

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



Triển khai quản lý doanh nghiệp với Spring Boot

Sinh viên thực hiện:		
STT	Họ tên	MSSV
1	Nguyễn Minh Quân	19522082
2	Phạm Ngô Hoàng Sinh	19522129

TP. HỒ CHÍ MINH – 06/2023

1. GIỚI THIỆU

Đề tài triển khai quản lý doanh nghiệp với Spring Boot, nhằm giới thiệu việc sử dụng Spring Boot - một framework phát triển ứng dụng mạnh mẽ của java, để xây dựng một hệ thống quản lý doanh nghiệp linh hoạt và hiệu quả.

Đề tài tập trung vào sử dụng mô hình MVC (Model-View-Controller) để tách biệt các thành phần trong ứng dụng, bao gồm dữ liệu, giao diện người dùng và logic xử lý. Để đảm bảo bảo mật cho hệ thống, Spring Security được áp dụng, cung cấp các tính năng quan trọng như xác thực và ủy quyền. Đối với việc lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu quan hệ, đề tài sử dụng JPA (Java Persistence API) kết hợp với Hibernate, một triển khai phổ biến của JPA. MySQL được chọn làm hệ quản trị cơ sở dữ liệu cho dự án. Trong quá trình phát triển, Thymeleaf được sử dụng để phát triển giao diện người dùng web trong ứng dụng Spring. Java Servlet, công nghệ cung cấp API và công cụ cho phát triển ứng dụng web trên nền tảng Java, cũng được áp dụng để xây dựng các thành phần xử lý yêu cầu và phản hồi HTTP. Đặc biệt, đề tài sẽ sử dụng thư viện Lombok để giảm mã lặp và tăng hiệu suất phát triển bằng cách tự động tạo mã Java, bao gồm các phương thức getter, setter và constructor dựa trên các annotation.

Mục tiêu của đề tài "Triển khai quản lý doanh nghiệp với Spring Boot" là xây dựng một ứng dụng web server quản lý doanh nghiệp với 2 module chính là quản lý nhân viên và quản lý phương tiện, đáp ứng đầy đủ chức năng, sử dụng các công nghệ và thư viện mạnh mẽ của Spring Boot.

2. NỘI DUNG

2.1. Giới thiệu Spring Boot.

Spring Boot là một framework phát triển ứng dụng Java mạnh mẽ và linh hoạt. Nó được thiết kế nhằm giúp đơn giản hóa quá trình xây dựng ứng dụng, đồng thời cung cấp các tiện ích và cấu hình mặc định để giảm thiểu sự phức tạp và tăng hiệu suất phát triển.

Với Spring Boot, có thể nhanh chóng khởi tạo một ứng dụng Java mà không cần cấu hình phức tạp. Nó tự động cấu hình các thành phần cần thiết như Spring Framework, Spring Security, JPA và Hibernate, Thymeleaf, và nhiều công nghệ khác, cho phép bạn tập trung vào việc phát triển logic ứng dụng thay vì mất thời gian với các cấu hình ban đầu.

