**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

=========\*\*\*=========

****

**BÁO CÁO THÍ NGHIỆM/THỰC NGHIỆM LẬP TRÌNH JAVA**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ NHÂN VIÊN**

**BẰNG JAVA**

| GVHD: | TS. Hà Mạnh Đào |
| --- | --- |
| Sinh viên: | Phạm Văn Thành – 2021608254  Lại Hải Nam - 2023601539  Nguyễn Ngọc Hoàng Anh - 2023602727 |
| Nhóm: | 14 |
| Lớp: | 20241IT6019004 |
| Khóa: | K16 |
|  |  |

Hà Nội, Năm 2024

# **CHƯƠNG 1: MỞ ĐẦU**

## **1.1. Giới thiệu Java**

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động. Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak. Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem. Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA).

Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

## **1.2. Giới thiệu Java Swing**

Java Swing là cách gọi rút gọn khi người ta nhắc đến Swing của Java Foundation (JFC). Nó là bộ công cụ GUI mà Sun Microsystems phát triển để xây dựng các ứng dụng tối ưu dùng cho window (bao gồm các thành phần như nút, thanh cuộn,…).

Swing được xây dựng trên AWT API và hoàn toàn được viết bằng Java. Tuy nhiên, nó lại khác với AWT ở chỗ bộ công cụ này thuộc loại nền tảng độc lập, bao gồm các thành phần nhẹ và phức tạp hơn AWT.

Các gói javax.swing bao gồm các lớp cho Java Swing API như JMenu, JButton, JTextField, JRadioButton, JColorChooser,…

Việc xây dựng ứng dụng sẽ trở nên dễ dàng hơn với Java Swing vì chúng ta có các bộ công cụ GUI giúp đỡ công việc.

## **1.3. Tổng quan về bài toán quản lý nhân viên**

Bài toán quản lý nhân viên là quá trình quản lý và tối ưu hóa tất cả các hoạt động liên quan đến nhân sự trong một tổ chức. Nó bao gồm một loạt các hoạt động từ việc tuyển dụng, phát triển, quản lý hiệu suất, và duy trì mối quan hệ lao động.

Quản lý nhân sự là quá trình điều hành và tổ chức nguồn lực con người trong một tổ chức hoặc doanh nghiệp để đạt được các mục tiêu kinh doanh và mục tiêu chiến lược. Nó bao gồm việc tuyển dụng, đào tạo, phát triển, đánh giá hiệu suất, và duy trì mối quan hệ lao động trong tổ chức.

Quản lý nhân sự đảm bảo rằng tổ chức có đủ nguồn lực con người với kỹ năng và năng lực phù hợp để thực hiện công việc của mình một cách hiệu quả. Nó cũng liên quan đến việc tạo ra một môi trường làm việc tích cực và động viên nhân viên để họ phát triển và thăng tiến trong sự nghiệp của mình.Điều này có thể bao gồm việc thiết lập chính sách và quy trình liên quan đến tiền lương, phúc lợi, đào tạo, và phát triển nhân viên. Ngoài ra, quản lý nhân sự cũng phải giải quyết các vấn đề và xung đột liên quan đến nhân viên, đảm bảo tính công bằng và tuân thủ các quy định pháp luật lao động.

Tóm lại, quản lý nhân sự là một phần quan trọng trong việc xây dựng và duy trì một tổ chức hoạt động hiệu quả và mang lại thành công bền vững thông qua việc quản lý và phát triển nguồn lực con người. Các công nghệ thông tin và ứng dụng phần mềm hiện đại, như ứng dụng Swing trong Java, có thể được áp dụng để xây dựng các giải pháp tối ưu cho bài toán này.

## **1.4. Tổng quan về mô hình MVC**

Mô hình MVC là viết tắt của Model-View-Controller, là một kiến trúc phần mềm phổ biến được sử dụng để tổ chức mã nguồn trong các ứng dụng phần mềm. Mô hình này giúp tách biệt các thành phần của hệ thống để dễ quản lý và bảo trì.

1. Model

- Đại diện cho dữ liệu và logic xử lý dữ liệu.

- Cập nhật trạng thái của ứng dụng khi dữ liệu thay đổi.

- Không có thông tin về cách dữ liệu được hiển thị hoặc tương tác với người dùng.

b. View

- Hiển thị thông tin cho người dùng.

- Được cập nhật khi trạng thái của Model thay đổi.

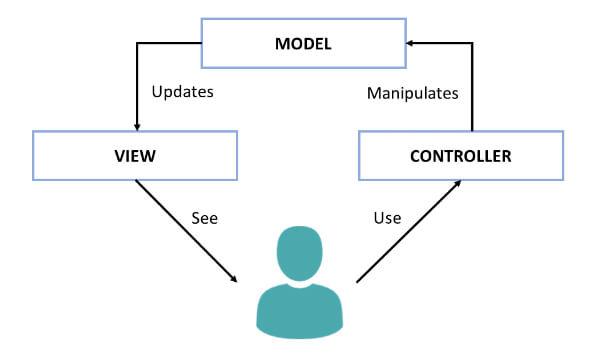
- Không chứa logic xử lý dữ liệu.

c. Controller

- Nhận đầu vào từ người dùng thông qua giao diện.

- Chịu trách nhiệm xử lý sự kiện và tương tác với Model.

- Cập nhật trạng thái của Model và thông báo cho View về các thay đổi.



Hình 1: Mô hình MVC

# **CHƯƠNG 2: KẾT QUẢ NGHIÊN CỨU**

## **2.2. Khảo sát hệ thống**

### **2.2.1. Khảo sát sơ bộ**

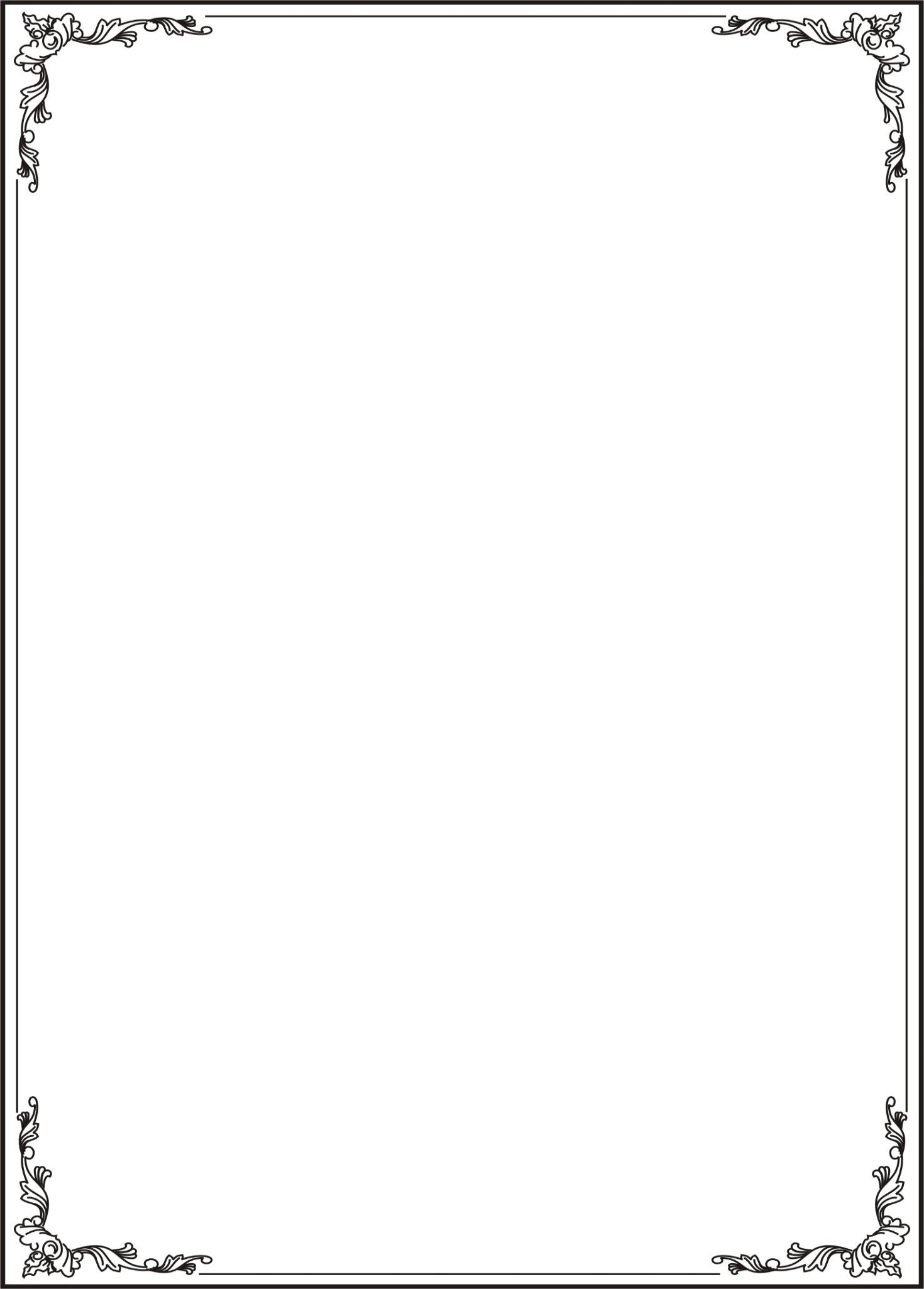
1. *Mục tiêu*

Điều tra, khảo sát về hệ thống quản lý nhân sự. Tìm hiểu cách quản lý thông tin thông tin phòng ban, lịch làm việc…

*b) Phương pháp*

**-** Phỏng vấn.

|  |  |
| --- | --- |
| Kế hoạch phỏng vấn | |
| Người phỏng vấn: Phạm Văn Thành | Người được hỏi: Nguyễn Văn An |
| Thời gian hẹn: 8h45  Thời gian bắt đầu: 9h15  Thời gian kết thúc: 10h30 | Địa chỉ: Tòa 9 tòa nhà FPT Software, Hà Nội. |
| Yêu cầu:  Vị trí: Người quản lý  Trình độ: Hiểu rõ về quản lý phòng ban, thông tin nhân viên, thời gian làm việc. | Đối tượng:  Vị trí: Quản lý phòng nhân sự  Cần thu thập dữ liệu: Cách thức quản lý phòng nhân sự và cung cấp phòng ban cho nhân viên. |

Bảng 1: Phiếu phỏng vấn 

*c) Kết quả sơ bộ.*

* Điểm mạnh và điểm yếu của hệ thống đang vận hành
  + Điểm mạnh: Hệ thống hoạt động ổn định.
  + Điểm yếu: Hệ thống phải dùng nhiều những giấy tờ thủ tục rườm rà, mất rất nhiều thời gian và công sức cho báo cáo và lên danh sách , đôi lúc còn bị sai sót, tính bảo mật chưa cao. Tài liệu đặc tả yêu cầu người dùng.

### **2.2.2. Khảo sát chi tiết**

Để thực hiện khảo sát chi tiết cho việc thiết kế hệ thống quản lý nhân viên bằng Java, nhóm thực hiện các bước sau:

1. Xác định đối tượng khảo sát:

* Người dùng chính của hệ thống sẽ là quản lý nhân sự, các bên liên quan khác như trưởng phòng, giám đốc.
* Cần xác định số lượng đại diện cần được phỏng vấn/khảo sát từ mỗi nhóm người dùng này.

1. Thu thập thông tin:

Xây dựng bộ câu hỏi khảo sát chi tiết bao gồm:

* Các chức năng hiện tại và mong muốn của hệ thống quản lý nhân viên (quản lý thông tin nhân viên, phòng ban, quản lí lương, thông tin nhân viên, đánh giá)
* Các vấn đề, thách thức trong quản lý nhân viên hiện tại
* Lựa chọn phương pháp thu thập thông tin như phỏng vấn trực tiếp, khảo sát online, quan sát quy trình, etc.

1. Phân tích và tổng hợp thông tin:

* Tổng hợp, phân loại và phân tích dữ liệu thu thập được.
* Xác định các chức năng chính, quy trình quan trọng và các yêu cầu ưu tiên của người dùng.
* Xác định các yêu cầu chức năng và phi chức năng chính cho hệ thống mới, bao gồm:
* Quản lý thông tin nhân viên
* Quản lý thông tin phòng ban
* Quản lý thông tin chức vụ
* Quản lý tài khoản
* Quản lý lương
* Yêu cầu về thống kê, báo cáo.

1. Báo cáo và trình bày kết quả:

* Đảm bảo các yêu cầu được ghi lại đầy đủ và rõ ràng để làm cơ sở cho giai đoạn thiết kế và phát triển.

