out' button. The main content area is titled 'Your Dashboard' and features a table with 8 columns: Full Name, Who Are You, Amount Requested (CAD), Group or Team Name, Payment Timeframe, Reimbursement or Payment, Timestamp, and Status. The table contains five rows of data, with the last row partially cut off.

Full Name	Who Are You	Amount Requested (CAD)	Group or Team Name	Payment Timeframe	Reimbursement or Payment	Timestamp	Status
Houam Alamour	Ratified Club, Team, or Program Society	67	Chem E Car	2024-11-13	Payment Request	2024-11-12, 11:02:12 p.m.	In Progress
Sufyan Motala	Student Projects and New Club Seed Funding	1224	Bachelor of Technology Association	2024-11-21	Reimbursement Request	2024-11-12, 11:13:57 p.m.	Submitted
ads	Ratified Club, Team, or Program Society	12341	Civil Engineering Society	2024-11-13	Reimbursement Request	2024-11-13, 12:17:03 p.m.	Submitted
jkknkk	Ratified Club, Team, or Program Society	55555	Chem Eng Society	2024-11-14	Reimbursement Request	2024-11-13, 12:18:22 p.m.	Submitted
	Ratified Club Team				Reimbursement	2024-11-13	

Figure 2: Dashboard View

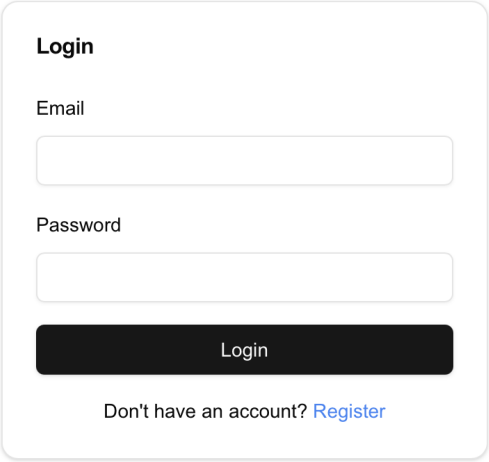
### 10.2 Form



The screenshot shows a form titled 'MES Payment and Reimbursement Form'. It contains a dropdown menu for 'Who Are You?' with 'Select Option' as the current selection. Below the dropdown are four text input fields for 'Email Address', 'Full Name', and 'Contact Phone Number'. A blue 'Next' button is located at the bottom left of the form. A small circular icon with a plus sign is visible in the bottom left corner of the page.

Figure 3: Form View

## 10.3 Login



The image shows a login form within a light gray rounded rectangle. At the top left of the form is the title "Login" in bold. Below it are two input fields: "Email" and "Password". The "Email" field is a simple white rectangle with a thin gray border. The "Password" field is a white rectangle with a thin gray border and a small gray eye icon on the right side to toggle visibility. Below the password field is a dark gray button with the text "Login" in white. At the bottom of the form, there is a link that says "Don't have an account? [Register](#)".

**Login**

Email

Password

Login

Don't have an account? [Register](#)

Figure 4: Login View

## 10.4 Receipt Input

Welcome, meserp15 [Sign out](#)

### Your Dashboard

#### Upload Your Photos

First Name  Last Name

ex: myname@example.com  
example@example.com

0/50

Drag and drop files here

Full Name	Ratified	Club	or Project	Status
Houam	11-			In Progress
Alamour	12			
Sufyan	11-			Submitted
Motale	57			
ads	11-			Submitted
jkkkkk	03			
	11-			Submitted
	22			