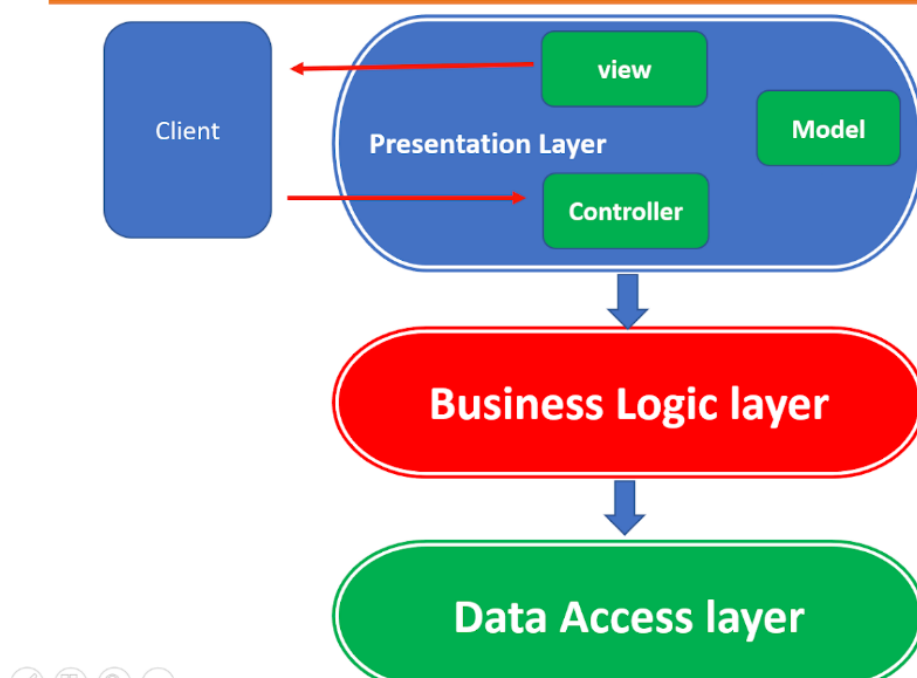
Spring Boot cũng hỗ trợ việc triển khai và quản lý ứng dụng dễ dàng. Nó tích hợp sẵn các công cụ giúp đóng gói ứng dụng thành file JAR hoặc WAR, và có khả năng tự động cấu hình và điều chỉnh ứng dụng trong môi trường khác nhau.

Tóm lại sử dụng Spring boot nhằm giảm thiểu sự phức tạp và tăng hiệu suất phát triển đồng thời tăng khả năng quản lý mã nguồn.

2.1.1 Kiến trúc mã nguồn.

Trong đề tài này, kiến trúc mã nguồn Spring boot được cấu tạo nên từ mô hình MVC trong việc xử lý yêu cầu của người dùng kết hợp mô hình three-layer để tách biệt logic nghiệp vụ và truy xuất dữ liệu.

Three-Tier architecture vs MVC pattern



Hình 1: sự kết hợp giữa Mô hình MVC và three-layer

- Persentation Layer: lớp này tương tác với người dùng thông qua View, Controller của mô hình MVC.
- Business Logic layer: lớp này chứa toàn bộ logic của chương trình, nó chính là Service trong mã nguồn, để định nghĩa service ta sử dụng annotation @Service.
- Data access layer: lớp này chính là Repository có nhiệm vụ tương tác trực tiếp với database và sau đó trả kết quả về cho tầng trên là Service, sử dụng annotation @Repository để định nghĩa.

2.1.2 Kiến trúc dữ liệu.

Để đảm bảo dữ liệu được, đồng thời thực hiện nguyên lý đơn nghiệm cho từng class. Có nghĩa là mỗi class chỉ thực hiện một nhiệm vụ, mục đích duy nhất.