## **2.3. Phân tích hệ thống**

### **2.3.1. Mô hình hóa chức năng của hệ thống**

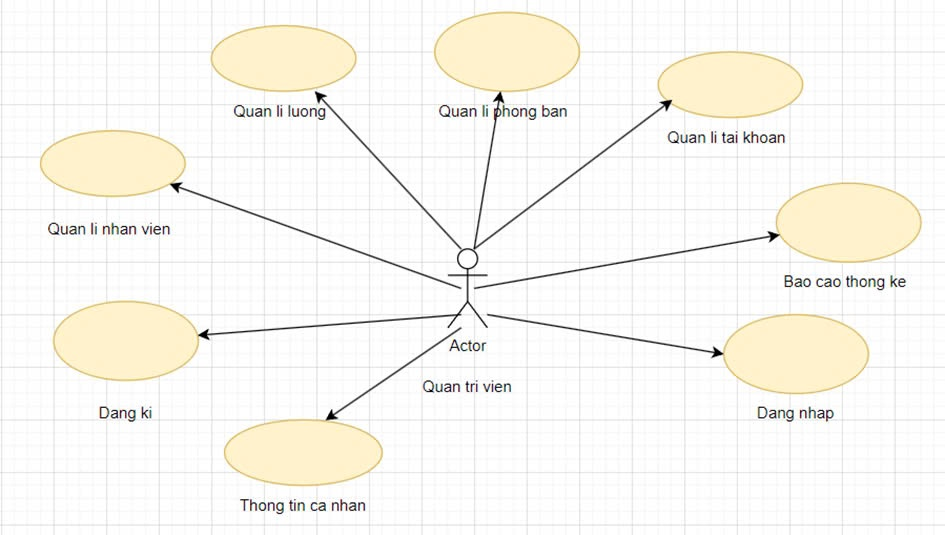
**2.3.1.1. Biểu đồ use case**

|  |  |
| --- | --- |
| Tên tác nhân | Mô tả ngắn gọn tác nhân |
| Quản trị viên | Người quản trị hệ thống , nhân viên, phòng ban, chức vụ , cấp quyền tài khoản |

Bảng 2: Tác nhân use-case

|  |  |  |
| --- | --- | --- |
| ID | Tên use case | Mô tả ngắn gọn use case |
| UC\_01 | Đăng ký | Cho phép quản trị viên đăng ký tài khoản |
| UC\_02 | Đăng nhập | Cho phép quản trị viên đăng nhập khi đã có tài khoản |
| UC\_03 | Quản lý tài khoản | Quản lí thông tin tài khoản thực hiện các hành động thêm, sửa, xóa, tìm kiếm |
| UC\_04 | Quản lí nhân viên | Quản lí số lượng, thông tin nhân viên, thực hiện các hành động thêm, sửa, xóa, tìm kiếm |
| UC\_05 | Quản lí phòng ban | Quản lí số lượng phòng ban, thực hiện các hành động thêm, sửa, xóa, tìm kiếm |
| UC\_06 | Quản lý chức vụ | Quản lí các chức vụ, thực hiện các hành động thêm, sửa, xóa, tìm kiếm |
| UC\_07 | Quản lí lương, thông tin nhân viên | Quản lí lương của nhân viên, thông tin chi tiết của nhân viên. Sửa hệ số lương của nhân viên theo lương |
| UC\_08 | Thống kê | Thống kê tổng số nhân viên, số lượng nhân viên theo chức vụ, phòng ban lương theo từng chức vụ, phòng ban |

Bảng 3: Danh sách usecase



Hình 2: Biểu đồ use case tổng quát

A diagram of a network

Description automatically generated

Hình 3: Biểu đồ use case chính

A diagram of a diagram

Description automatically generated

Hình 4: Biểu đồ phân rã use case chính

**2.3.1.2. Mô tả chi tiết use case**

**2.3.1.2.1. Mô tả use case “Đăng nhập”**

1. Tên Use case: Đăng nhập

2. Mô tả vắn tắt: Use case này cho phép người quản trị đăng nhập vào hệ thống

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên nhập tài khoản vào các trường dữ liệu và kích vào nút “Login” trên giao diện đăng nhập.
2. Hệ thống sẽ kiểm tra tài khoản nhập vào với danh sách tài khoản trong cơ sở dữ liệu và đối chiếu.
3. Hệ thống kiểm tra quyền đăng nhập của tài khoản, nếu có hệ thống sẽ hiển thị giao diện quản trị.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bước 1 trong luồng cơ bản, nếu người dùng không nhập thông tin hoặc nhập thông tin tài khoản, mật khẩu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
2. Tại bước 2 nếu không có dữ liệu trong file dữ liệu hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và Use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Quản trị viên phải có tài khoản đã được đăng ký.

6. Hậu điều kiện: Khi use case kết thúc thành công, hiện thông báo đăng nhập thành công và hiển thị giao diện cho quản trị viên

7. Điểm mở rộng: Không có.

**2.3.1.2.2. Mô tả use case “Đăng ký”**

1. Tên Use case: Đăng ký

2. Mô tả vắn tắt: Use case này cho phép quản trị viên đăng ký tài khoản hệ thống.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi quản trị viên nhập các thông tin và kích nút “Sign Up” trên giao diện Đăng ký.
2. Hệ thống sẽ lưu thông tin tài khoản vào bảng Account trong cơ sở dữ liệu

Use case kết thúc

3.2. Các luồng rẽ nhánh

1. Tại bước 1 luồng cơ bản khi người dùng không nhập đầy đủ các trường hệ thống thông báo lỗi và use case kết thúc.
2. Tại bước 2 luồng cơ bản, hệ thống sẽ kiểm tra tên tài khoản đã tồn tại trong cơ sở dữ liệu chưa, nếu đã tồn tại thông báo lỗi sẽ xuất hiện, nếu chưa thông tin tài khoản đăng ký sẽ lưu vào cơ sở dữ liệu.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Tài khoản muốn đăng ký chưa được sử dụng trước đó.

6. Hậu điều kiện: Sau khi đăng ký thành công tài khoản được lưu vào cơ sở dữ liệu.

7. Điểm mở rộng: Không có

**2.3.1.2.3 Mô tả use case “Quản lý Tài Khoản”**

1. Tên Use case: Quản lý tài khoản.

2. Mô tả vắn tắt: Use case này cho phép người quản trị thực hiện các chức năng thêm, sửa, xóa, tìm kiếm các tài khoản và xuất danh sách các tài khoản trong hệ thống.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi người quản trị click vào “Tài khoản” trên màn hình .Hệ thống hiển thị cửa sổ quản lý chức vụ đồng thời lấy dữ liệu cơ sở dữ liệu và hiển thị danh sách toàn bộ tài khoản lên bảng danh sách tài khoản trên màn hình.
2. Thêm tài khoản mới:
   1. Người quản trị kích vào nút “Thêm” trên giao diện.
   2. Người quản trị nhập các thông tin tài khoản mới bao gồm: Tên tài khoản, Mật khẩu,
   3. Người quản trị click vào nút “Xác nhận”. Hệ thống sẽ tạo một bản ghi mới gồm các thông tin Tên tài khoản , Mật khẩu và ghi thông tin vào bảng Account trong cơ sở dữ liệu
3. Sửa thông tin tài khoản:
   1. Người quản trị kích chọn nút “Sửa” trên giao diện
   2. Người quản trị nhập tên tài khoản cần sửa và mật khẩu mới.
   3. Sau đó kích nút “Xác nhận”. Hệ thống sẽ tự động sửa thông tin của tài khoản trên bảng danh sách tài khoản trên màn hình. Hệ thống sẽ cập nhật dữ liệu vào bảng Account trong cơ sở dữ liệu
4. Xóa tài khoản:
   1. Người quản trị kích chọn nút “Xóa” trên giao diện.
   2. Người quản nhập tên tài khoản cần xóa.
   3. Sau đó kích chọn nút “Xác nhận”, hệ thống sẽ xóa bản ghi đó khỏi danh sách tài khoản trên giao diện, hệ thống sẽ cập nhật lại dữ liệu trong bảng Account trong cơ sở dữ liệu.
5. Tìm kiếm tài khoản:
6. Người dùng nhập tên tài khoản cần tìm kiếm vào ô tìm kiếm trong giao diện quản lý tài khoản, sau đó kích vào nút “Tìm kiếm” bên cạnh ô tìm kiếm.
7. Hệ thống sẽ lấy thông tin tài khoản từ bảng Account trong cơ sở dữ liệu bao gồm: Tên tài khoản, mật khẩu và hiển thị lên bảng danh sách tài khoản.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bước 2b, 3b, 4b, 5a luồng cơ bản khi người quản trị không nhập đầy đủ thông tin thì hệ thống thông báo lỗi và use case kết thúc.
2. Tại bước 2c luồng cơ bản khi người dùng nhập tên tài khoản trùng với tên tài khoản đã lưu trong cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi “Tài khoản đã tồn tại”.
3. Tại bước 5b trong luồng cơ bản nếu người quản trị nhập tên tài khoản không tồn tại trong cơ cở dữ liệu thông báo lỗi “Tài khoản không tồn tại” xuất hiện.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với file dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người quản trị có tài khoản đã đăng ký hoặc người dùng là nhân viên được cấp quyền quản trị.

6. Hậu điều kiện: Khi use case kết thúc thành công, hiện thông báo cập nhật thành công.

7. Điểm mở rộng: Không có.

**2.3.1.2.4. Mô tả use case “Quản lý nhân viên”**

1. Tên Use case: Quản lý nhân viên.

2. Mô tả vắn tắt: Use case này cho phép người quản trị thực hiện các chức năng thêm, sửa, xóa, tìm kiếm nhân viên và xuất danh sách nhân viên trong hệ thống.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi người quản trị click vào “Nhân viên” trên thanh menu. Hệ thống hiển thị cửa sổ quản lý nhân viên đồng thời lấy dữ liệu từ bảng Employee trong cơ sở dữ liệu và hiển thị danh sách toàn bộ nhân viên lên bảng danh sách nhân viên trên màn hình.
2. Thêm mới nhân viên:
3. Người quản trị kích vào nút “Thêm” trên giao diện.
4. Người quản trị nhập các thông tin cho nhân viên mới bao gồm: ID nhân viên, họ tên, ngày sinh, quê quán, giới tính, số điện thoại ,ngày làm việc, hệ số lương, id phòng ban, id chức vụ vào các trường điền thông tin nhân viên.
5. Người quản trị click vào nút “Xác nhận”. Hệ thống sẽ tạo một bản ghi mới gồm các thông tin trên và hiển thị lên bảng danh sách nhân viên, hệ thống sẽ lưu bản ghi mới vào bảng Employee trong cơ sở dữ liệu
6. Sửa thông tin nhân viên:
7. Người quản trị kích vào nút “Sửa” trên giao diện.
8. Người quản trị kích nhập ID nhân viên cần sửa và nhập thông tin mới cho nhân viên.
9. Người quản kích nút “Xác nhận”. Hệ thống sẽ tự động sửa thông tin của nhân viên trên bảng danh sách nhân viên trên màn hình và cập nhật dữ liệu vào bảng Employee trong cơ sở dữ liệu.
10. Xóa nhân viên:
11. Người quản trị kích vào nút “Xóa” trên giao diện.
12. Người quản trị nhập ID nhân viên cần xóa.
13. Người quản trị kích nút “Xác nhận”. Hệ thống sẽ xóa bản ghi đó khỏi danh sách nhân viên trên giao diện và cập nhật lại dữ liệu trong bảng Employee trong cơ sở dữ liệu.
14. Tìm kiếm nhân viên:
15. Người quản trị nhập ID nhân viên cần tìm kiếm vào ô tìm kiếm trên giao diện, sau đó kích vào nút “Tìm kiếm” bên cạnh ô tìm kiếm.
16. Hệ thống sẽ lấy thông tin nhân viên từ trong bảng Employee trong cơ sở dữ liệu :bao gồm: ID nhân viên, họ tên, ngày sinh, quê quán, giới tính, số điện thoại, ngày làm việc, hệ số lương, id phòng ban, id chức vụ và hiển thị lên bảng danh sách nhân viên.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bước 2b, 3b, 4b, 5a luồng cơ bản khi người quản trị không nhập đầy đủ thông tin thì hệ thống thông báo lỗi và use case kết thúc.
2. Tại bước 2c trong luồng cơ bản nếu người quản trị nhập ID nhân viên trùng với ID đã lưu trong cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi “ID trùng”.
3. Tại bước 3c trong luồng cơ bản nếu người quản trị nhập vào id chức vụ/ id phòng ban không tồn tại trong bảng Department và Position trong cơ sở dữ liệu, hệ thống sẽ thông báo lỗi “Không tồn tại id chức vụ đã nhập”/ “Không tồn tại id phòng ban đã nhập” xuất hiện.
4. Tại bước 5b trong luồng cơ bản nếu ID nhân viên được nhập không tồn tại trong cơ sở dữ liệu thông báo lỗi “ID không tồn tại” xuất hiện.
5. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với file dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người quản trị có tài khoản đã đăng ký.

