Software Design Specifications

for

Employee Management System

Prepared by:

N.Chakrika-
Se22ucse064
K.Devisree-
Se22ucse080
E.Sai Aashrith-
Se22ucse090
G.Vamshi-
Se22ucse102
K.Lithin-
Se22ucse130
M.Meghana-
Se22ucse167
P.Aashritha-
Se22ucse215

Document Information

Title: Software Design Document - College Connect	Document Version No: 1.0
Project Manager: Nartakani Ma'am	Document Version Date: 09 April 2025
Prepared By: Team X	Preparation Date: 09 April 2025

Version History

Version no	Version Date	Revised By	Description	File Nmae
1.0	09 April 2025	Team X	Initial version of the document	SDD_College_Connect_TeamX

Table of Contents

1 INTRODUCTION 4

1.1 PURPOSE 4
1.2 SCOPE 4
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 4
1.4 REFERENCES 4

2 USE CASE VIEW 4

2.1 USE CASE 4

3 DESIGN OVERVIEW 4

3.1 DESIGN GOALS AND CONSTRAINTS 5
3.2 DESIGN ASSUMPTIONS 5
3.3 SIGNIFICANT DESIGN PACKAGES 5
3.4 DEPENDENT EXTERNAL INTERFACES 5
3.5 IMPLEMENTED APPLICATION EXTERNAL INTERFACES 5

4 LOGICAL VIEW 5

4.1 DESIGN MODEL 6
4.2 USE CASE REALIZATION 6

5 DATA VIEW 6

5.1 DOMAIN MODEL 6
5.2 D ATA MODEL (PERSISTENT DATA VIEW) 6
5.2.1 Data Dictionary 6

6 EXCEPTION HANDLING 6

7 CONFIGURABLE PARAMETERS 6

8 QUALITY OF SERVICE 7

8.1 AVAILABILITY 7
8.2 SECURITY AND AUTHORIZATION 7
8.3 LOAD AND PERFORMANCE IMPLICATIONS 7
8.4 MONITORING AND CONTROL 7

1.1 Purpose

This document describes the architecture, components, and system design for the Employee Management System (EMS).

It provides a detailed view of how the system is structured and how its components interact to fulfill the functional requirements
of managing employee data, payroll, attendance, and performance.

1.2 Scope

The Employee Management System is a web-based platform designed to handle employee-related functions such as:

- Managing employee profiles.
- Processing payroll.
- Tracking attendance and leaves.
- Evaluating employee performance.
- Facilitating employee feedback and grievances.

1.3 Definitions, Acronyms, and Abbreviations

Employee Management System (EMS) - The system being described.

1.4 References

Employee Management System Requirements Specification Document

2. Use Case View

2.1 Use Case

Key use cases for EMS include:

- Employee Login: Allows employees to log in and view their profiles, payroll, and attendance records.
- Manage Employee Profiles: HR Admin can add, update, or remove employee records.
- Process Payroll: HR Admin processes payroll for employees based on their attendance and other criteria.
- Attendance Management: Employees can mark attendance, request leave, and view attendance reports.

3. Design Overview

3.1 Design Goals and Constraints

The key goals include:

- Secure login and data handling.
- Clear separation of roles (HR Admin, Employee).
- Real-time attendance updates.

Constraints:

- Technology stack includes PostgreSQL for the database and Express.js for the backend API.
- The initial deployment is on a LAN or locally for internal use.

3.2 Design Assumptions

Assumptions include:

- Employees and HR Admins are distinct users.
- Employees access the system via a modern web browser.
- Payroll processing follows standard guidelines and regulations.

3.3 Significant Design Packages

The system is decomposed into three layers:

1. Frontend (UI) - HTML, CSS, and JavaScript.
2. Backend (API) - Express.js for handling business logic.
3. Database (Data Layer) - PostgreSQL for storing employee and payroll data.

3.4 Dependent External Interfaces

The system depends on the following external interfaces:

- PostgreSQL Database: Stores employee data, payroll details, attendance records.
- Email Service: Sends payroll details and attendance notifications to employees.

The table below lists the public interfaces this design requires from other modules or applications.