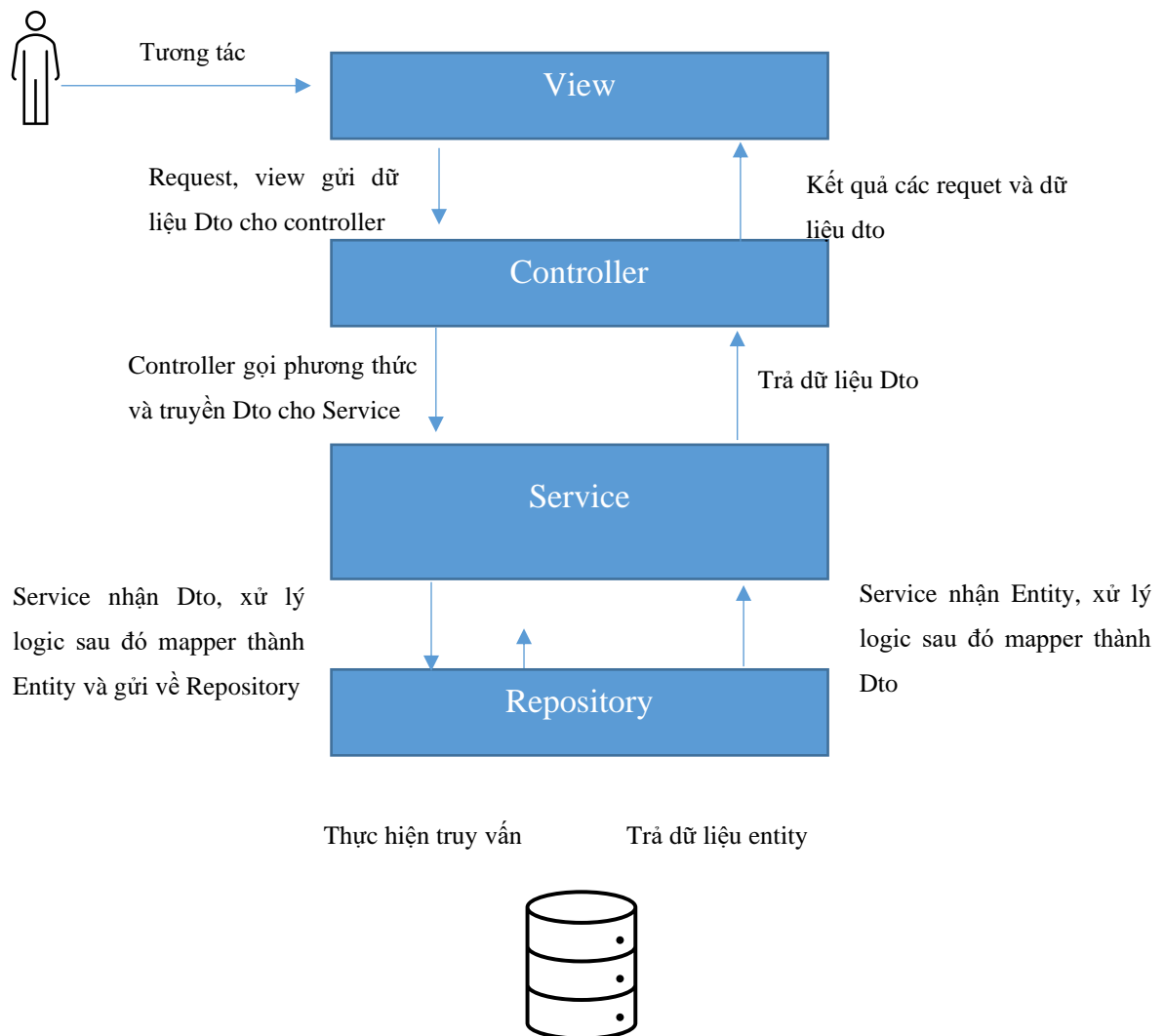
Hình 2: Kiến trúc dữ liệu

- Web layer: chỉ sử dụng dữ liệu public dạng Dto để nhận dữ liệu từ người dùng và truyền dữ liệu đến cho Service và ngược lại web layer nhận dữ liệu từ service ở dạng Dto sau đó hiển thị nên cho người dùng cũng là Dto.
- Service layer: sử dụng cả public lẫn private là cầu nối giữa Dto và Entity. Có thể nhận vào DTO từ web layer(controller), dữ liệu được xử lý cuối cùng được

Service trả về Web layer dưới dạng DTO. Đồng thời nhận Entity từ Repository sau đó mapper thành Dto để xử lý hoặc ngược lại sau đó trả về cho Repository.

- Repository layer: Có thể nhận entity từ Service bằng cách mapper, chỉ thao tác trên Entity, vì đó là đối tượng thích hợp, có thể mapping vào DB.

