Project Specification: Employee Management System (EMS)

1. Project Overview

The goal of this project is to design and develop an **Employee Management System (EMS)** using **Spring Boot**.
The system will automate HR processes, manage employee data, and provide **role-based access** for different user types (**Admin**, **HR**, **Manager**, **Employee**).

The system will include:

- Role-based authentication and authorization.
- CRUD operations for employees, departments, roles, and leave requests.
- Performance and reporting modules.
- Secure data management using **Spring Security** and a **MySQL** database.

2. Objectives

- Provide a scalable, secure, and user-friendly HR management system.
- Automate employee data handling and administrative tasks.
- Implement **role-based access control** (Admin, HR, Manager, Employee).
- Allow HR users to manage employees and departments.
- Allow managers to view and evaluate their department's performance.
- Enable employees to view and update their personal information.
- Generate reports on employee performance and leave history.
- Ensure performance, security, and maintainability of the system.

3. Needs Analysis & Specification

3.1 Study of Existing Systems

An analysis of existing HR management systems (such as **BambooHR**, **Zoho People**, and **OrangeHRM**) was conducted to identify best practices and essential features, including:

- Centralized employee data management.
- Role-based dashboards and views.
- Automated leave request tracking.
- Performance and report generation tools.

3.2 Formalism

The project will be modeled using:

- **UML diagrams:** class diagram

4. Functional Requirements

4.1 Authentication & User Roles

- Secure login and registration using email and password.
- Role-based access control (RBAC) managed by **Spring Security**.
- User roles:
  - **Admin:** Manage all system data, including users, departments, and roles.
  - **HR:** Add and edit employee records, and manage leave requests.
  - **Manager:** View team performance and approve or reject leave requests.
  - **Employee:** View and update personal information and submit leave requests.

4.2 Employee Management

- Perform CRUD operations on employee profiles (name, phone, email, salary, department, role).
- Link each employee to a specific department and role.
- Allow document uploads (such as resumes).

4.3 Department & Role Management

- Admin or HR users can create, modify, and delete departments and roles.
- Departments have **one-to-many** relationships with employees.

4.4 Leave Management

- Employees can submit leave requests specifying type, duration, and reason.
- Managers review and approve or reject requests.
- System maintains a complete leave history for each employee.

4.5 Performance & Reporting

- HR and Managers can evaluate employee performance.
- Generate reports in **PDF or CSV** formats such as:
    - Employee lists
    - Department summaries
    - Performance and salary statistics

5. Non-Functional Requirements

- **Performance:** fast CRUD operations.
- **Scalability:** The system must support more than 100 concurrent users.
- **Security:** Passwords encrypted using **BCrypt**, secure HTTPS communication, and session management.
- **Usability:** Simple and responsive user interface .
- **Reliability:** proper exception handling.
- **Maintainability:** Modular layered architecture (Controller, Service, Repository).
- **Compatibility:** Works seamlessly across all major browsers.