External Application and Interface Name	**Module Using the Interface**	**Functionality/ Description**
PostgreSQL Database (Employee Data Interface)	Employee Management Module	Used to store and retrieve employee details, attendance records, and payroll data.
Email Service (Notification API)	Notification Module	Sends payroll details, attendance notifications, and HR communications to employees.

3.5 Implemented Application External Interfaces (and SOA web services)

The table below lists the implementation of public interfaces this design makes available for other applications.

Interface Name	**Module Implementing the Interface**	**Functionality/ Description**
Employee Data Service API	Employee Management Module Payroll Module	Provides employee data (name, ID, role, attendance, etc.) for integration with HR tools or reporting dashboards.

Payroll Summary Service		Exposes processed payroll summaries to external financial or auditing systems.

4. Logical View

4.1 Overview

The Employee Management System (EMS) is structured using a modular and layered architecture that separates concerns across different system functionalities. The architecture supports different actors — Employee, HR Admin, and Manager — and allows scalable implementation of core features like attendance tracking, leave management, task assignment, performance evaluation, and payroll processing.

The design ensures loose coupling and high cohesion by grouping functionalities into dedicated modules/services. Each actor interacts with the system through clearly defined interfaces that map to business use cases. The system utilizes a standard three-tier architecture:

- Presentation Layer (UI): Web interface for users to interact with the system.
- Application Layer (Controllers/Services): Handles business logic and user workflows.
- Data Layer (Repositories/DAOs): Manages persistent data access and integrity.

4.2 Design Model

The system is decomposed into the following modules and key classes/components:

Modules & Components

1. 1. Authentication Module

   • Class: UserLogin

   - - Attributes: username, password, role
   - - Operations: validateUser(), redirectToDashboard()

   • Class: SessionManager

   - - Attributes: userID, sessionToken, expiry
   - - Operations: startSession(), validateSession(), endSession()

2. 2. Employee Profile Module

   • Class: Employee

   - - Attributes: employeeID, name, email, phone, department, role, joinDate
   - - Operations: viewProfile(), updateProfile()

   • Class: ProfileHandler

- - Operations: fetchEmployeeDetails(), updateDetails()

3. 3. Attendance Management Module

   • Class: Attendance

   - - Attributes: attendanceID, employeeID, date, clockIn, clockOut, status
   - - Operations: clockIn(), clockOut(), markAttendance()

   • Class: WorkHoursTracker

   - - Operations: calculateDailyHours(), generateAttendanceReport()

4. 4. Leave Management Module

   • Class: LeaveApplication

   - - Attributes: leaveID, employeeID, startDate, endDate, leaveType, status, reason
   - - Operations: applyLeave(), cancelLeave(), trackLeaveStatus()

   • Class: LeaveService

   - - Operations: validateLeaveDates(), approveLeave(), rejectLeave()

5. 5. Task Management Module

   • Class: TaskAssignment

   - - Attributes: taskID, assignedTo, assignedBy, description, deadline
   - - Operations: assignTask(), updateTaskDetails(), deleteTask()

   • Class: TaskUpdate

   - - Attributes: updateID, taskID, submissionDate, status, comment
   - - Operations: submitTaskUpdate(), viewTaskStatus()

6. 6. Performance Evaluation Module

   • Class: PerformanceReview

   - - Attributes: reviewID, updateID, score, feedback, evaluatedBy
   - - Operations: evaluateTask(), viewReview()

   • Class: EvaluationService

   - - Operations: generateEvaluationForm(), calculatePerformanceScore()

7. 7. Payroll and HR Reporting Module

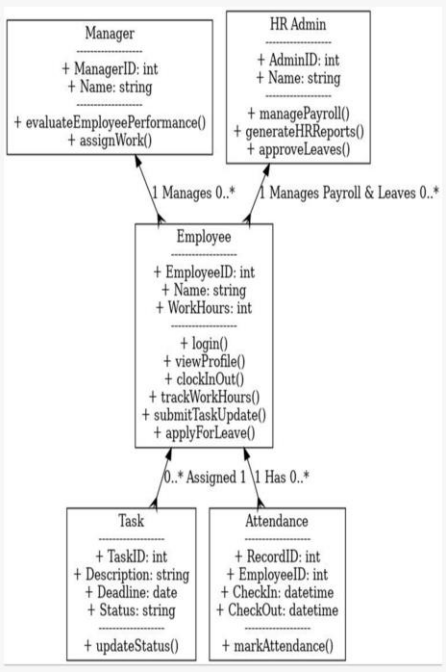
• Class: Payroll

- - Attributes: payrollID, employeeID, month, grossSalary, deductions, netSalary
- - Operations: generatePayroll(), calculateSalary()