Figure 5: Receipt Scanning View

## 10.5 Wesbsite Tutorial

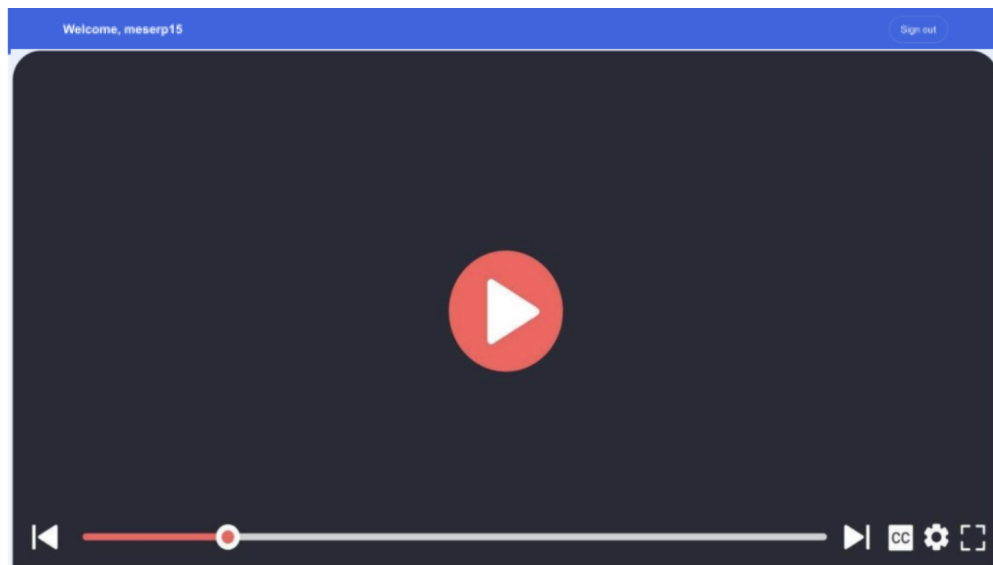


Figure 6: Tutorial Page

## 11 Design of Communication Protocols

- **APIs:** APIs will be used to communicate between the backend and the frontend of the application
- **Email Registration:** When creating an account there will be email authentication to ensure valid users
- **Email communication:** When a reimbursement request is made or edited, the correct groups will be notified via email

## 12 Timeline

- **Team Formed, Project Selected:** September 16
  - **All Team Members (Omar, Taaha, Rachid, Sufyan, Housam):** Establish initial roles, select project scope, and identify stakeholder needs.
- **Problem Statement, POC Plan, Development Plan:** September 23
  - **All Team Members:** Finalize the problem statement for the MES-ERP system. Outline Proof-of-Concept (POC) objectives and develop a high-level development plan.
  - **Sufyan & Omar:** Draft initial architecture sketches for key modules (Hardware-Hiding, Behaviour-Hiding, Software Decision).
- **Requirements Document Revision 0:** October 9
  - **Housam & Rachid:** Gather and refine user stories for Reimbursement Submission, Reimbursement Review, Budget Management, and Notification System.
  - **Taaha:** Integrate feedback on Data Validation and Database Module requirements.
  - **All Team Members:** Review and finalize the Requirements Specification for Rev 0.
- **Hazard Analysis 0:** October 23
  - **Omar & Sufyan:** Identify potential failure points for the Data Validation Module (e.g., corrupted or missing receipts) and the Database Module (e.g., concurrency issues).
  - **Rachid:** Document mitigations for critical hazards in the Reimbursement Submission workflow (e.g., incorrect approvals).
- **V&V Plan Revision 0:** November 1