2.1.3 Tương tác giữa các thành phần.



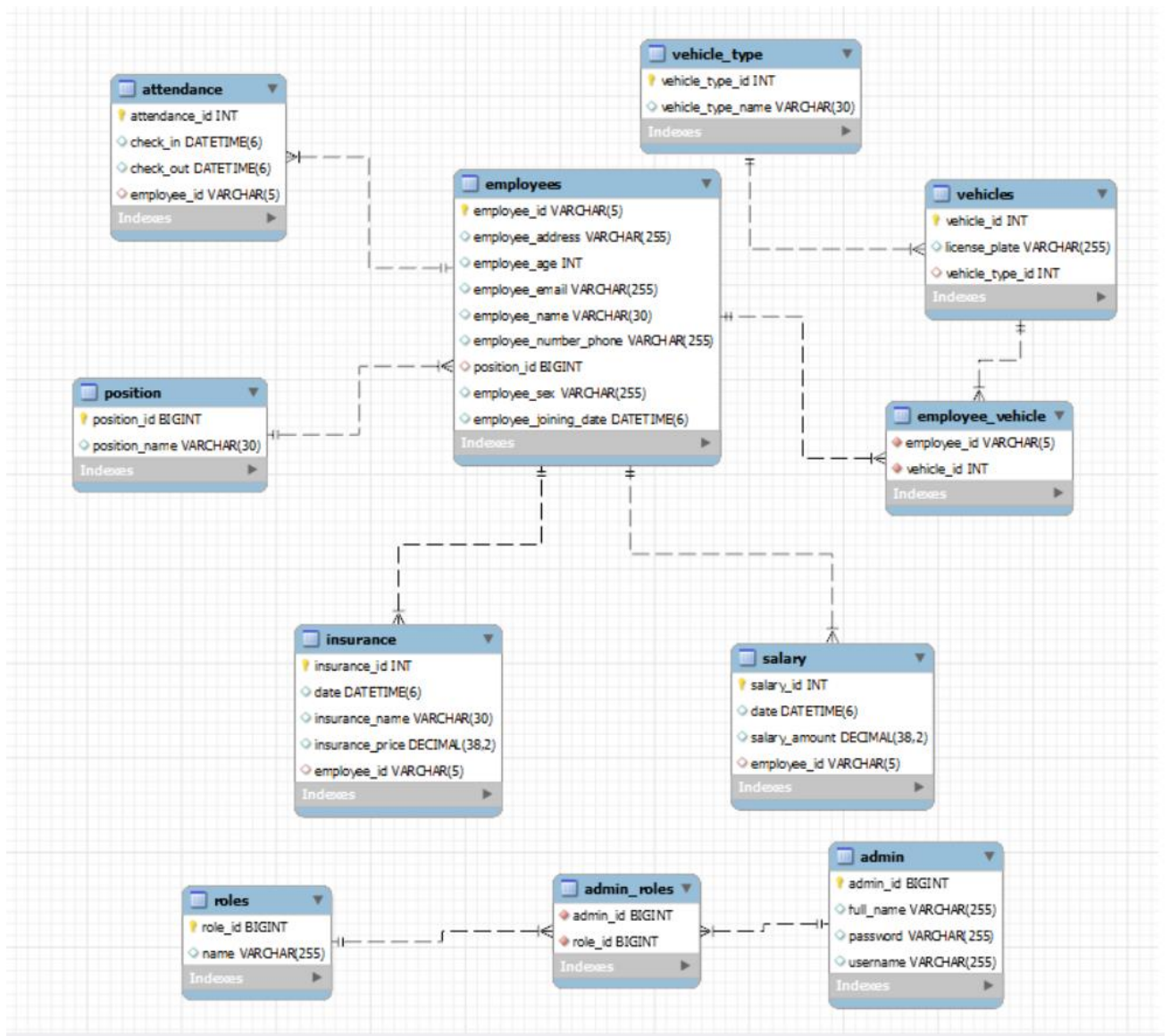
Hình 3: Luồng tương tác giữa các thành phần

2.2 Triển khai hệ thống quản lý doanh nghiệp

2.2.1 Thiết kế cơ sở dữ liệu

Cơ dữ liệu gồm 2 phần thành phần chính:

- Đăng nhập
- Quản lý doanh nghiệp



Hình 4: cơ sở dữ liệu của hệ thống

Hệ thống đăng nhập gồm 2 bảng chính và 1 bảng mối quan hệ:

- Admin:
 - **admin_id: mã chính**
 - full_name: họ và tên
 - password: mật khẩu
 - username: tài khoản
- Roles:
 - **Role_id: mã chính**
 - name: tên role
- Admin_Roles:
 - admin_id: mã admin
 - roles_id: mã role

Hệ thống quản lý gồm 6 bảng chính và 2 bảng phụ:

6 bảng chính:

- Employees
 - **employees_id: mã chính nhân viên**
 - employees_address: địa chỉ nhân viên
 - employees_age: tuổi của nhân viên
 - employees_email: địa chỉ email nhân viên
 - employees_name: Tên nhân viên
 - employees_number_phone: số điện thoại
 - employees_sex: giới tính
 - employees_start_date: ngày bắt đầu làm
 - position_id: mã vị trí
- Attendance
 - **attendance_id: mã chính điểm danh**
 - check_in: thời gian bắt đầu làm việc
 - check_out: thời gian kết thúc làm việc
 - employees_id: mã chính nhân viên

- Position
 - **position_id**: mã chính vị trí
 - position_name: tên vị trí
- Insurance
 - insurance_id: mã chính bảo hiểm
 - date: ngày phát lương
 - insurance_name: tên bảo hiểm
 - insurance_price: giá bảo hiểm
 - **employees_id**: mã chính nhân viên
- Salary
 - salary_id: mã chính lương
 - date: ngày đăng ký
 - salary_amount: mức lương
 - **employees_id**: mã chính nhân viên
- Employees_vehicles
 - vehicle_id: mã chính phương tiện di chuyển
 - **employees_id**: mã chính nhân viên

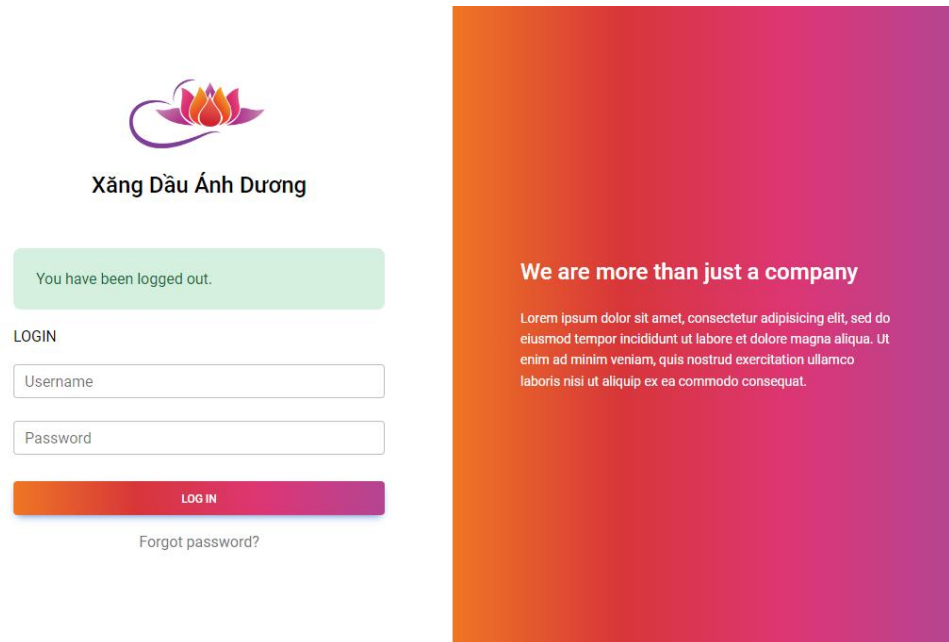
2 bảng phụ:

- Vehicles
 - vehicle_id: mã chính phương tiện di chuyển
 - **license_plate**: giấy phép lưu thông
 - vehicle_type_id: mã chính loại phương tiện
- Vehicles Type
 - vehicle_type_id: mã chính loại phương tiện
 - Vehicle_type_name: tên loại phương tiện

2.2.2 Chức năng của các module.

a. Đăng nhập.

Giao diện đăng nhập yêu cầu Admin đăng nhập trước khi vào các page, mật khẩu được mã.