6. Hậu điều kiện: Khi use case kết thúc thành công, hiện thông báo cập nhật thành công.

7. Điểm mở rộng: Không có

**2.3.1.2.5. Mô tả use case “Quản lý phòng ban”**

1. Tên Use case: Quản lý phòng ban.

2. Mô tả vắn tắt: Use case này cho phép người quản trị thực hiện các chức năng thêm, sửa, xóa, tìm kiếm phòng ban và xuất danh sách phòng ban trong hệ thống.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi người quản trị click vào “Phòng ban” trên menu.Hệ thống hiển thị cửa sổ quản lý phòng ban đồng thời lấy dữ liệu từ bảng Department trong cơ sở dữ liệu và hiển thị danh sách toàn bộ phòng lên bảng danh sách phòng ban trên màn hình.
2. Thêm phòng ban mới:
3. Người quản trị kích vào nút “Thêm” trên giao diện.
4. Người quản trị nhập các thông tin cho phòng mới bao gồm: Mã phòng, tên phòng, địa chỉ, số điện thoại vào các trường điền thông tin.
5. Sau đó click vào nút “Xác nhận”. Hệ thống sẽ tạo một bản ghi mới gồm các thông tin mã phòng, tên phòng, địa chỉ, số điện thoại và hiển thị lên bảng danh sách phòng ban, hệ thống sẽ lưu bản ghi mới vào trong bảng Department trong cơ sở dữ liệu.
6. Sửa thông tin phòng ban:
   * 1. Người quản trị kích vào nút “Sửa” trên giao diện.
     2. Người quản trị nhập ID phòng ban cần sửa và các thông tin phòng ban mới.
     3. Người quản trị kích nút “Xác nhận”. Hệ thống sẽ tự động sửa thông tin của phòng ban trên bảng danh sách phòng ban trên màn hình, hệ thống sẽ cập nhật thông tin vào bảng Department cơ sở dữ liệu.
7. Xóa phòng ban:
8. Người quản trị kích vào nút “Xóa” trên giao diện.
9. Người quản trị nhập ID phòng ban cần xóa.
10. Người quản trị kích nút “Xác nhận”. Hệ thống sẽ tự động sửa thông tin của phòng ban trên bảng danh sách phòng ban trên màn hình, hệ thống sẽ cập nhật lại dữ liệu bảng Department trong cơ sở dữ liệu.
11. Tìm kiếm phòng ban:
12. Người dùng nhập tên phòng ban cần tìm kiếm vào ô tìm kiếm trong giao diện quản lý phòng ban, sau đó kích vào nút “Tìm kiếm” bên cạnh ô tìm kiếm.
13. Hệ thống sẽ lấy thông tin phòng ban từ bảng Department trong cơ sở dữ liệu bao gồm: mã phòng, tên phòng, địa chỉ, số điện thoại và hiển thị lên bảng danh sách phòng ban.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bước 2b, 3b, 4b, 5a luồng cơ bản khi người quản trị không nhập đầy đủ thông tin thì hệ thống thông báo lỗi và use case kết thúc.
2. Tại bước 2c luồng cơ bản khi người dùng nhập ID phòng ban trùng với ID đã lưu trong cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi “ID trùng”.
3. Tại bước 5a trong luồng cơ bản nếu người quản trị nhập ID phòng ban không tồn tại trong ô tìm kiếm thông báo lỗi “ID không tồn tại” xuất hiện.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người quản trị có tài khoản đã đăng ký.

6. Hậu điều kiện: Khi use case kết thúc thành công, hiện thông báo cập nhật thành công.

7. Điểm mở rộng: Không có.

**2.3.1.2.6. Mô tả use case “Quản lý chức vụ”**

1. Tên Use case: Quản lý chức vụ.

2. Mô tả vắn tắt: Use case này cho phép người quản trị thực hiện các chức năng thêm, sửa, xóa, tìm kiếm các chức vụ và xuất danh sách chức vụ trong hệ thống.

3. Luồng các sự kiện

3.1. Luồng cơ bản

1. Use case này bắt đầu khi người quản trị click vào “ Chức vụ” trên màn hình “Quản trị”. Hệ thống hiển thị cửa sổ quản lý chức vụ đồng thời lấy dữ liệu từ bảng Position trong cơ sở dữ liệu và hiển thị danh sách toàn bộ chức vụ lên bảng danh sách chức vụ trên màn hình.
2. Thêm chức vụ mới:
   1. Người quản trị kích vào nút “Thêm” trên giao diện.
   2. Người quản trị nhập các thông tin cho chức vụ mới bao gồm: Mã chức vụ, tên chức vụ, lương cơ bản vào các trường điền thông tin chức vụ.
   3. Sau đó click vào nút “Xác nhận”. Hệ thống sẽ tạo một bản ghi mới gồm các thông tin mã chức vụ, tên chức vụ, lương cơ bản và hiển thị lên bảng danh sách chức vụ, hệ thống sẽ lưu bản ghi mới vào bảng Position trong cơ sở dữ liệu.
3. Sửa thông tin chức vụ:
   1. Người quản trị kích vào nút “Sửa” trên giao diện.\
   2. Người quản trị nhập ID chức vụ cần sửa và nhập các thông tin mới cho chức vụ.
   3. Người quản trị kích nút “Xác nhận”. Hệ thống sẽ tự động sửa thông tin của chức vụ trên bảng danh sách chức vụ trên màn hình, hệ thống sẽ cập nhật dữ liệu vào bảng Position trong cơ sở dữ liệu.
4. Xóa chức vụ:
   1. Người quản trị kích vào nút “Xóa” trên giao diện.
   2. Người quản nhập ID chức vụ cần xóa.
   3. Người quản trị kích nút “Xác nhận”. Hệ thống sẽ xóa bản ghi đó khỏi danh sách chức vụ trên giao diện, hệ thống sẽ cập nhật lại dữ liệu vào bảng Position trong cơ sở dữ liệu.
5. Tìm kiếm chức vụ:
   1. Người dùng nhập ID chức vụ cần tìm kiếm vào ô tìm kiếm trong trong giao diện quản lý chức vụ, sau đó kích vào nút tìm kiếm bên cạnh ô tìm kiếm.
   2. Hệ thống sẽ lấy thông tin chức vụ từ bảng Position trong cơ sở dữ liệu bao gồm: mã chức vụ, tên chức vụ, lương cơ bản và hiển thị lên bảng danh sách chức vụ.

Use case kết thúc.

3.2. Các luồng rẽ nhánh

1. Tại bước 2b, 3b, 4b, 5a luồng cơ bản khi người quản trị không nhập đầy đủ thông tin thì hệ thống thông báo lỗi và use case kết thúc.
2. Tại bước 2c luồng cơ bản khi người dùng nhập ID chức vụ trùng với ID đã lưu trong cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi “ID trùng”.
3. Tại bước 5a luồng cơ bản nếu người quản trị nhập ID chức vụ không tồn tại trong bảng Position trong cơ sở dữ liệu, hệ thống sẽ thông báo lỗi “ID không tồn tại” xuất hiện.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người quản trị có tài khoản đã đăng ký.

6. Hậu điều kiện: Khi use case kết thúc thành công, hiện thông báo cập nhật thành công.

7. Điểm mở rộng: Không có.

**2.3.1.2.7. Mô tả use case “Quản lý lương”**

1. Tên Use case: Quản lý lương

2. Mô tả vắn tắt: Use case này thực hiện quá trình thống kê danh sách các thông tin chi tiết nhân viên và lương của từng nhân viên

3.1. Luồng cơ bản

1. Use case bắt đầu khi người quản trị kích vào nút “ Lương“ trên menu quản trị. Hệ thống hiển thị giao diện và các tùy chọn chức năng.

3.2. Các luồng rẽ nhánh

* 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc..

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người quản trị đã đăng nhập vào hệ thống và có quyền truy cập vào chức năng.

6. Hậu điều kiện: Không có.

7. Điểm mở rộng: Không có.

**2.3.1.2.8. Mô tả use case “Thống kê”**

1. Tên Use case: Báo cáo thống kê

2. Mô tả vắn tắt: Use case này thực hiện quá trình thống kê tổng số lượng nhân viên, số lượng nhân viên theo chức vụ và tổng lương của nhân viên các tất cả phòng ban và chức vụ

3.1. Luồng cơ bản

1. Use case bắt đầu khi người quản trị kích vào nút “ Thống kê “ trên menu quản trị. Hệ thống hiển thị giao diện thống kê.

3.2. Các luồng rẽ nhánh

a) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt: Không có.

5. Tiền điều kiện: Người quản trị đã đăng nhập vào hệ thống và có quyền truy cập vào chức năng.

6. Hậu điều kiện: Không có.

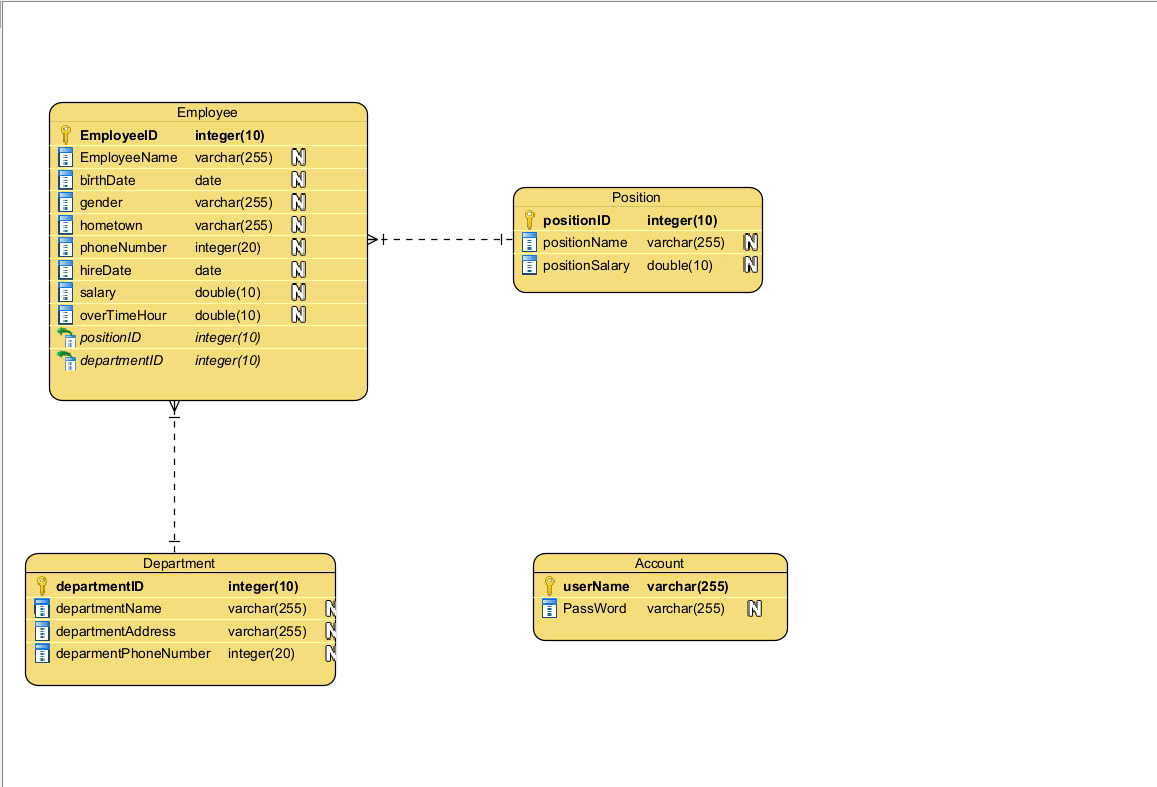
7. Điểm mở rộng: Không có.