- **Housam & Taaha:** Draft initial testing approach for each module:
  - \* **Hardware-Hiding Module:** Verify database connections and file I/O mocks.
  - \* **Behaviour-Hiding Module:** Unit tests for Reimbursement Submission, Notifications, and Budget Dashboard.
  - \* **Software Decision Module:** Automated tests for Data Validation, CI/CD, and Test Automation Framework.
- **All Team Members:** Incorporate plan into the project schedule.
- **Proof of Concept Demonstration:** November 11–22
  - **Omar, Taaha, Housam:** Implement a simplified Reimbursement Submission + Data Validation flow (happy path) for the demonstration.
  - **Rachid & Sufyan:** Set up the Database Module with mock data; create a minimal GUI prototype for user interaction.
  - **All Team Members:** Conduct internal testing before the demonstration.
- **Design Document Revision 0:** January 17
  - **Omar & Housam:** Document final architecture decisions for each module:
    - \* **Hardware-Hiding Module:** Database and environment specifics.
    - \* **Behaviour-Hiding Module:** Detailed class diagrams for Reimbursement, Notifications, and GUI.
    - \* **Software Decision Module:** Data Validation logic and CI/CD pipeline configuration.
  - **Rachid & Sufyan:** Cross-check design artifacts with SRS and POC feedback.
  - **Taaha:** Integrate all design sections into a cohesive Revision 0 document.
- **Revision 0 Demonstration:** February 3–February 14
  - **GUI Module (Rachid & Sufyan):** Build an interactive front-end for Reimbursement Submission, including field-level validation feedback.
  - **Data Validation Module (Taaha):** Finalize core validation checks (e.g., correct file formats, mandatory receipt uploads).
  - **Reimbursement Review & Approval (Omar):** Implement the basic approval workflow with role-based permissions.
  - **Budget Dashboard (Housam):** Provide partial functionality (view-only) and real-time data fetch from the Database Module.
  - **All Team Members:** Present Rev 0 to stakeholders, focusing on end-to-end demonstration of a reimbursement request.
- **V&V Report Revision 0:** March 7

- **Taaha & Omar:** Compile test results from module-level tests, including pass/fail statistics and coverage metrics.
- **Sufyan, Housam, Rachid:** Validate the correctness of reported bugs and retest major fixes in Reimbursement Submission and Budget Dashboard.
- **Final Demonstration (Revision 1):** March 24–March 30
  - **All Team Members:** Incorporate feedback from Revision 0 and finalize all modules.
  - **GUI Enhancements (Sufyan & Rachid):** Polish front-end UX and incorporate advanced form validations for user-friendly error handling.
  - **Notifications System (Housam):** Integrate push notifications and advanced triggers (e.g., overdue approvals).
- **Add Receipt Scanning/Image Processing:** March 24–March 30
  - **Omar, Rachid, Housam:** Incorporate an OCR (Optical Character Recognition) service in the Reimbursement Submission module.
  - **Taaha:** Update the Data Validation Module to handle scanned data edge cases (e.g., partial text scans).
- **Add User Manual to Application:** March 24–March 30
  - **Sufyan:** Create comprehensive end-user documentation and how-to guides, focusing on reimbursements and approvals.
  - **All Team Members:** Review for clarity and completeness before final release.
- **Refine the UI, Functions, and Backend Connectivity:** March 24–March 30
  - **Taaha:** Implement advanced error handling across all modules and polish back-end endpoints for reliability.
  - **All Team Members:** Conduct integration testing for the updated features.
- **Reach Out to MES Rep (Weekly):** March 24–March 30
  - **Omar, Taaha, Rachid, Sufyan, Housam:** Schedule status meetings, collect user feedback, and prioritize final tweaks for Rev 1.
- **EXPO Demonstration:** April (TBD)
  - **All Team Members:** Showcase the fully functional application, highlighting compliance auditing, budgeting, and user flows.
- **Final Documentation (Revision 1):** April 2

- **All Team Members:** Update the Design Document, MG, MIS, V&V Report, and User Manual with final revisions and new features.
- **Omar & Taaha:** Proofread overall consistency and finalize submission to the department.

## References

David L. Parnas. On the criteria to be used in decomposing systems into modules. *Comm. ACM*, 15(2):1053–1058, December 1972.

David L. Parnas. Designing software for ease of extension and contraction. In *ICSE '78: Proceedings of the 3rd international conference on Software engineering*, pages 264–277, Piscataway, NJ, USA, 1978. IEEE Press. ISBN none.

D.L. Parnas, P.C. Clement, and D. M. Weiss. The modular structure of complex systems. In *International Conference on Software Engineering*, pages 408–419, 1984.