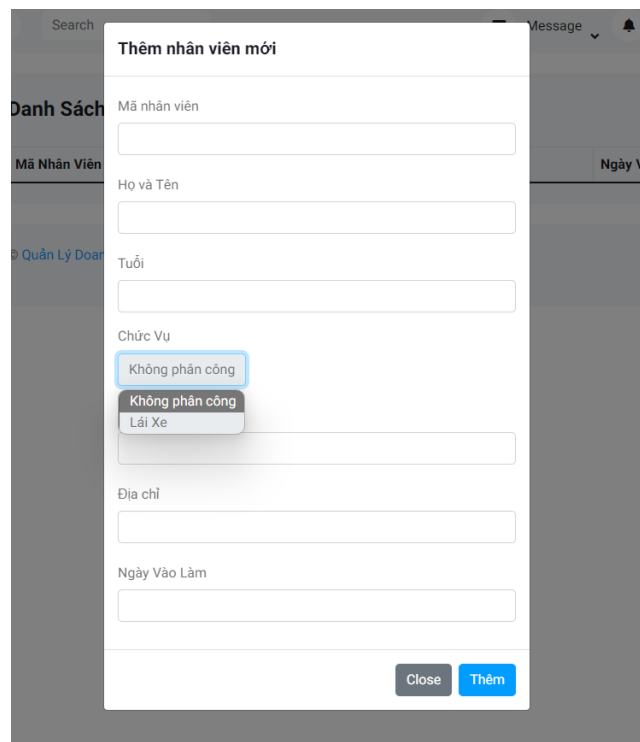


The image shows a login interface for 'Xăng Dầu Ánh Dương'. At the top is a logo of a stylized flower. Below it, a green message box says 'You have been logged out.' The 'LOGIN' section contains a 'Username' field, a 'Password' field, and a 'LOGIN' button. A link for 'Forgot password?' is below the button. To the right is a large orange-to-purple gradient box with the text 'We are more than just a company' and a paragraph of Lorem Ipsum text.

Hình 5: Chức năng đăng nhập

b. Danh sách nhân viên.

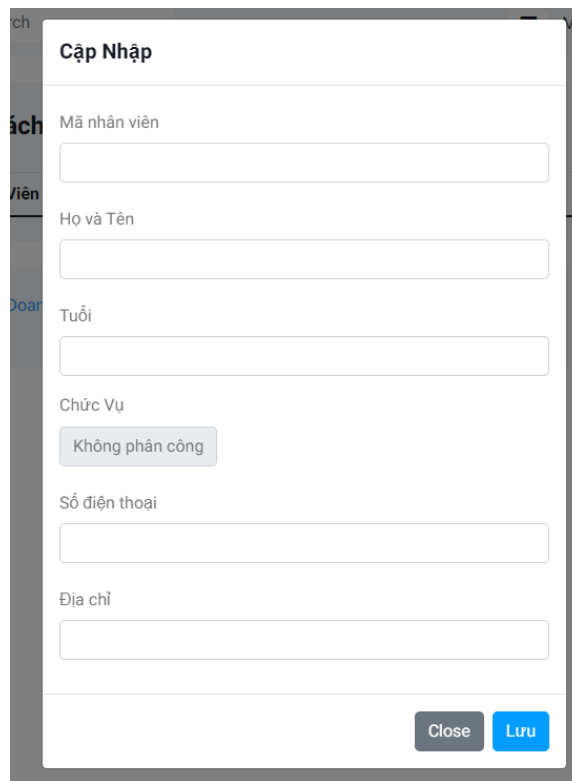
Chức năng thêm nhân viên



The image shows a mobile app interface for adding a new employee. A modal titled 'Thêm nhân viên mới' is open over a background showing a list of employees. The form fields are: 'Mã nhân viên', 'Họ và Tên', 'Tuổi', 'Chức Vụ' (with a dropdown menu showing 'Không phân công', 'Không phân công', and 'Lái Xe'), 'Địa chỉ', and 'Ngày Vào Làm'. At the bottom are 'Close' and 'Thêm' buttons.

Hình 6: Thêm nhân viên mới vào danh sách nhân viên

Chức năng sửa nhân viên



The screenshot shows a modal window titled "Cập Nhật" (Update). It contains several input fields for updating employee information: "Mã nhân viên" (Employee ID), "Họ và Tên" (Full Name), "Tuổi" (Age), "Số điện thoại" (Phone Number), and "Địa chỉ" (Address). Below the "Tuổi" field, there is a section for "Chức Vụ" (Job Title) with a button labeled "Không phân công" (No assignment). At the bottom right of the modal, there are two buttons: "Close" and "Lưu" (Save).

