



Java Technologies for Web Applications

Project/Long Assignment

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/11/2020

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	25/Jun/2020	Create a new assignment	Create new	DieuNT1	VinhNV

Contents

Project/Long Assignment.....	4
1. TỔNG QUAN (OVERVIEW).....	4
1.1. Mục tiêu.....	4
1.2. Mô tả bài toán.....	4
2. THIẾT KẾ CƠ SỞ DỮ LIỆU (DATABASE SCHEMA)	4
2.1. Phân hệ Security (Quản lý Tài khoản & Quyền).....	4
2.2. Phân hệ HR (Quan hệ 1-N)	4
2.3. Phân hệ Project (Quan hệ N-N)	5
3. YÊU CẦU CHỨC NĂNG (FUNCTIONAL REQUIREMENTS).....	5
3.1. Module Authentication & Authorization (Security)	5
3.2. Module Department (Quản lý Phòng ban)	5
3.3. Module Employee (Quản lý Nhân viên)	5
3.4. Module Project (Quản lý Dự án & Phân công)	6
4. YÊU CẦU PHI CHỨC NĂNG (NON-FUNCTIONAL)	6
5. HƯỚNG DẪN KIẾN TRÚC DỰ ÁN (PROJECT ARCHITECTURE GUIDE)	6
5.1. Cấu trúc thư mục (Package Structure)	6
5.2. Tổ chức tài nguyên (Resources)	6
5.3. Các nguyên tắc (Best Practices)	7
6. YÊU CẦU KỸ THUẬT (TECHNOLOGY STACK)	8
7. THANG ĐIỂM ĐÁNH GIÁ (MARKING SCALE).....	8
8. HƯỚNG DẪN NỘP BÀI (SUBMISSION).....	9

CODE:	WEB-202_MiniProject_01 (HumanResource Management)
TYPE:	Project/Long Assignment
LOC:	N/A

Project/Long Assignment

1. TỔNG QUAN (OVERVIEW)

1.1. Mục tiêu

Bài tập lớn này nhằm mục đích kiểm tra và củng cố các kiến thức nâng cao về phát triển ứng dụng web với Java Spring Boot, bao gồm:

1. **Spring Security:** Xây dựng hệ thống xác thực (Authentication) và phân quyền (Authorization) dựa trên Role.
2. **Advanced Hibernate/JPA:** Xử lý các mối quan hệ phức tạp trong cơ sở dữ liệu: One-to-Many (1-N) và Many-to-Many (N-N).
3. **Thymeleaf & Bootstrap:** Xây dựng giao diện người dùng Responsive, xử lý Form binding phức tạp.
4. **Architecture:** Tuân thủ mô hình MVC (Model-View-Controller) và kiến trúc 3 lớp (Repository - Service - Controller).

1.2. Mô tả bài toán

Xây dựng hệ thống quản lý nội bộ cho một công ty phần mềm bao gồm 3 phân hệ chính:

1. **Quản lý Nhân sự (HR):** Quản lý nhân viên theo phòng ban.
2. **Quản lý Dự án (Projects):** Quản lý các dự án và phân công nhân sự vào dự án.
3. **Bảo mật (Security):** Quản lý tài khoản và quyền truy cập hệ thống.

2. THIẾT KẾ CƠ SỞ DỮ LIỆU (DATABASE SCHEMA)

Hệ thống yêu cầu thiết kế ERD với các bảng và mối quan hệ sau:

2.1. Phân hệ Security (Quản lý Tài khoản & Quyền)

- **Account:** Lưu thông tin đăng nhập.
 - account_id (PK, Identity)
 - username (Unique, Not null)
 - email (Unique, Not null)
 - password (Lưu chuỗi mã hóa BCrypt, **tuyệt đối không lưu text thuần**)
 - status (Active, Blocked)
- **Role:** Danh sách quyền hạn.
 - role_id (PK)
 - role_name (Ví dụ: ADMIN, MANAGER, EMPLOYEE)
- **Account_Role:** Bảng trung gian (Quan hệ N-N).
 - Một tài khoản có thể có nhiều quyền (Ví dụ: Một người vừa là Manager, vừa là Admin hệ thống).

2.2. Phân hệ HR (Quan hệ 1-N)

- **Department:**
 - department_id (PK)
 - dept_name (Not null, Unique - VD: "IT", "HR", "Sales")

- description
- **Employee:**
 - employee_id (PK)
 - full_name, birth_date, gender, phone, address
 - department_id (FK): Tham chiếu đến bảng Department (Một phòng ban có nhiều nhân viên).
 - account_id (FK): Tham chiếu đến bảng Account (Quan hệ 1-1).