**2.3.2. Mô hình hóa dữ liệu của hệ thống**

#### **2.3.2.1. Xác định các lớp và thuộc tính**

* Lớp Position (Chức vụ):
* positionID (int)
* positionName (String)
* positionSalary (double)
* Lớp Account (Tài khoản):
* username (String)
* password (String)
* Lớp Department (Phòng ban):
* departmentID (int)
* departmentName (String)
* departmentAddress (String)
* departmentPhoneNumber (String)
* Lớp Employee (Nhân viên):
* employeeID (int)
* employeeName (String)
* birthDate (Date)
* gender (String)
* hometown(String)
* phoneNumber(String)
* hireDate(date)
* salary (double)
* overtimeHour (float)
* PositionID (int)
* DepartmentID (int)

#### **2.3.2.2. Quan hệ giữa các lớp**



Hình 5: Các lớp sử dụng

## **2.4. Thiết kế hệ thống**

### **2.4.1. Thiết kế dữ liệu**

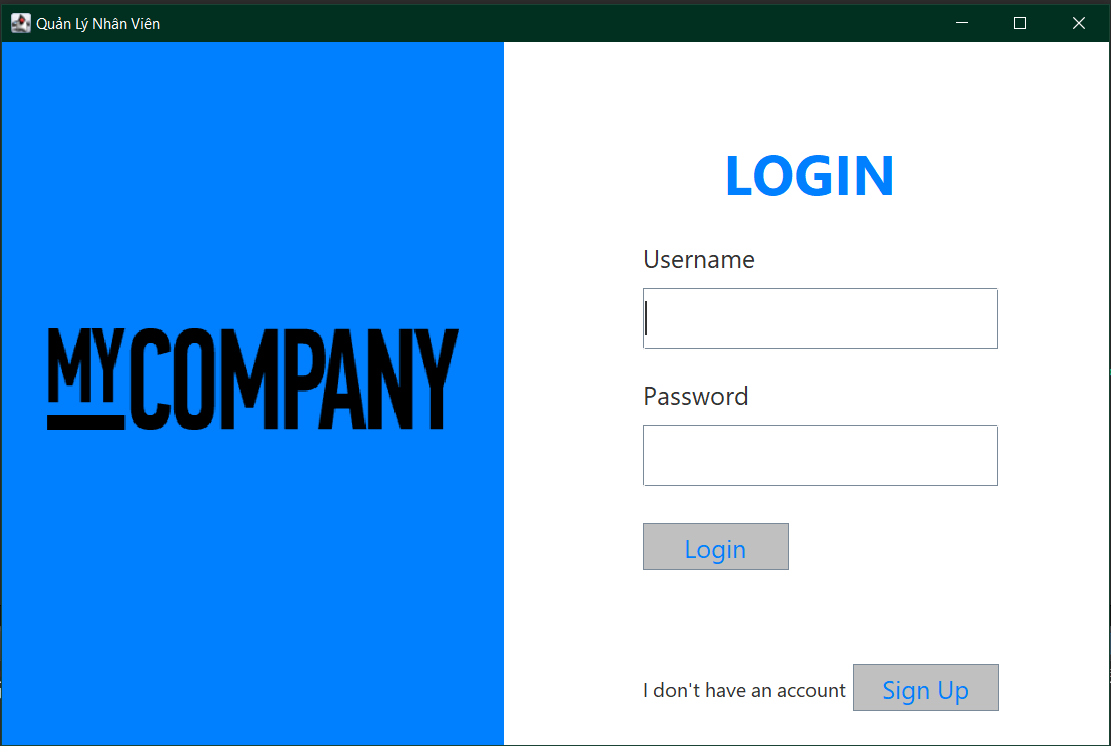
Các bảng:

* **Position (**positionID, positionName, positionSalary)
* **Account** (username, password)
* **Department**(departmentID,departmentName,departmentAddress, departmentPhoneNumber)
* **Employee** (employeeID, employeeName, birthDate,

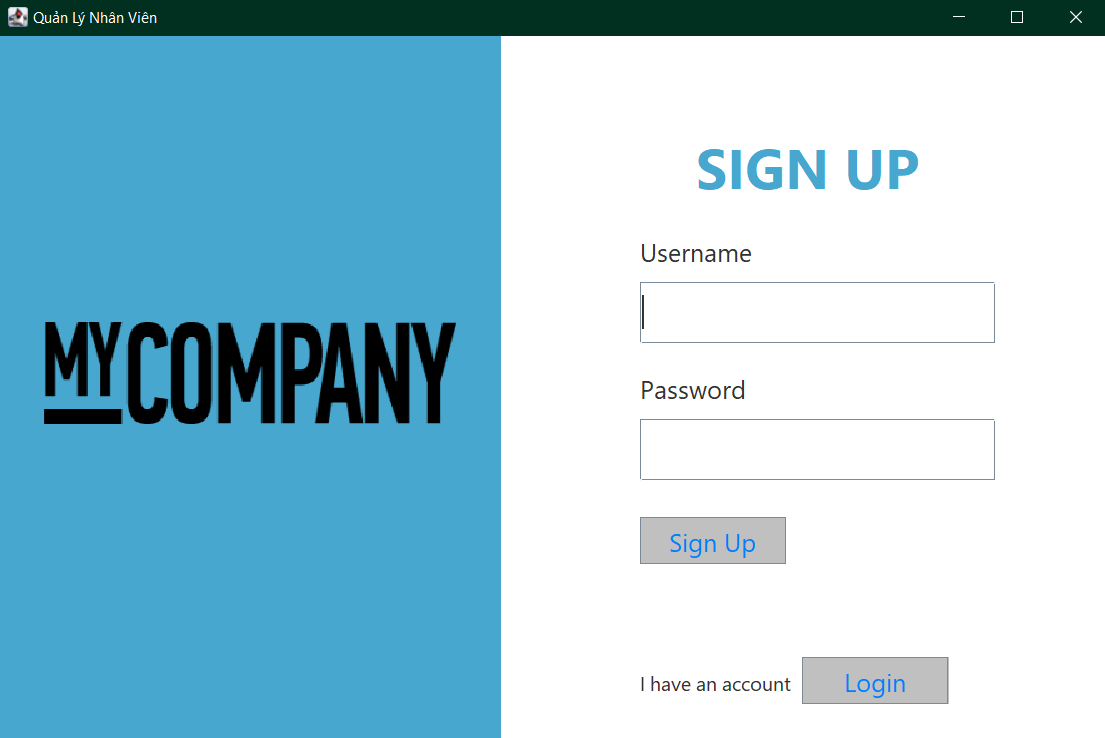
gender,hometown,phoneNumber,hireDate,salary,overtimeHour,

positionID, departmentID)