• Class: HRReport

- - Attributes: reportID, reportType, generatedOn
- - Operations: generateReport(), exportToPDF()

Class Diagram



4.3 Use Case Realization

Use Case: Clock In / Clock Out

Actors: Employee
Flow:
1. Employee selects clock-in via the UI.
2. The UI sends a request to AttendanceController.
3. AttendanceService validates and saves the timestamp in the database.
4. Triggers TrackWorkHours() to update total hours.

Use Case: Apply for Leave

Actors: Employee, HR Admin
Flow:
1. Employee fills a leave form in the portal.
2. LeaveService checks conflicts and validates dates.

3. LeaveApplication is stored with status "Pending".
4. HR Admin later updates the status after review.

Use Case: Submit Task Update

Actors: Employee
Flow:
1. Employee submits a task update.
2. TaskUpdateService logs the update.
3. Notifies manager and links update to a task ID.

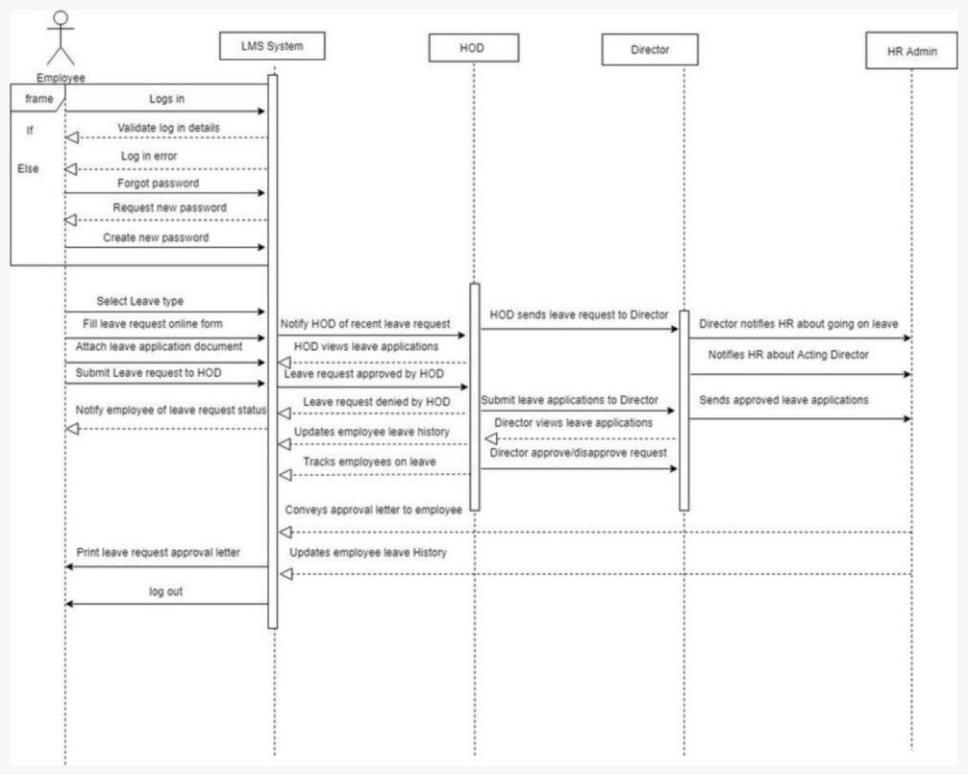
Use Case: Evaluate Employee Performance

Actors: Manager, HR Admin
Flow:
1. Manager accesses submitted task.
2. Evaluates task via EvaluationService.
3. Score and feedback are recorded in PerformanceReview.

Use Case: Manage Payroll and Generate HR Reports

Actors: HR Admin
Flow:
1. HR triggers payroll generation for the month.
2. PayrollService fetches attendance and leave data.
3. Salary is calculated and stored.
4. ReportService compiles data into downloadable HR reports.