2.3. Phân hệ Project (Quan hệ N-N)

- **Project:**
 - project_id (PK)
 - project_name (Not null)
 - start_date, end_date
 - status (Planning, Ongoing, Completed)
- **Project_Assignment (Bảng trung gian):**
 - assignment_id (PK)
 - project_id (FK)
 - employee_id (FK)
 - role_in_project (String - VD: "Developer", "Tester", "PM")
 - join_date (Ngày tham gia dự án)

3. YÊU CẦU CHỨC NĂNG (FUNCTIONAL REQUIREMENTS)

3.1. Module Authentication & Authorization (Security)

- **F1. Đăng nhập (Login):**
 - Sử dụng Spring Security Form Login.
 - Hiển thị thông báo lỗi nếu sai username/password hoặc tài khoản bị khóa.
 - Chuyển hướng người dùng về trang Dashboard tương ứng với quyền sau khi login.
- **F2. Phân quyền (RBAC):**
 - **ADMIN:** Toàn quyền hệ thống (CRUD User, Department, Project, Employee).
 - **MANAGER:** Được quản lý Dự án, Nhân viên, xem danh sách Phòng ban. Không được xóa tài khoản Admin khác.
 - **EMPLOYEE:** Chỉ được xem thông tin cá nhân (Profile) và danh sách dự án mình được phân công.
- **F3. Logout:** Đăng xuất an toàn, xóa session.

3.2. Module Department (Quản lý Phòng ban)

- **F4. CRUD Department (Admin only):**
 - Xem danh sách, Thêm, Sửa, Xóa phòng ban.
 - **Logic ràng buộc:** Không cho phép xóa Phòng ban nếu trong phòng ban đó vẫn còn nhân viên (Cần check count Employee trước khi delete).

3.3. Module Employee (Quản lý Nhân viên)

- **F5. Quản lý nhân viên (Admin/Manager):**
 - Hiển thị danh sách nhân viên có phân trang (Pagination).
 - **Thêm mới/Sửa nhân viên:**
 - Trường "Department" phải hiển thị dưới dạng **Dropdown List** (Select box) lấy dữ liệu từ DB.
 - Tự động tạo Account mặc định khi tạo mới Employee.
 - Tìm kiếm nhân viên theo Tên hoặc lọc theo Phòng ban.

3.4. Module Project (Quản lý Dự án & Phân công)

- **F6. Quản lý Dự án:**
 - Thêm mới dự án với validate: Ngày kết thúc (end_date) phải lớn hơn ngày bắt đầu (start_date).
 - Cập nhật trạng thái dự án.
- **F7. Phân công nhân sự (Project Assignment):**
 - Đây là tính năng quan trọng xử lý quan hệ **Many-to-Many**.
 - Trong màn hình chi tiết dự án (Project Detail), cung cấp form "Add Member":
 1. Chọn Nhân viên (Từ danh sách hoặc Search).
 2. Nhập vai trò (Role in Project).
 3. Bấm "Add" -> Lưu vào bảng Project_Assignment.
- **F8. Xem thành viên dự án:**
 - Hiển thị danh sách các nhân viên đang tham gia dự án đó ngay trong trang chi tiết dự án.

4. YÊU CẦU PHI CHỨC NĂNG (NON-FUNCTIONAL)

1. **Validation (Server-side):** Sử dụng Hibernate Validator (@Valid, @NotNull, @Size, @Email...).
 - Không được để trống các trường bắt buộc.
 - Email phải đúng định dạng.
2. **Giao diện (UI/UX):**
 - Sử dụng **Thymeleaf Layout Dialect** để cấu trúc trang (Header, Footer, Menu dùng chung).
 - Menu phải thay đổi (ẩn/hiện) dựa trên quyền của người dùng đang đăng nhập (Sử dụng sec:authorize).
 - Sử dụng Bootstrap 5 để đảm bảo giao diện đẹp và responsive.
3. **Exception Handling:**
 - Xử lý trang lỗi 403 (Access Denied) thân thiện.
 - Xử lý trang lỗi 404 (Not Found).

5. HƯỚNG DẪN KIẾN TRÚC DỰ ÁN (PROJECT ARCHITECTURE GUIDE)

Để đảm bảo tính bảo trì và mở rộng, dự án bắt buộc tổ chức theo mô hình phân lớp chuẩn (Layered Architecture).

