Software Design Document (SDD)

# Project Title: Campus Expense Reimbursement Portal (CERP)

# Software Design Document (SDD)

## Project Title: Campus Expense Reimbursement Portal (CERP)

## 1. Introduction

### 1.1 Purpose

This Software Design Document (SDD) provides a comprehensive design solution for the Campus Expense Reimbursement Portal (CERP). It is intended to guide the development, implementation, and maintenance of the system by detailing its architecture, data design, and component-level structure.

### 1.2 Scope

This document covers the detailed software design aspects of the system including architectural diagrams, database schema, component breakdown, and interface design. It supports the implementation of the functionality and requirements as outlined in the SRS.

### 1.3 Intended Audience

* Development Team
* QA Engineers
* Project Stakeholders
* Maintenance Engineers

## 2. System Architecture Design

### 2.1 Architecture Overview

The system uses a three-tier architecture:

* Frontend: React.js (SPA)
* Backend: Spring Boot REST API
* Database: MySQL

### 2.2 Architecture Diagram

### 

## 3. Component Design

### 3.1 Frontend Components (React.js)

* LoginPage.jsx: Handles user authentication
* ClaimForm.jsx: Submits expense details and uploads documents
* Dashboard.jsx: Displays submitted claims and their statuses
* ApprovalPanel.jsx: Used by HOD and Principal to view, approve/reject claims

### 3.2 Backend Modules (Spring Boot)

* AuthController.java: Handles login and token generation
* ClaimController.java: Manages CRUD operations on claims
* ApprovalController.java: Manages HOD/Principal approvals
* NotificationService.java: Sends alerts
* AuditLogger.java: Logs all major events with timestamps and users

### 3.3 Services

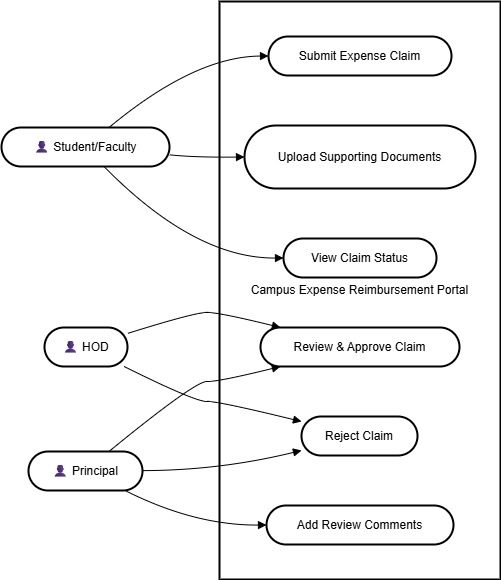
* ClaimService: Business logic for claim handling
* UserService: Authentication and authorization logic
* AuditService: Records actions into logs

**4. Data Design**

### 4.1 Database Schema (MySQL)

### 

## 5. Use Case Diagram



## 6. Sequence Diagram

## 

## 7. Security Design

* Use JWT tokens for authentication.
* Role-based authorization at controller level using Spring Security.
* Input validation and sanitization to prevent SQL Injection and XSS.
* Encrypted storage for uploaded documents.

## 8. Conclusion

The software design outlined in this document provides a modular, scalable, and maintainable architecture for the Campus Expense Reimbursement Portal. With clear separation between frontend, backend, and data layers, and well-defined modules and workflows, the system is engineered to meet functional and non-functional requirements efficiently. This SDD will serve as a blueprint for the implementation and ongoing enhancement of CERP.