Sequence Diagram



5 Data View

The Employee Management System is data-intensive and requires structured data persistence for managing employees, attendance, leave, payroll, and performance evaluations. This section explains the data structure in five key areas inspired by the use cases and actors shown in the UML diagram.

5.1 Domain Model

The domain model defines the core entities and their relationships in the EMS. These entities map directly to the real-world elements the system handles, such as employees, departments, leave records, tasks, and payroll.

Key Entities and Relationships:

Employee: Central entity linked to attendance, tasks, payroll, and leave.
Department: A grouping of employees with a department manager.
Attendance: Tracks daily presence with timestamps.
LeaveApplication: Manages employee leave requests and statuses.
TaskAssignment: Tasks assigned by managers to employees.
TaskUpdate: Employees' responses or submissions to tasks.
PerformanceReview: Records evaluations based on task completion.
Payroll: Stores monthly salary and deduction details.
User: Login credentials and role-based access.

Relationships:

One Department has many Employees.

One Employee can have many Attendance, LeaveApplications, TaskAssignments, and Payrolls.
One TaskAssignment can have many TaskUpdates.
One TaskUpdate has one PerformanceReview.

5.2 Data Model (persistent data view)

T he data model represents the physical schema of how data is stored in the database. The model uses relational tables, ensuring data normalization and referential integrity.
Primary Tables and Key Attributes:
Employee
EmployeeID (PK)
Name, Email, Phone, DepartmentID, JoinDate, Role, Salary
Department
DepartmentID (PK)
DepartmentName, ManagerID (FK → Employee)
Attendance
AttendanceID (PK)
EmployeeID (FK), Date, ClockIn, ClockOut, Status
LeaveApplication
LeaveID (PK)
EmployeeID (FK), StartDate, EndDate, LeaveType, Status, Reason
TaskAssignment
TaskID (PK)
AssignedBy (ManagerID FK), AssignedTo (EmployeeID FK), Deadline, Description
TaskUpdate
UpdateID (PK)
TaskID (FK), EmployeeID (FK), Status, SubmissionDate, Comment
PerformanceReview
ReviewID (PK)
UpdateID (FK), Score, Feedback, EvaluatedBy (FK)
Payroll
PayrollID (PK)
EmployeeID (FK), Month, GrossSalary, Deductions, NetSalary, GeneratedDate
User
UserID (PK)
Username, PasswordHash, Role, EmployeeID (nullable, FK)

5.2.1 Data Dictionary

Entity	Field Name	Data Type	Description
Employee	EmployeeID	INT (PK)	Unique employee identifier
	Name	VARCHAR(100)	Full name of the employee
	Email	VARCHAR(100)	Unique email ID
	Phone	VARCHAR(15)	Contact number
	DepartmentID	INT (FK)	Linked to Department
	JoinDate	DATE	Date the employee joined
	Role	VARCHAR(50)	Role or designation in the company
	Salary	DECIMAL(10,2)	Monthly base salary
Department	DepartmentID	INT (PK)	Unique department identifier
	DepartmentName	VARCHAR(100)	Name of the department
	ManagerID	INT (FK)	Head of the department
Attendance	AttendanceID	INT (PK)	Unique attendance record
	EmployeeID	INT (FK)	Linked to Employee
	Date	DATE	Date of attendance
	ClockIn	TIME	Time employee clocks in
	ClockOut	TIME	Time employee clocks out
	Status	ENUM	Present / Absent / Half Day
LeaveApplication	LeaveID	INT (PK)	Unique leave request ID
	EmployeeID	INT (FK)	Requesting employee
	StartDate	DATE	Start date of leave
	EndDate	DATE	End date of leave
	LeaveType	VARCHAR(50)	Type of leave (Casual, Sick, etc.)
	Status	ENUM	Pending / Approved / Rejected
	Reason	TEXT	Reason for leave
TaskAssignment	TaskID	INT (PK)	Unique task identifier
	AssignedBy	INT (FK)	Manager assigning the task
	AssignedTo	INT (FK)	Employee assigned