6. Technical Stack

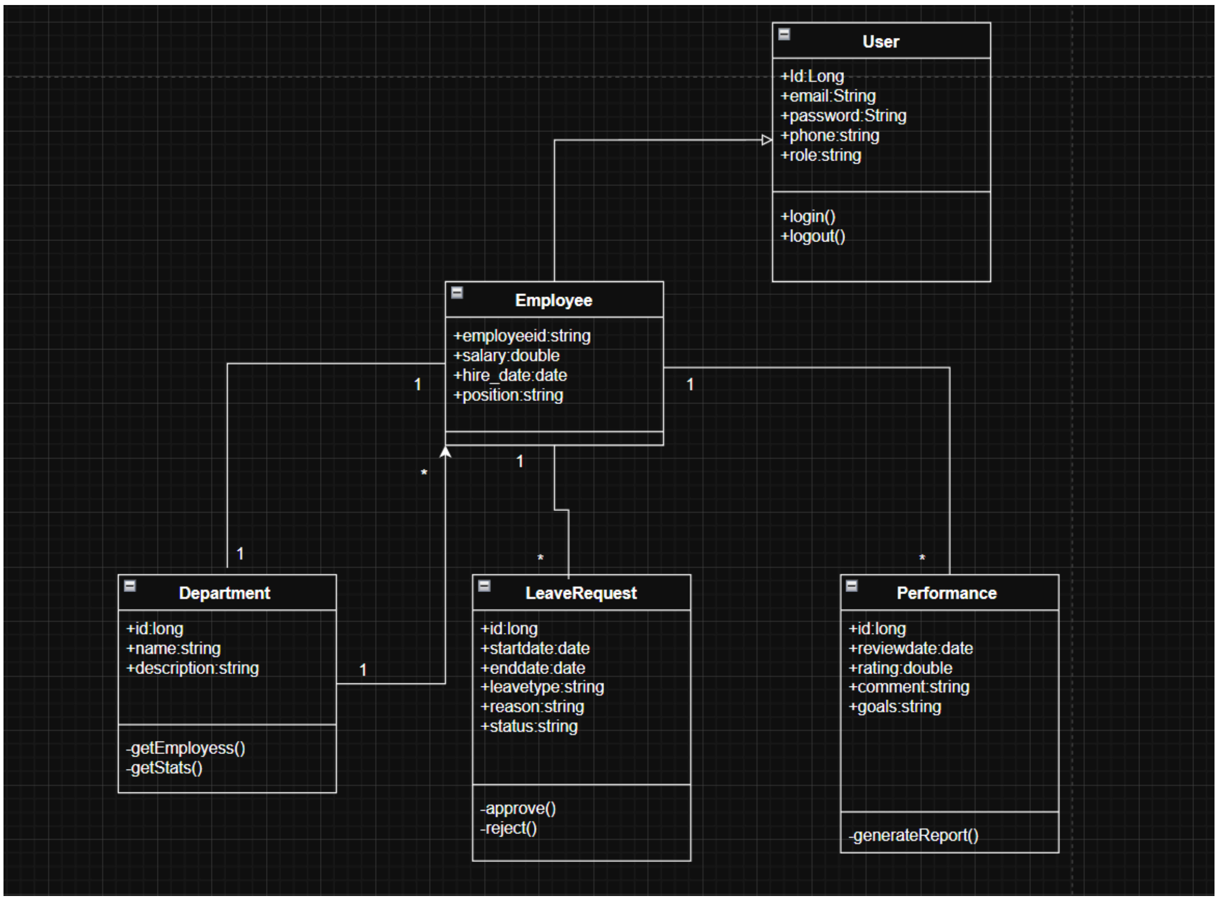
- **Backend Framework:** Spring Boot (Java)
- **Frontend:** Thymeleaf (html/bootstrap)
- **Database:** MySQL using **Spring Data JPA**
- **Authentication & Security:** Spring Security with JWT
- **Build Tool:** Maven

7. Working Environment

- **Languages:** Java, HTML, bootstrap, JavaScript
- **Tools:**
  - IntelliJ IDEA
  - Postman (API testing)
  - MySQL (database management)
  - Draw.io (UML diagrams)

Group Members

- Sameh Ghanmi
- Mohamed Dhia Chairet
- Firas Bouraoui
- Rayen Dakhli


User
+Id:Long
+email:String
+password:String
+phone:string
+role:string
+login()
+logout()
Employee
+employeeid:string
+salary:double
+hire_date:date
+position:string
1
1
1
*
1
1
*
*
Department
+id:long
+name:string
+description:string
-getEmployess()
-getStats()
LeaveRequest
+id:long
+startdate:date
+enddate:date
+leavetype:string
+reason:string
+status:string
-approve()
-reject()
Performance
+id:long
+reviewdate:date
+rating:double
+comment:string
+goals:string
-generateReport()