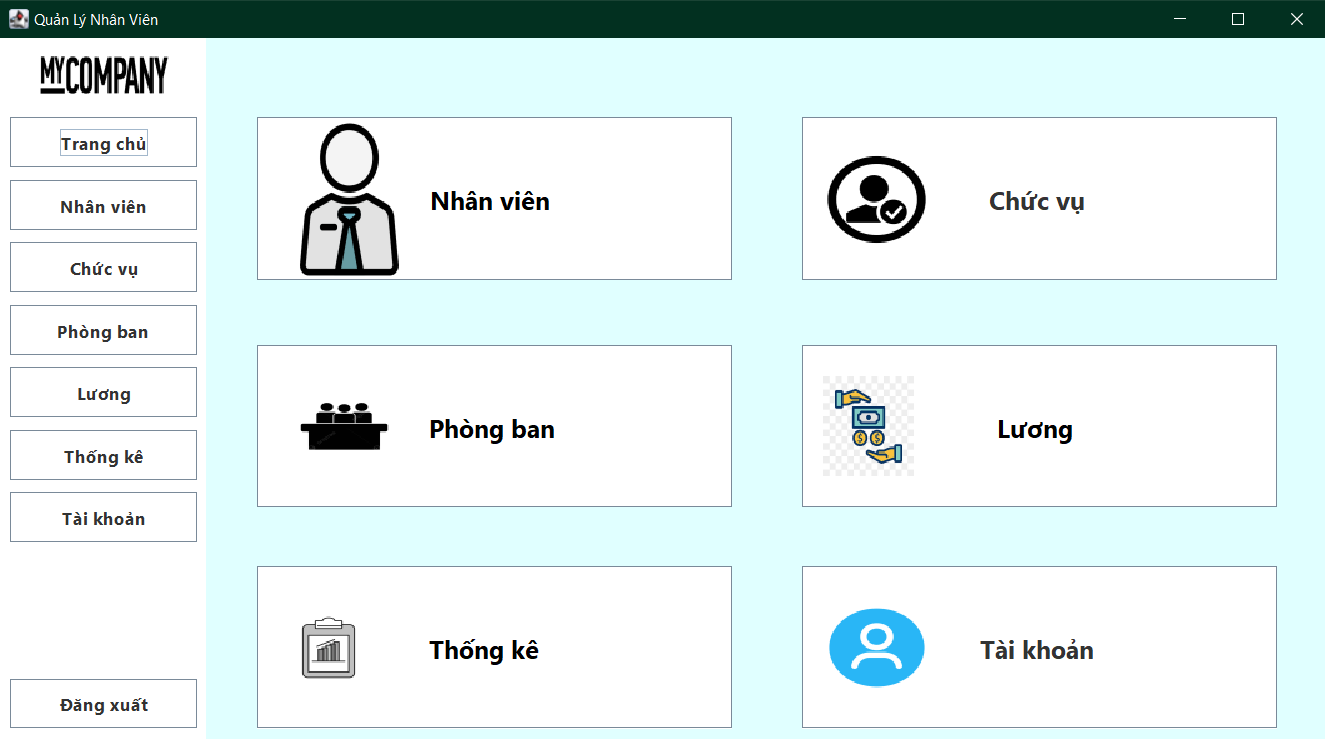
**2.4.2. Thiết kế giao diện**



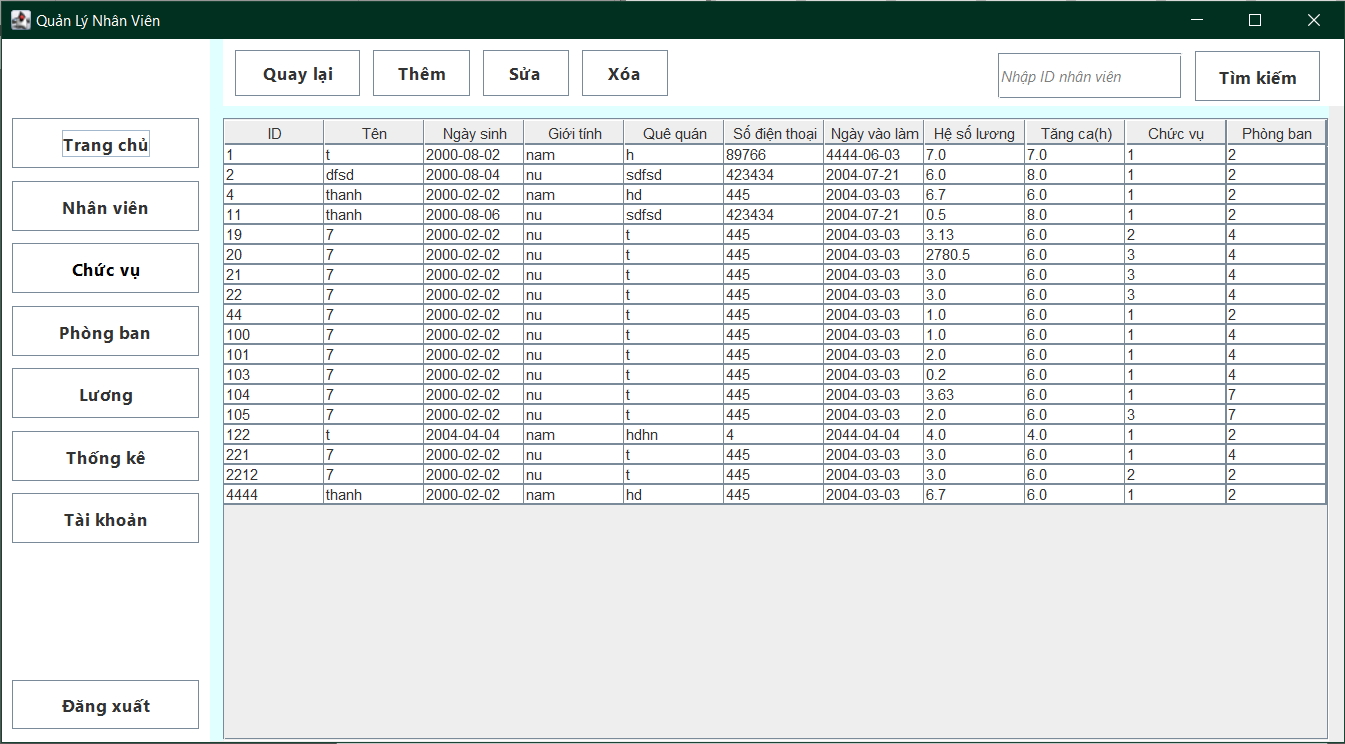
Hình 6: Giao diện đăng nhập



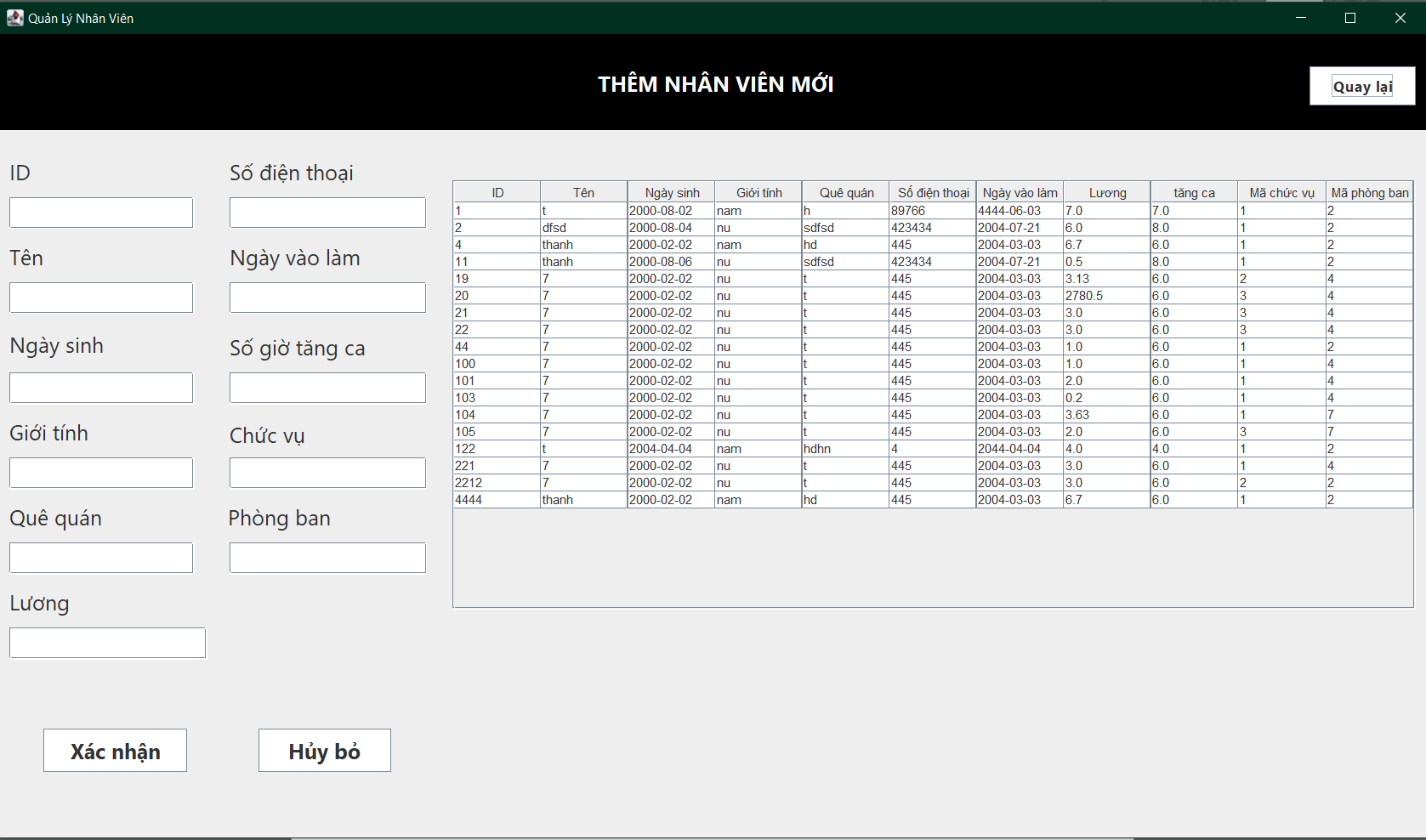
Hình 7: Giao diện đăng kí



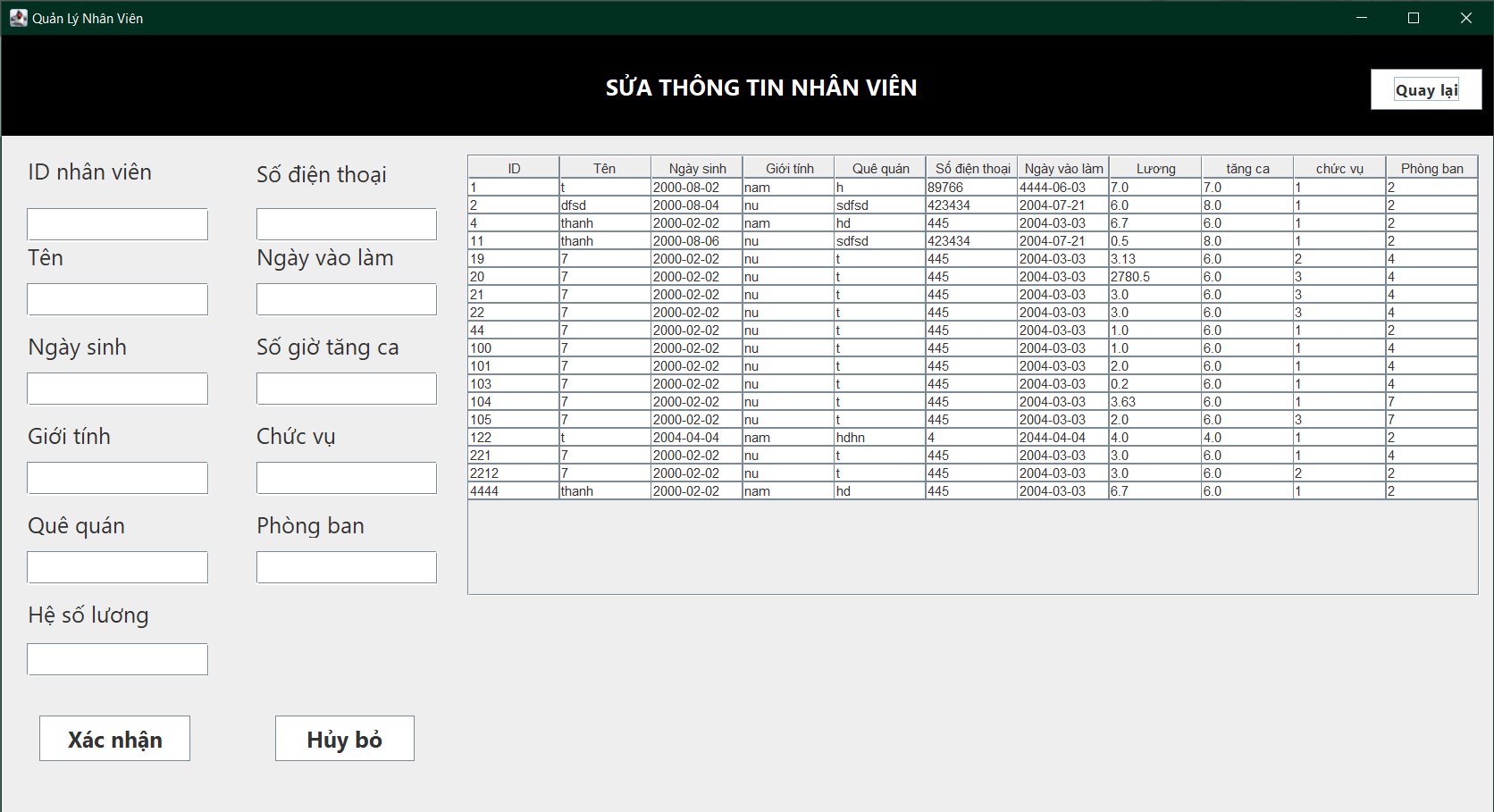