5.1. Cấu trúc thư mục (Package Structure)

```
src/main/java/com/company/miniproject
├── config          // Chứa các class cấu hình
│   ├── SecurityConfig.java
│   └── MvcConfig.java
├── controller      // Xử lý request từ người dùng (Web Controller)
│   ├── AuthController.java
│   ├── DepartmentController.java
│   ├── EmployeeController.java
│   └── ProjectController.java
├── entity          // Các POJO class ánh xạ với Database (JPA Entities)
│   ├── Account.java
│   ├── Department.java
│   ├── Employee.java
│   └── Project.java
```

```
|   └── Role.java
├── repository    // Giao tiếp với Database (Interface extends JpaRepository)
|   ├── AccountRepository.java
|   ├── EmployeeRepository.java
|   └── ...
├── service       // Chứa Business Logic (Interface & Implementation)
|   ├── impl
|   |   ├── EmployeeServiceImpl.java
|   |   └── ProjectServiceImpl.java
|   ├── EmployeeService.java
|   └── ProjectService.java
├── security      // Các class hỗ trợ bảo mật
|   └── CustomUserDetailsService.java
├── dto           // Data Transfer Object (Dùng cho Form Data)
|   ├── ProjectAssignmentDto.java
|   └── EmployeeRegistrationDto.java
└── MiniprojectApplication.java
```

5.2. Tổ chức tài nguyên (Resources)

src/main/resources

```
├── static        // Tài nguyên tĩnh (CSS, JS, Images)
|   ├── css
|   ├── js
|   └── images
├── templates     // Thymeleaf Views
|   ├── fragments // Các phần dùng chung (Header, Footer, Sidebar)
|   ├── department // Các file HTML của module Department
|   ├── employee   // Các file HTML của module Employee
|   ├── project    // Các file HTML của module Project
|   ├── error      // Trang lỗi (403.html, 404.html)
|   ├── login.html
|   └── index.html // Dashboard
└── application.properties
```

5.3. Các nguyên tắc (Best Practices)

- **Controller:** Chỉ nhận Request, validate dữ liệu đầu vào cơ bản, gọi Service để xử lý và trả về View. **Không** viết logic nghiệp vụ (business logic) hoặc gọi Repository trực tiếp tại Controller.
- **Service:** Chứa toàn bộ logic nghiệp vụ, xử lý transaction (@Transactional).
- **DTO (Data Transfer Object):**
 - Khuyến khích sử dụng DTO để nhận dữ liệu từ Form (đặc biệt là Form đăng ký hoặc Form phân công dự án phức tạp).
 - Tránh binding trực tiếp Entity vào Form để tránh lỗi bảo mật "Mass Assignment" và vòng lặp vô

tận (Infinite Recursion) khi làm việc với quan hệ 2 chiều JSON/Hibernate.

- **Security:** Tuyệt đối không hard-code mật khẩu hoặc thông tin nhạy cảm trong code.

6. YÊU CẦU KỸ THUẬT (TECHNOLOGY STACK)

- **Ngôn ngữ:** Java 17+
- **Framework:** Spring Boot 3.x
- **Web MVC:** Spring MVC, Thymeleaf.
- **Database:** MySQL hoặc SQL Server.
- **ORM:** Spring Data JPA, Hibernate.
- **Security:** Spring Security 6.x.
- **Build Tool:** Maven hoặc Gradle.
- **Frontend:** HTML5, CSS3, Bootstrap 5.

7. THANG ĐIỂM ĐÁNH GIÁ (MARKING SCALE)

Tiêu chí	Nội dung chi tiết	Điểm
Project Structure	Cấu trúc Project chuẩn, naming convention tốt, file config hợp lý.	10%
Security Implementation	Login/Logout, Mã hóa Password, Phân quyền URL, Ẩn hiện Menu theo Role.	25%
Database & Relationships	Thiết kế đúng ERD. Ánh xạ thành công các quan hệ One-to-Many và Many-to-Many trong Entity JPA.	15%
HR Module (1-N)	CRUD Department, Employee. Xử lý Dropdown phòng ban. Validate xóa phòng ban.	15%
Project Module (N-N)	CRUD Project. Chức năng Add Member to Project hoạt động đúng. Hiển thị danh sách thành viên.	20%
Validation & UI	Form validation chặt chẽ. Giao diện Bootstrap gọn gàng, bố cục hợp lý.	15%

TỔNG CỘNG		100%
-----------	--	------

8. HƯỚNG DẪN NỘP BÀI (SUBMISSION)

1. **Source Code:** Nén toàn bộ project (loại bỏ thư mục target, .mvn, node_modules nếu có) thành file .zip.
2. **Database Script:** Xuất file .sql chứa cấu trúc bảng và dữ liệu mẫu (Data seeding: ít nhất 2 Admin, 3 Dept, 5 Employee, 2 Project).
3. **Báo cáo (Optional):** File README.md hướng dẫn cách chạy project và danh sách tài khoản test (username/password).

Lưu ý: Gian lận (Copy code) sẽ bị 0 điểm.

-- THE END --