Project: Employee Shift Management System

1. Introduction

This document outlines the Low-Level Design (LLD) for an **Employee Shift Management System**, which helps organizations efficiently schedule employee work shifts, manage leaves, and track working hours.

The system supports both **Spring Boot (Java)** and **ASP.NET Core (.NET)** for backend development.

2. Module Overview

2.1 Employee Management Module

- Enables the creation and management of employee records.
- Assigns employees to different departments and roles.

2.2 Shift Scheduling Module

- Allows managers to create, update, and delete shift schedules.
- Supports different types of shifts (morning, evening, night).

2.3 Leave Management Module

- Employees can request leaves, and managers can approve or reject them.
- Tracks leave balances and generates reports.

2.4 Attendance Tracking Module

- Employees can check in and check out from shifts.
- Tracks working hours for payroll and compliance.

2.5 Notification Module

- Sends shift reminders to employees.
- Alerts managers about conflicts or pending approvals.

3. Architecture Overview

3.1 Architectural Style

- **Frontend:** Angular or React.
- **Backend:** REST API-based architecture.

- **Database:** Relational Database (MySQL/PostgreSQL/SQL Server).

3.2 Component Interaction

- The frontend interacts with the backend through REST APIs.
- The backend connects to the database for data storage and retrieval.

4. Module-Wise Design

4.1 Employee Management Module

4.1.1 Features

- Manage employee profiles.
- Assign roles and departments.

4.1.2 Entities

- **Employee**
  - EmployeeID
  - Name
  - DepartmentID
  - Role
  - ContactDetails

4.2 Shift Scheduling Module

4.2.1 Features

- Create and modify shifts.
- Prevent overlapping shift assignments.

4.2.2 Entities

- **Shift**
  - ShiftID
  - EmployeeID
  - StartTime
  - EndTime
  - ShiftType (Morning/Evening/Night)

4.3 Leave Management Module

4.3.1 Features

- Employees can apply for leaves.
- Managers approve or reject leave requests.

4.3.2 Entities

- **LeaveRequest**
  - LeaveID
  - EmployeeID
  - StartDate
  - EndDate
  - Status (Pending/Approved/Rejected)

4.4 Attendance Tracking Module

4.4.1 Features

- Tracks employee check-in and check-out times.
- Generates monthly attendance reports.

4.4.2 Entities

- **Attendance**
  - AttendanceID
  - EmployeeID
  - Date
  - CheckInTime
  - CheckOutTime

4.5 Notification Module

4.5.1 Features

- Sends automated shift reminders.
- Alerts employees about approved or rejected leave requests.

4.5.2 Entities

- **Notification**

    - NotificationID
    - UserID
    - Type (Email/SMS)
    - MessageContent

5. Deployment Strategy

5.1 Local Deployment

- Frontend and backend deployed on developer machines for testing.

5.2 Testing Environments

- Uses containerized setups for staging environments.

6. Database Design

6.1 Tables and Relationships

- **Employee**: Primary Key: EmployeeID.
- **Shift**: Primary Key: ShiftID, Foreign Key: EmployeeID.
- **LeaveRequest**: Primary Key: LeaveID, Foreign Key: EmployeeID.
- **Attendance**: Primary Key: AttendanceID, Foreign Key: EmployeeID.

7. User Interface Design

7.1 Wireframes

- **Dashboard**: Displays upcoming shifts and notifications.
- **Shift Management**: Allows managers to assign shifts.
- **Leave Request Page**: Employees can apply for leaves.

8. Non-Functional Requirements

8.1 Performance

- Capable of managing shifts for up to **5,000 employees** concurrently.

8.2 Usability

- User-friendly UI designed for HR managers and employees.

8.3 Security

- Secure authentication with **role-based access control**.

8.4 Scalability

- Supports multiple departments and locations.

9. Assumptions and Constraints

9.1 Assumptions

- Employees will use a web-based portal for shift management.

9.2 Constraints

- Initial phase limited to a **single organization** before expansion.