Hình 8: Giao diện trang chủ



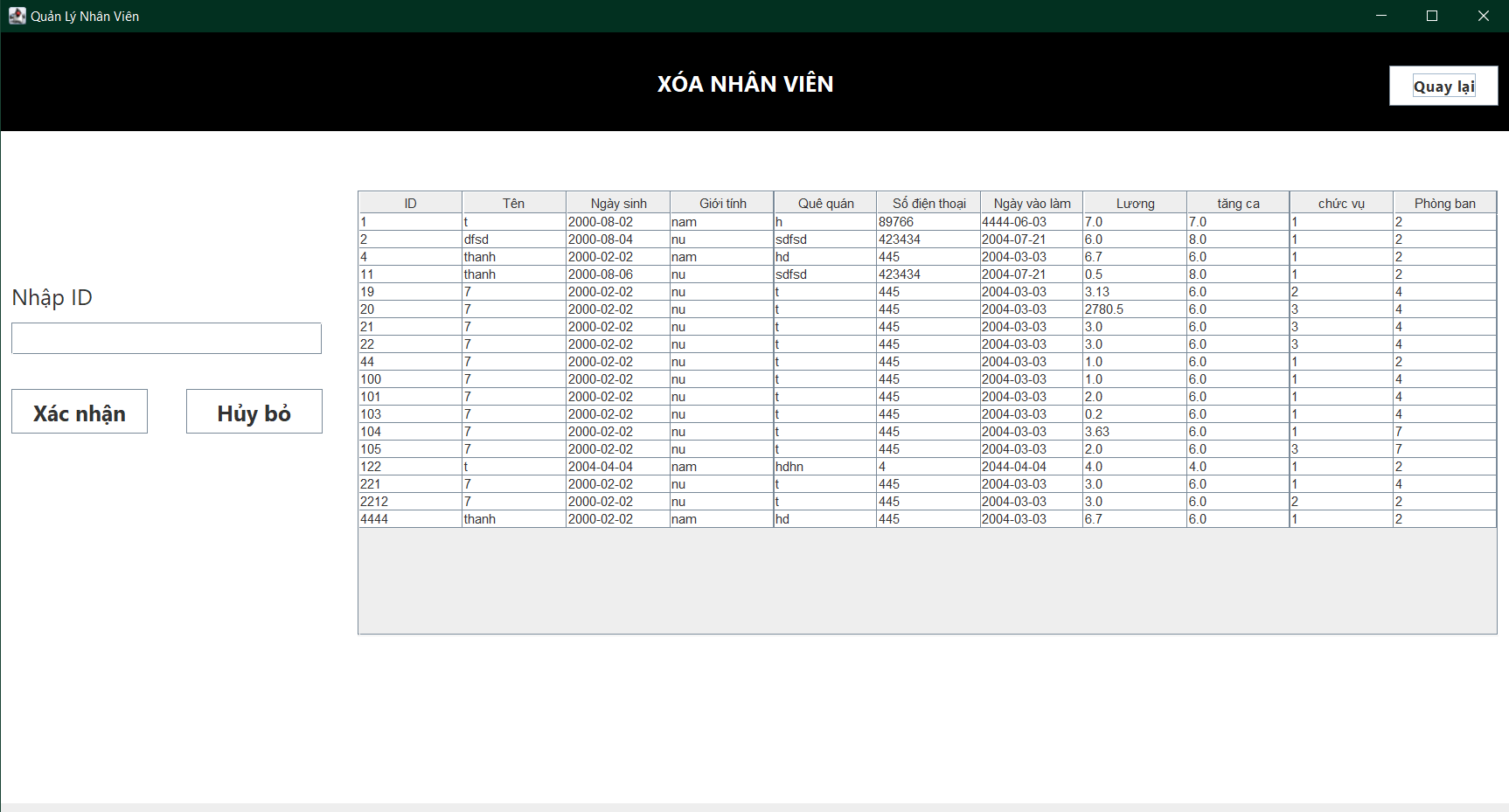
Hình 9: Giao diện quản lý nhân viên



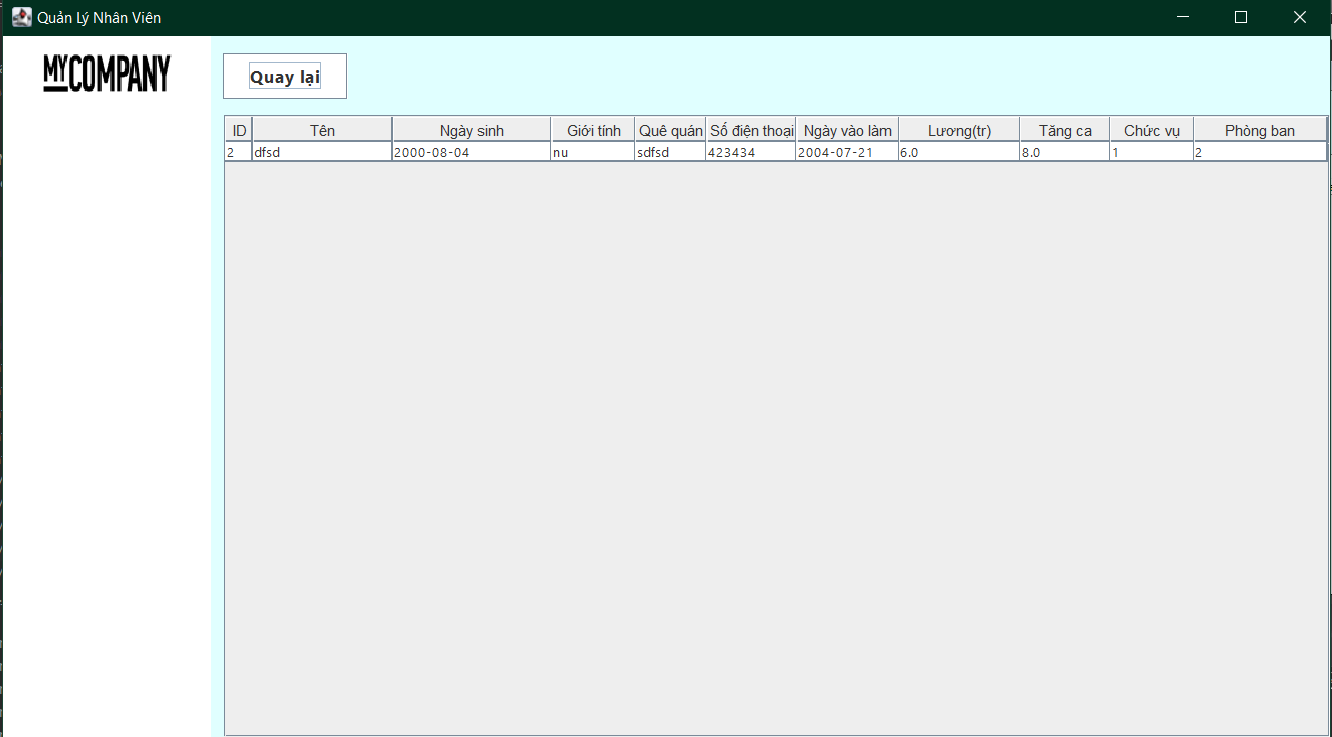
Hình 10: Giao diện thêm nhân viên



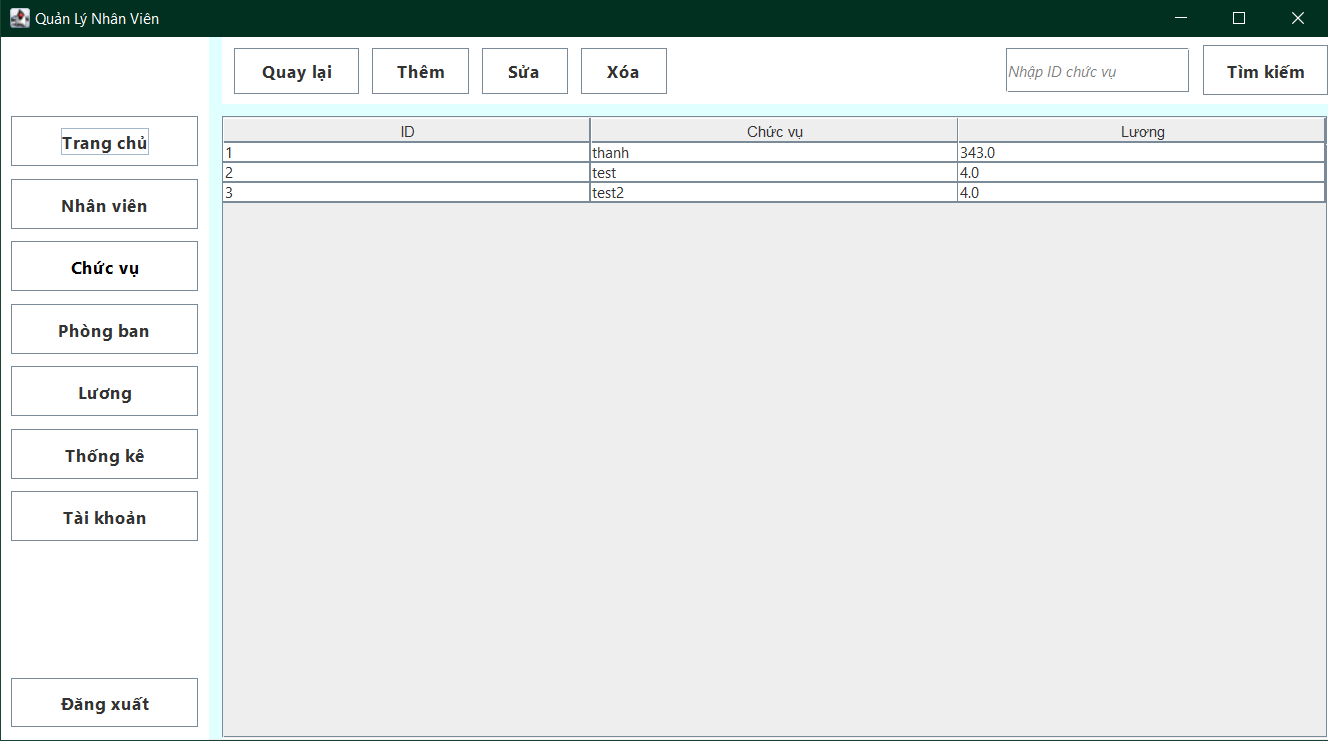
Hình 11: Giao diện sửa thông tin nhân viên



