Human Resource Management System – Requirements Document

Day 2 – Design

# Project Overview

The Employee Management System (EMS) manages HR processes such as employee records, payroll, departments, attendance, and leave requests. - Relational SQL Database (via DBConnectionDAO) is used for structured, relational data: - Admin - Employee - Department - Payroll - Attendance - DynamoDB (via DynamoDBService) is used for flexible, fast storage of Leave Requests, since leave data can scale quickly and needs quick approval workflows. This hybrid approach allows the system to combine transactional integrity (SQL) with scalability (DynamoDB).

# Design Details

### Updated Class Diagram

Project

├── src

│ └── main

│ └── java

│ └── com.hrms.model

│ ├── dao

│ │ ├── AdminDAO.java

│ │ ├── AttendanceDAO.java

│ │ ├── DBConnection.java

│ │ ├── DepartmentDAO.java

│ │ ├── DynamoDBService.java

│ │ ├── EmployeeDAO.java

│ │ ├── LeaveRequestDAO.java

│ │ └── PayrollDAO.java

│ │

│ ├── service

│ │ ├── AdminApp.java

│ │ ├── EmployeeApp.java

│ │ ├── LeaveDynamoService.java

│ │ └── PayrollService.java

│ │

│ ├── Admin.java

│ ├── Attendance.java

│ ├── Department.java

│ ├── Employee.java

│ ├── LeaveRequest.java

│ └── Payroll.java

│

├── resources

└── test

### Workflow for Leave  Attendance

1. Employee applies leave  stored in DynamoDB (LeaveRequest). 2. Admin approves/rejects via DynamoDBService.updateStatus(). 3. If APPROVED, system automatically adds records into SQL Attendance Table with status = 'Leave' for each leave day.

* 1. ***SQL Database Schema***

## Admin Table:

CREATE TABLE Admin (

id INT PRIMARY KEY AUTO\_INCREMENT, username VARCHAR(50) NOT NULL UNIQUE, password VARCHAR(255) NOT NULL

);

## Employee Table:

CREATE TABLE Employee (

id INT PRIMARY KEY AUTO\_INCREMENT, name VARCHAR(100) NOT NULL,

department\_id INT, salary DECIMAL(10,2), email VARCHAR(100), phone VARCHAR(15),

date\_of\_joining DATE, password VARCHAR(255),

FOREIGN KEY (department\_id) REFERENCES Department(id)

);

## Department Table:

CREATE TABLE Department (

id INT PRIMARY KEY AUTO\_INCREMENT, name VARCHAR(100) NOT NULL,

location VARCHAR(100)

);

## Payroll Table:

CREATE TABLE Payroll (

id INT PRIMARY KEY AUTO\_INCREMENT,

employee\_id INT NOT NULL, basic DECIMAL(10,2),

hra DECIMAL(10,2),

allowances DECIMAL(10,2), deductions DECIMAL(10,2), net\_salary DECIMAL(10,2), date DATE,

FOREIGN KEY (employee\_id) REFERENCES Employee(id)

);

## Attendance Table:

CREATE TABLE Attendance (

id INT PRIMARY KEY AUTO\_INCREMENT,

employee\_id INT NOT NULL, date DATE NOT NULL,

status ENUM('Present','Absent','Leave') NOT NULL, FOREIGN KEY (employee\_id) REFERENCES Employee(id)

);

### DynamoDB Schema (NoSQL)

LeaveRequest Table (DynamoDB):

* + - Partition Key: requestId (String, UUID)
    - Sort Key: employeeId (Number)
    - Attributes:
      * startDate (String, ISO format)
      * endDate (String, ISO format)
      * reason (String)
      * status (String: PENDING/APPROVED/REJECTED)
      * employeeName (String)

# Deliverables

- UML Class Diagram including SQL DAOs + DynamoDBService - MySQL schema for Admin, Employee, Department, Payroll, Attendance - DynamoDB schema for LeaveRequest - Workflow showing Leave  Attendance synchronization