Project: Employee Leave and Attendance Management System

1. Introduction

This document outlines the Low-Level Design (LLD) for an **Employee Leave and Attendance Management System**, which automates the process of tracking employee attendance, managing leave requests, and calculating working hours.

This design supports both **Java (Spring Boot)** and **.NET (ASP.NET Core)** frameworks for backend development.

2. Module Overview

2.1 Employee Attendance Module

- Allows employees to mark attendance, view their attendance history, and track daily clock-in and clock-out times.
- Provides managers with a dashboard to monitor attendance records and detect irregularities.

2.2 Leave Management Module

- Employees can request different types of leave (sick leave, vacation, etc.) and track the status of leave requests.
- Provides managers with a system for approving or rejecting leave requests.

2.3 Leave Balance Module

- Tracks and updates the leave balances for each employee based on the type of leave and the company's leave policies.
- Automatically adjusts leave balances as leaves are approved or rejected.

2.4 Shift Management Module

- Allows the assignment of shifts to employees based on company scheduling needs.
- Employees can view and swap shifts based on availability and manager approval.

2.5 Reports and Analytics Module

- Generates reports on attendance trends, leave usage, and shift coverage.

- Provides insights into workforce availability and helps in identifying patterns in leave requests or attendance issues.

3. Architecture Overview

3.1 Architectural Style

- **Frontend**: Angular or React for creating a dynamic user interface.
- **Backend**: REST API-based architecture for handling all business logic and data processing.
- **Database**: Relational Database (MySQL/PostgreSQL/SQL Server) for storing employee attendance, leave, and shift data.

3.2 Component Interaction

- The frontend communicates with the backend through REST APIs to manage employee attendance, leave requests, and shift assignments.
- The backend handles data storage and retrieval from the database.

4. Module-Wise Design

4.1 Employee Attendance Module

4.1.1 Features

- Employees can clock in and out for each workday.
- Attendance records are stored in real-time and can be accessed by both employees and managers.

4.1.2 Data Flow

- Employees log their attendance via the frontend.
- The backend processes the attendance data and stores it in the database. Managers can access the records in real-time.

4.1.3 Entities

- **Attendance**
  - EmployeeID
  - ClockInTime
  - ClockOutTime
  - WorkHours

4.2 Leave Management Module

4.2.1 Features

- Employees can request leaves by specifying the type and duration of the leave.
- Managers can approve or reject leave requests.

4.2.2 Entities

- **LeaveRequest**
    - LeaveID
    - EmployeeID
    - LeaveType (Sick, Vacation, etc.)
    - StartDate
    - EndDate
    - Status (Approved/Rejected)

4.3 Leave Balance Module

4.3.1 Features

- Automatically updates the employee's leave balance upon leave approval.
- Ensures that employees cannot request more leave than their available balance.

4.3.2 Entities

- **LeaveBalance**
    - EmployeeID
    - LeaveType
    - Balance

4.4 Shift Management Module

4.4.1 Features

- Managers can create and assign shifts to employees.
- Employees can view assigned shifts and request to swap shifts.

4.4.2 Entities

- **Shift**
    - ShiftID
    - EmployeeID

    - ShiftDate
    - ShiftTime

4.5 Reports and Analytics Module

4.5.1 Features

- Generate reports on attendance records, leave balances, and shift assignments.
- Provides analytics on leave trends and attendance patterns to management.

4.5.2 Entities

- **AttendanceReport**
    - ReportID
    - EmployeeID
    - DateRange
    - TotalAttendance
    - Absenteeism

5. Deployment Strategy

5.1 Local Deployment

- The system can be deployed locally for internal testing and review by HR or payroll teams.

5.2 Testing Environments

- Use containerized environments (Docker) for staging and quality assurance testing to ensure consistency.

6. Database Design

6.1 Tables and Relationships

- **Attendance**: Primary Key: AttendanceID, Foreign Key: EmployeeID.
- **LeaveRequest**: Foreign Key: EmployeeID.
- **LeaveBalance**: Foreign Key: EmployeeID.
- **Shift**: Foreign Key: EmployeeID.

7. User Interface Design

7.1 Wireframes

- **Attendance Dashboard**: Displays daily attendance logs, current attendance status, and history.
- **Leave Requests**: Allows employees to submit leave requests and view the status of previous requests.
- **Shift Schedule**: Displays upcoming shifts and allows employees to request shift swaps.

8. Non-Functional Requirements

8.1 Performance

- The system must handle the attendance logging for up to 1,000 employees simultaneously without significant delay.

8.2 Usability

- The user interface must be easy to use for both non-technical employees and HR personnel.

8.3 Security

- Implement role-based access control (RBAC) to ensure that only authorized personnel can approve leave requests or view sensitive data.

8.4 Scalability

- The system should scale to accommodate an increasing number of employees and leave records without requiring major infrastructure changes.

9. Assumptions and Constraints

9.1 Assumptions

- All employees have internet access and are familiar with basic computing.

9.2 Constraints

- Initially, the system will be limited to a single branch or location with fewer than 500 employees.