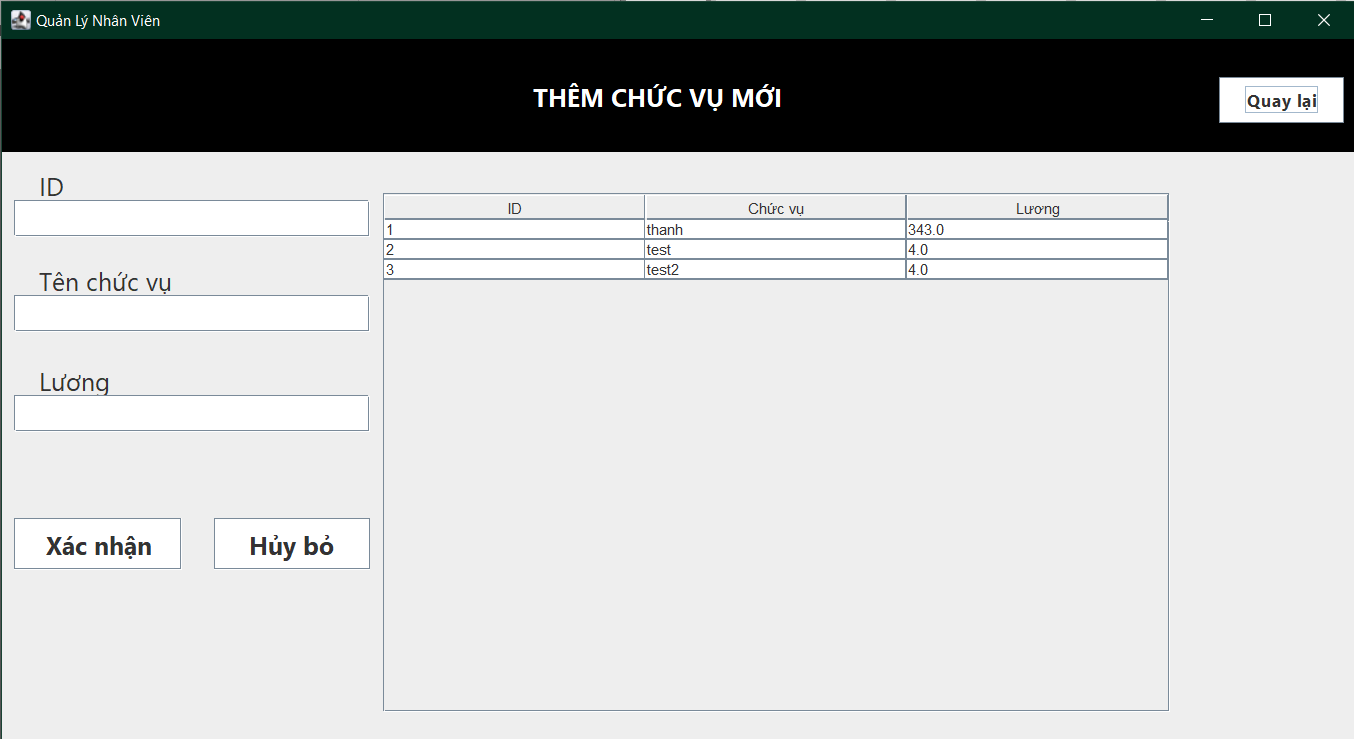
Hình 12: Giao diện xóa nhân viên



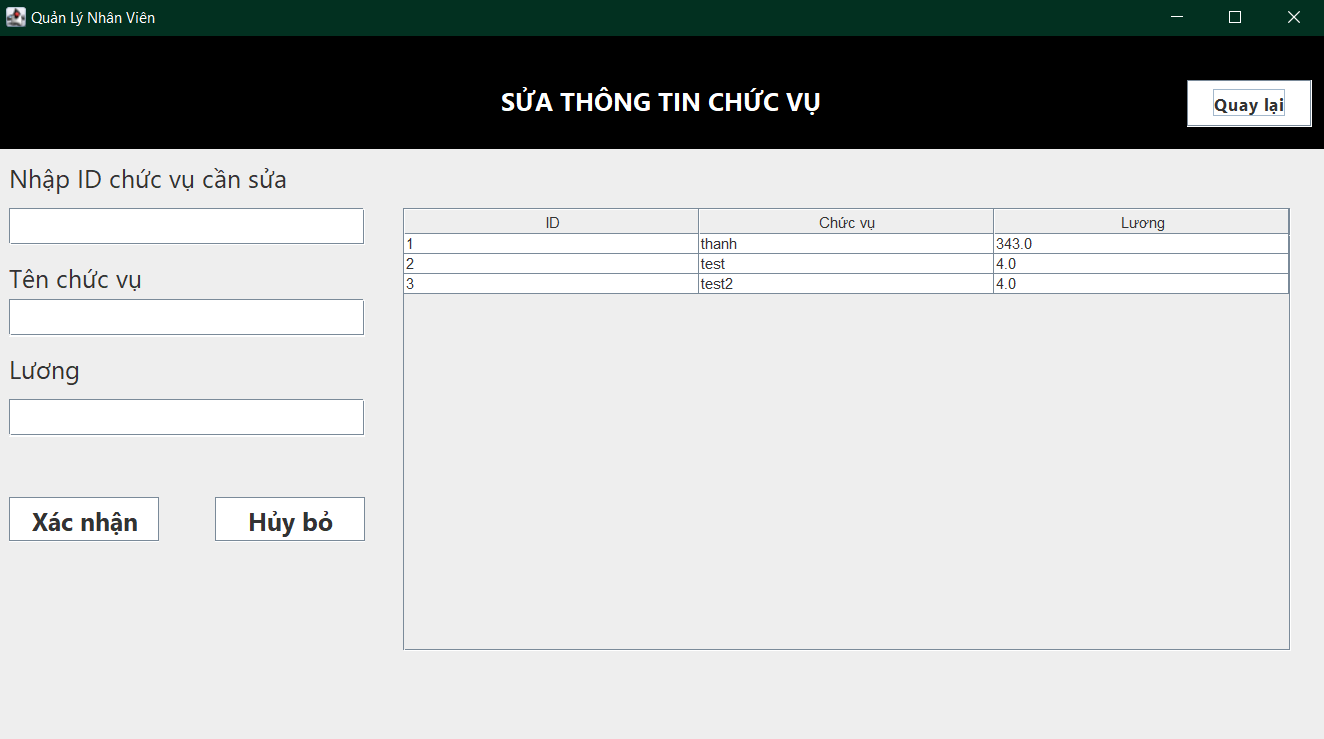
Hình 13: Giao diện tìm kiếm nhân viên



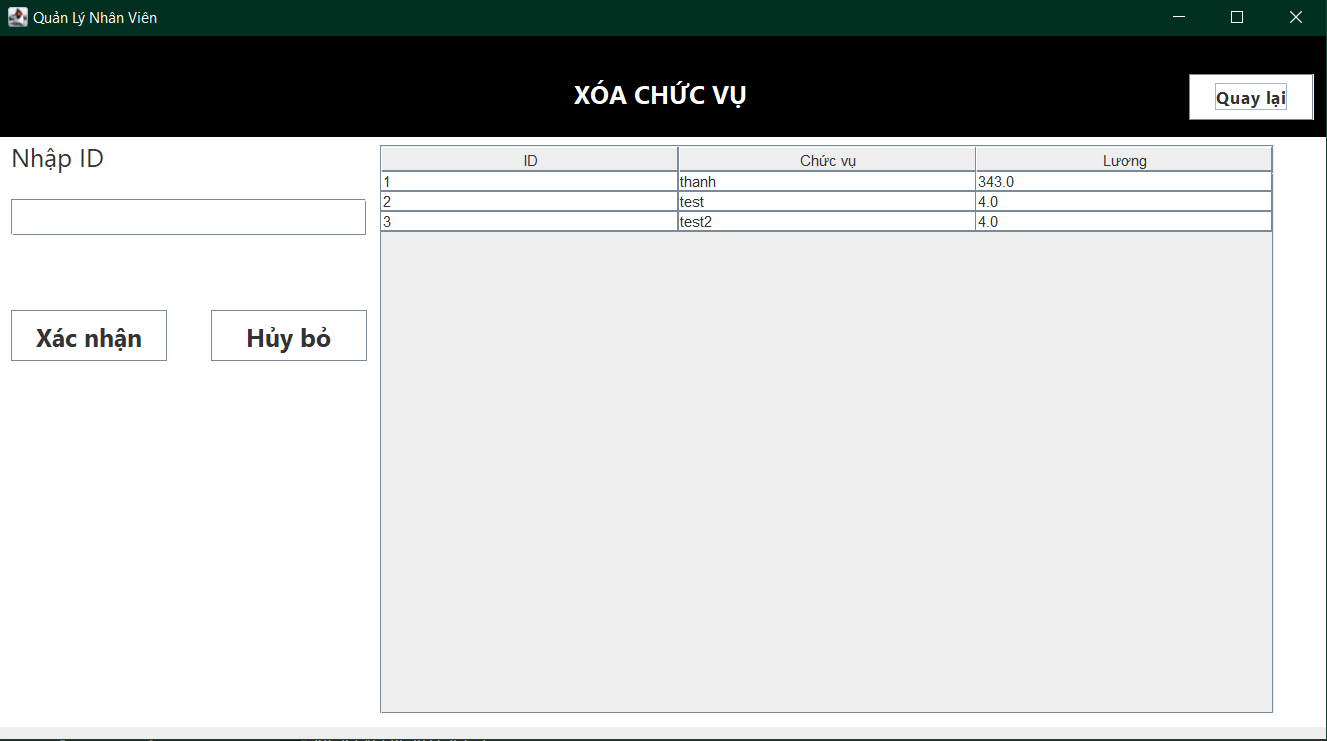
Hình 14: Giao diện quản lý chức vụ



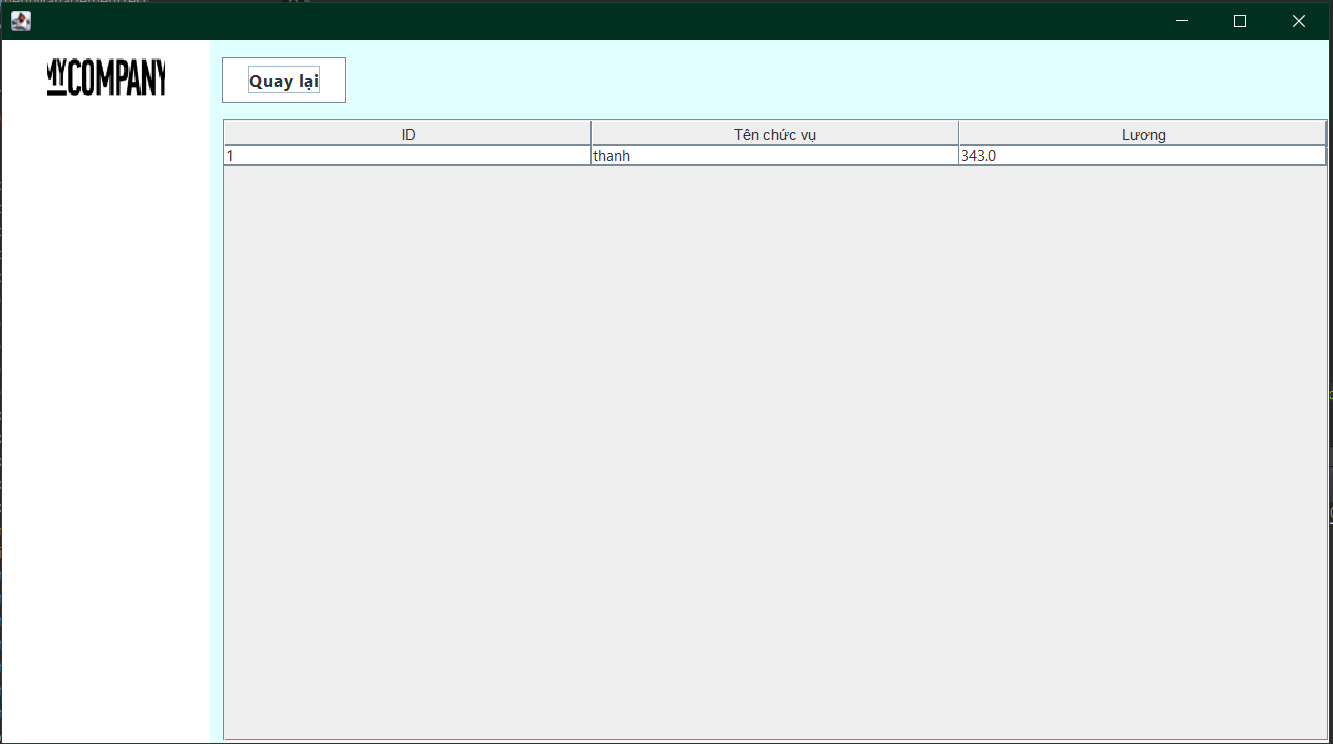
Hình 15: Giao diện thêm chức vụ



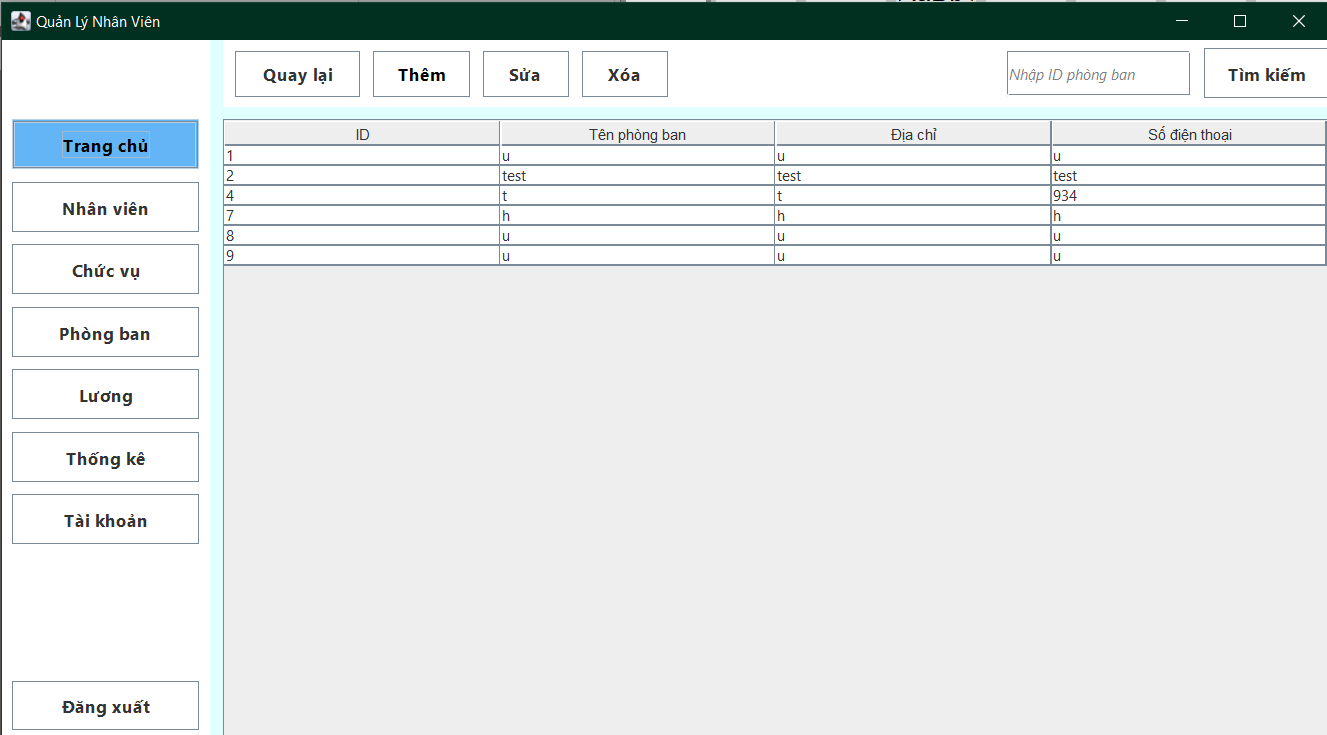
Hình 16: Giao diện sửa thông tin chức vụ