Hình 7: Sửa nhân viên

Xóa nhân viên theo mã Id

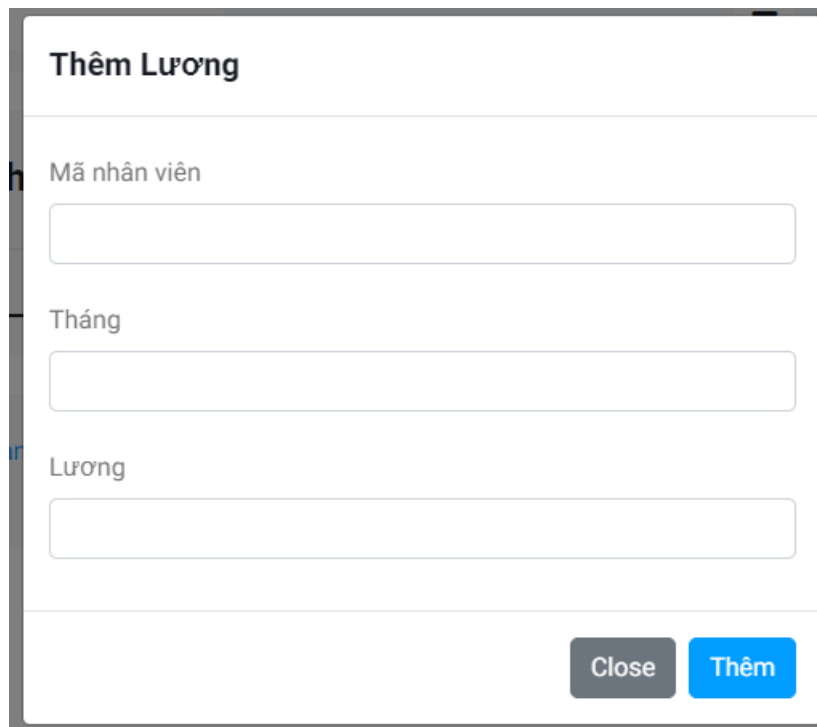


The screenshot shows a modal window titled "Xóa Nhân Viên" (Delete Employee). It contains a single input field labeled "Mã nhân viên" (Employee ID). At the bottom right of the modal, there are two buttons: "Close" and "Xóa" (Delete).

Hình 8: Xóa nhân viên

c. Quản lý lương

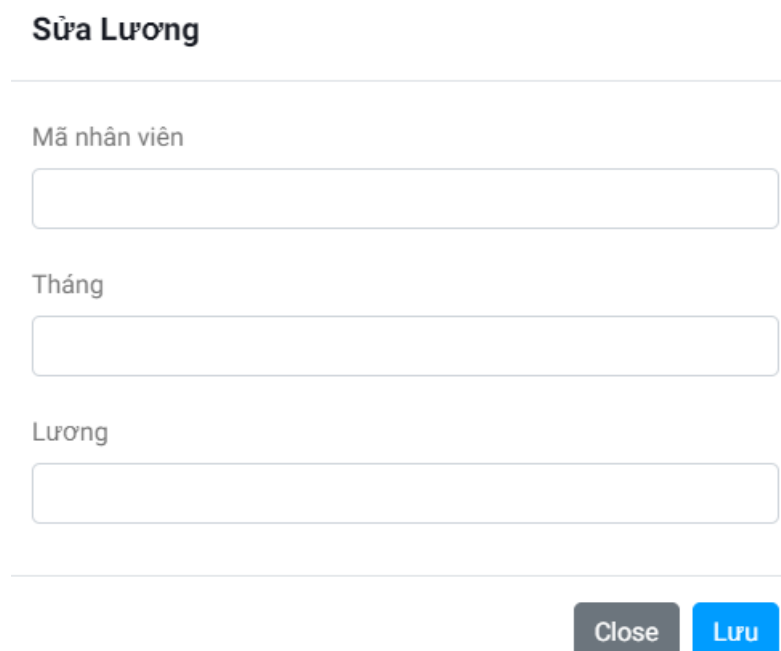
Thêm lương cho nhân viên



The form is titled "Thêm Lương" (Add Salary). It contains three input fields: "Mã nhân viên" (Employee ID), "Tháng" (Month), and "Lương" (Salary). At the bottom right, there are two buttons: "Close" and "Thêm" (Add).