			to the task
	Deadline	DATE	Submission deadline
	Description	TEXT	Description of the task
TaskUpdate	UpdateID	INT (PK)	Task update record ID
	TaskID	INT (FK)	Linked to assigned task
	EmployeeID	INT (FK)	Submitting employee
	SubmissionDate	DATE	Date task update was submitted
	Status	ENUM	Submitted / Pending / Late
	Comment	TEXT	Notes or comments about the submission
PerformanceReview	ReviewID	INT (PK)	Evaluation record ID
	UpdateID	INT (FK)	Related task update
	Score	INT	Evaluation score
	Feedback	TEXT	Manager feedback/comments
	EvaluatedBy	INT (FK)	Evaluator (Manager or HR)
Payroll	PayrollID	INT (PK)	Unique payroll record
	EmployeeID	INT (FK)	Employee paid
	Month	VARCHAR(10)	Format: YYYY-MM
	GrossSalary	DECIMAL(10,2)	Total before deductions
	Deductions	DECIMAL(10,2)	Total deductions
	NetSalary	DECIMAL(10,2)	Final salary after deductions
	GeneratedDate	DATE	Date of payroll processing
User	UserID	INT (PK)	Login ID
	Username	VARCHAR(100)	Unique login name
	PasswordHash	TEXT	Encrypted password
	Role	ENUM	Admin / HR / Employee
	EmployeeID	INT (Nullable, FK)	Linked employee profile (if applicable)

6 Exception Handling

The Employee Management System incorporates a robust exception handling mechanism to ensure system reliability, maintain data integrity, and enhance user experience. Exceptions are categorized broadly into business logic errors and system-level failures. Common exceptions include invalid login attempts, unauthorized access, duplicate leave applications, late task submissions, and missing records. For example, if an employee submits a leave request that overlaps with existing approved leaves, a DuplicateLeaveException is raised, prompting the user with a clear validation message. Similarly, attempting to submit a task after its deadline triggers a LateTaskSubmissionException, which flags the submission during performance evaluation.

System-level exceptions such as DatabaseConnectionException are gracefully managed by retry mechanisms and fallback messages like "Please try again later." All exceptions are logged using a centralized logging framework, capturing essential details like timestamps, user ID, module, and stack trace. Errors are surfaced to users through simple and non-technical messages, often accompanied by recovery options such as retrying the action or returning to the dashboard. This ensures that users remain informed without exposing internal application details. Critical operations like attendance logging and task submissions are supported with retry queues and alerts to ensure continuity in case of transient failures. Overall, the exception handling design ensures that the system remains stable, secure, and user-friendly even in the face of unexpected events.

7 Configurable Parameters

This table describes the simple configurable parameters (name / value pairs).

Configuration Parameter Name	Definition and Usage	Dynamic?
MaxDailyWorkHours	Specifies the maximum number of hours considered for full attendance in a day.	Yes
LeaveApprovalWindowDays	Sets the time limit (in days) within which HR must respond to a leave application.	Yes
PayrollProcessingDate	Determines the day of each month when payroll is auto-generated.	Yes
TaskSubmissionGracePeriod	Defines the number of extra days allowed for late task submissions.	Yes
AllowedFileSizeMB	Sets the maximum upload file size (in MB) for documents like task reports or leave forms.	Yes
LoginSessionTimeoutMinutes	Specifies session timeout duration in minutes after inactivity.	Yes
MaxLeaveDaysPerMonth	Limits the number of leave days an employee can take each month.	Yes
EnableEmailNotifications	Toggles whether email alerts are sent for tasks, approvals, and login activities.	Yes
EvaluationScoreRange	Defines the scoring range used in performance evaluations (e.g., 1–5 or 1–10).	No

8.Quality of Service

8.1 Availability

System is designed for 99.9% uptime.

• Failover strategies include automatic DB backup and redundant server nodes

8.2 Security and Authorization

• Data encryption using HTTPS and JWT.
• Role-based access (employee, HR, admin).
• Passwords stored using hashing (bcrypt).

8.3 Load and Performance Implications

• Can handle 500 concurrent users.
• Indexed database queries for fast lookups.
• Load testing tools like JMeter used to validate performance.

8.4 ,Monitoring and Control

• Logs monitored through services like LogRocket or ELK stack.
• Performance metrics tracked (response time, CPU usage, request count).
• Alerts configured for server downtime, login failures, high response time.