Hình 9: thêm lương cho nhân viên

Sửa lương cho nhân viên



The form is titled "Sửa Lương" (Edit Salary). It contains three input fields: "Mã nhân viên" (Employee ID), "Tháng" (Month), and "Lương" (Salary). At the bottom right, there are two buttons: "Close" and "Lưu" (Save).

Hình 10: Sửa lương cho nhân viên

Xóa lương Theo Mã nhân viên và tháng

Xóa Lương

Mã nhân viên

Tháng

Close

Xóa

Hình 11: Xóa lương

d. Quản lý chức vụ

Thêm chức vụ

Thêm Chức Vụ

Tên Chức Vụ

Close

Thêm

Hình 12: Thêm chức vụ

Xóa chức vụ

Xóa Chức Vụ

Tên Chức Vụ

Close

Xóa

Hình 13: Xóa chức vụ

3. KẾT LUẬN

Trong quá trình thực hiện đồ án Spring Boot, nhóm đã sử dụng một loạt các công nghệ và kỹ thuật như Spring Security, Thymeleaf, JPA và MySQL. Đồ án được thiết kế theo mô hình MVC kết hợp với Three-layer.

Đã triển khai được các module Quản lý nhân viên, quản lý lương, quản lý chức vụ, song mặc dù chưa hoàn thành các module về bảo hiểm, phương tiện và chấm công, nhóm đã đạt được một số kết quả quan trọng trong quá trình triển khai đồ án. Tuy chưa hoàn thiện hoàn toàn, nhưng nhóm hy vọng đồ án này sẽ tiếp tục được phát triển và bổ sung các module còn thiếu để trở thành một hệ thống quản lý nhân sự hoàn chỉnh và hữu ích cho doanh nghiệp.

Tổng kết lại, nhóm đã đạt được một số chức năng quan trọng như đăng nhập và ba module chính gồm quản lý danh sách nhân viên, quản lý lương và quản lý chức vụ. Với Spring Security, nhóm đã xây dựng một hệ thống đăng nhập bảo mật, đảm bảo rằng chỉ người dùng có quyền truy cập mới có thể truy cập vào các tính năng quản lý. Qua việc sử dụng Thymeleaf, nhóm đã tạo giao diện người dùng thân thiện và tương tác, giúp người dùng dễ dàng sử dụng các chức năng và thực hiện các tác vụ quản lý nhân viên, lương và chức vụ. Nhóm đã sử dụng mô hình Three-layer để tổ chức mã nguồn một cách rõ ràng, phân tách logic kinh doanh, truy cập dữ liệu và giao diện người dùng.

TÀI LIỆU THAM KHẢO

- [1] "Cấu trúc dự án Spring Boot thế nào cho chuẩn" trên trang web ([Cấu trúc dự án Spring Boot thế nào cho chuẩn? \(viblo.asia\)](#)) do tonghoangvu: hướng dẫn cách tổ chức source code thế nào cho chuẩn Spring Boot. giúp bạn tìm hiểu về Spring Boot và cách sử dụng nó để xây dựng ứng dụng web. Nó bao gồm cả Spring Boot MVC và các chủ đề khác liên quan.
- [2] "Entity, domain model và DTO" trên trang web ([Entity, domain model và DTO - sao nhiều quá vậy? \(viblo.asia\)](#)) do tonghoangvu: Bài viết này cung cấp chi tiết lý thuyết, mối liên quan giữa Entity và Dto.
- [3] " Spring Boot Reference Documentation " ([Spring Boot Reference Documentation](#)) do nhiều tác giả đóng góp: Trang web này là tài liệu chuẩn của nhà phát hành, cung cấp mọi thứ liên quan đến spring boot

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
1	Nguyễn Minh Quân	Tìm hiểu Spring boot Tìm hiểu mô hình MVC Tìm hiểu Spring security Thymeleaf Triển khai các entity, repository, service cho Admin, Role, Employee, salary, Attendance. Các chức năng thêm sửa xóa, chi tiết. Viết báo cáo
2	Phạm Ngô Hoàng Sinh	Tìm hiểu Spring boot, Mô hình three-layer, thymeleaf Triển khai các entity, repository, service cho Position, Insurance, Vehicle. Các chức năng thêm sửa xóa. Viết báo cáo Làm powerpoint
3		