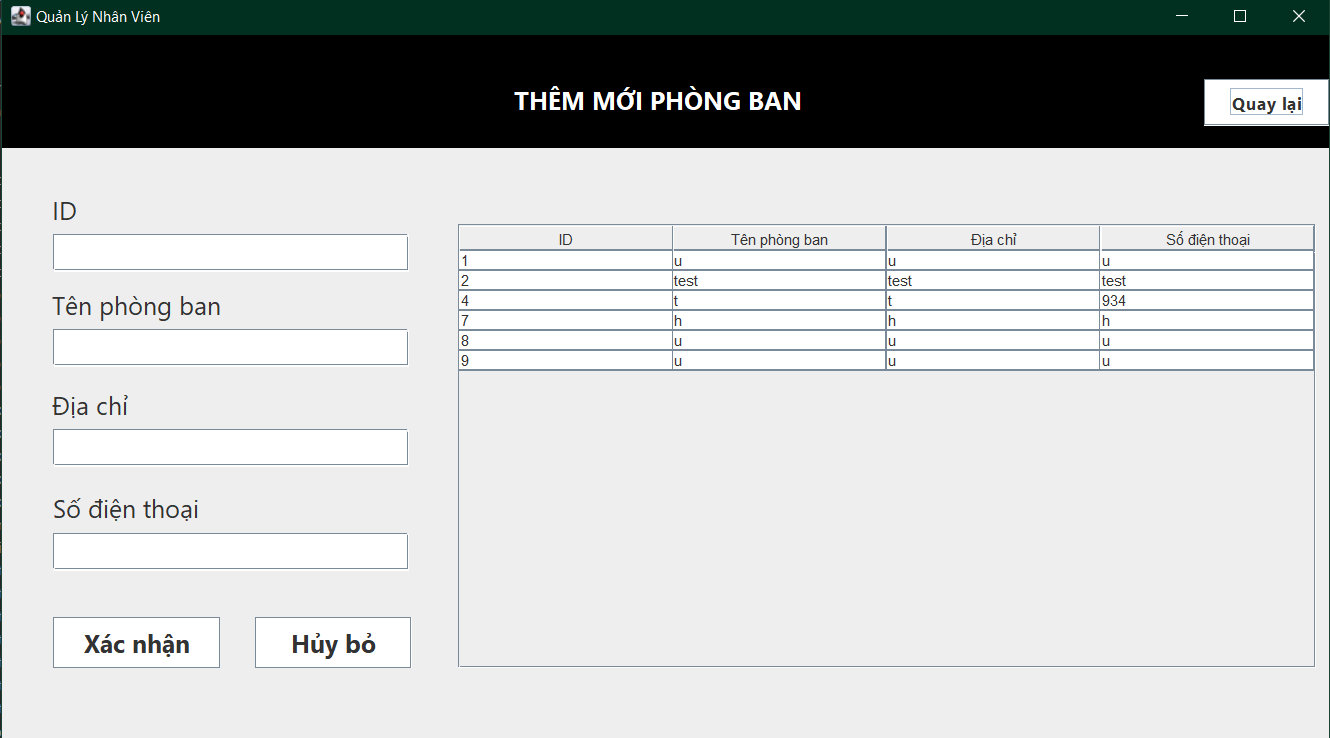
Hình 17: Giao diện xóa chức vụ



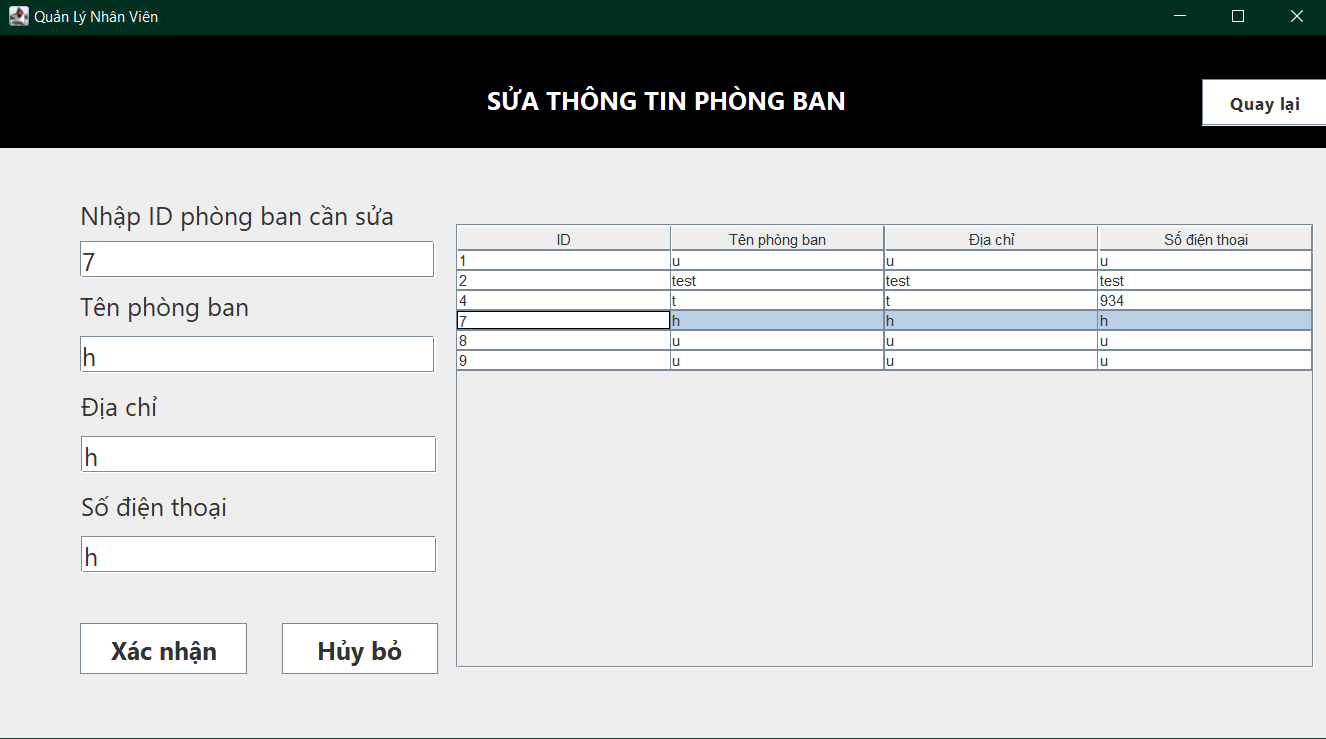
Hình 18: Giao diện tìm kiếm chức vụ



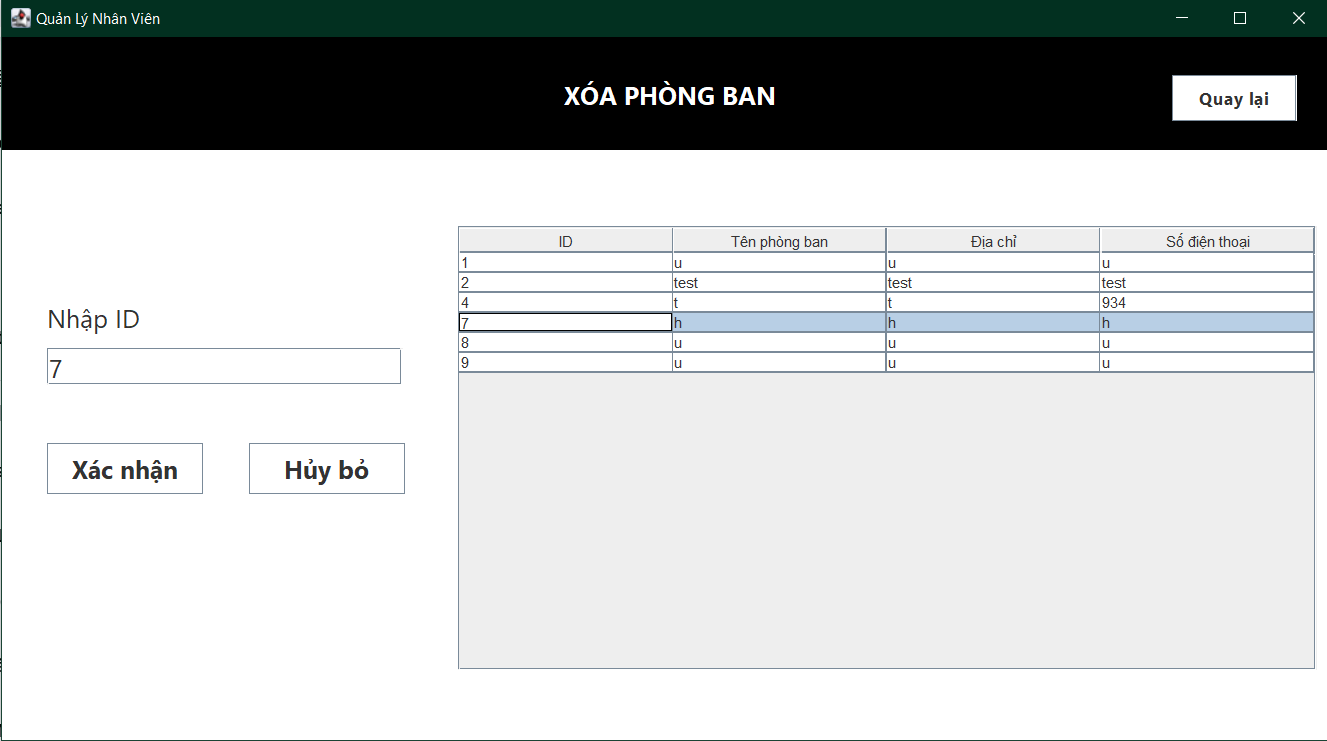
Hình 19: Giao diện quản lý phòng ban



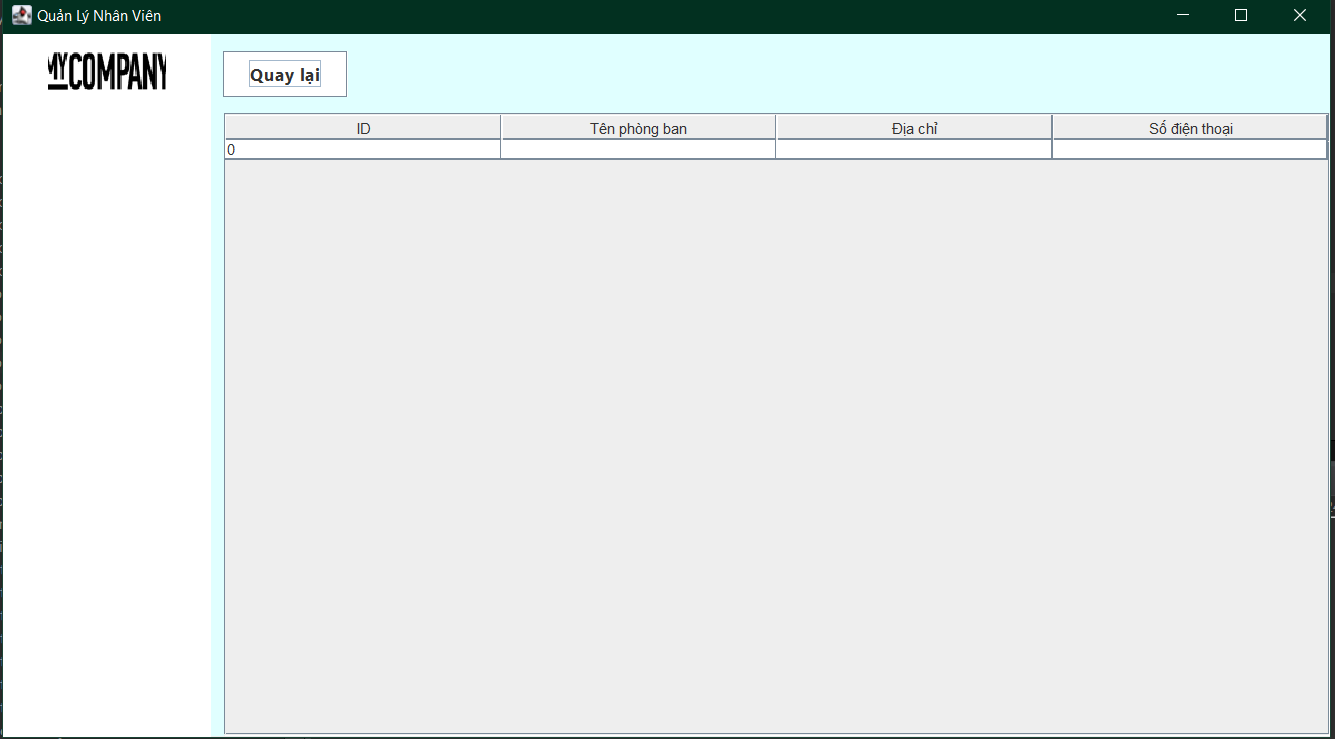
Hình 20: Giao diện thêm mới phòng ban



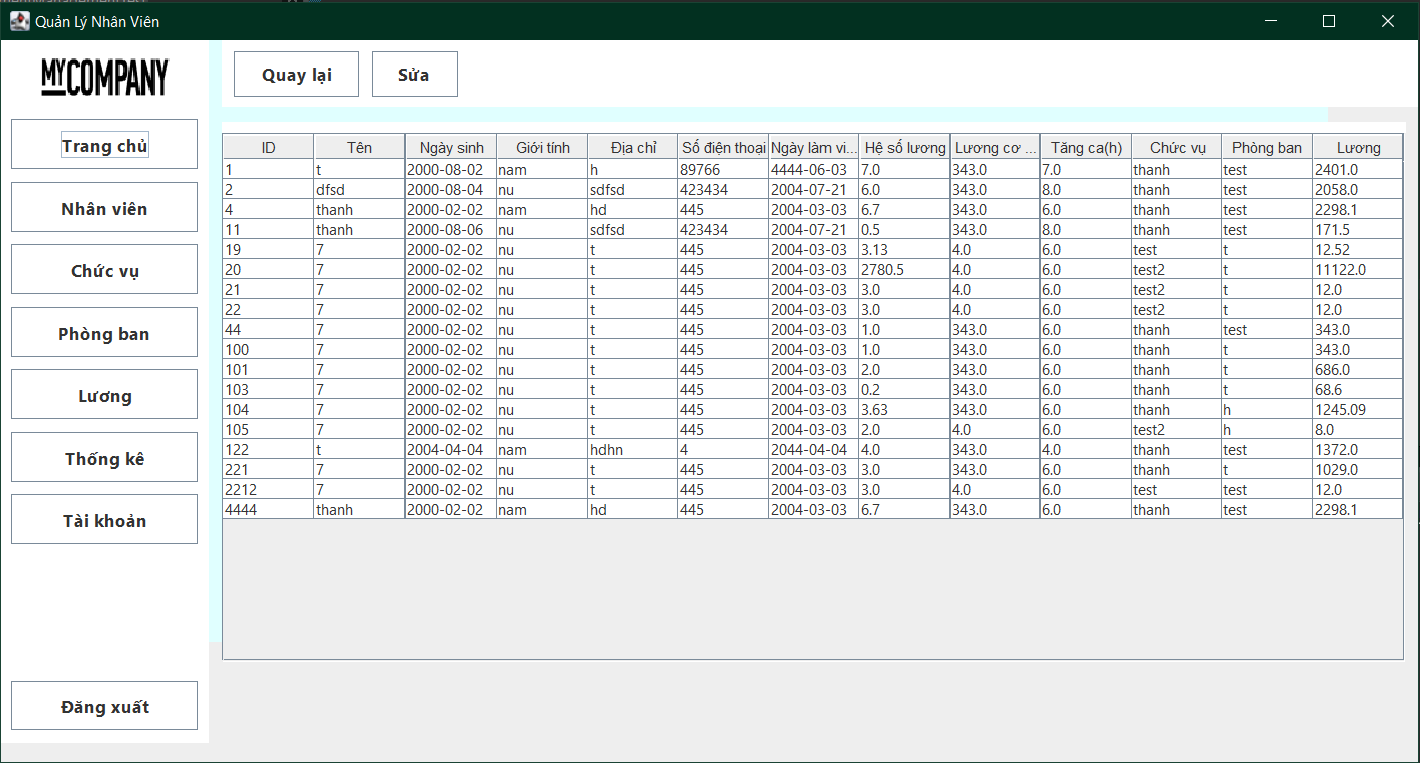
Hình 21: Giao diện sửa thông tin phòng ban



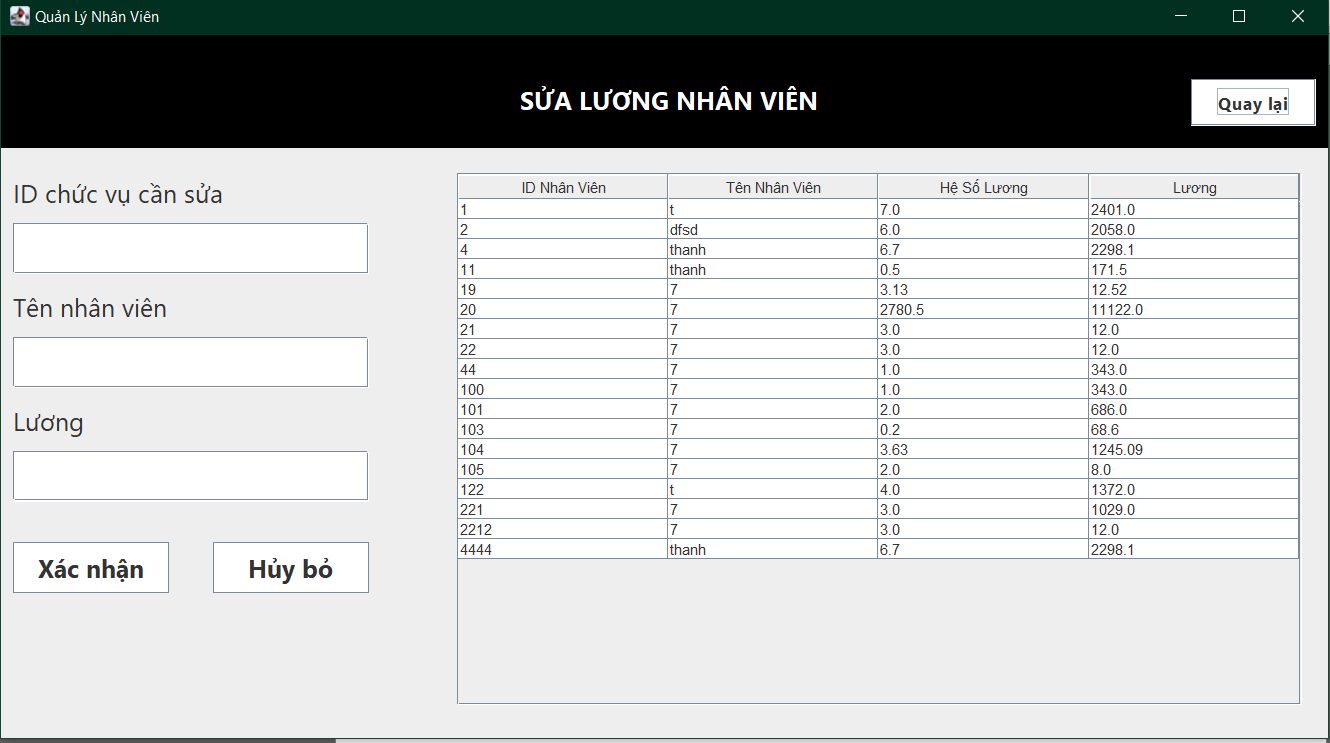
Hình 22: Giao diện xóa phòng ban



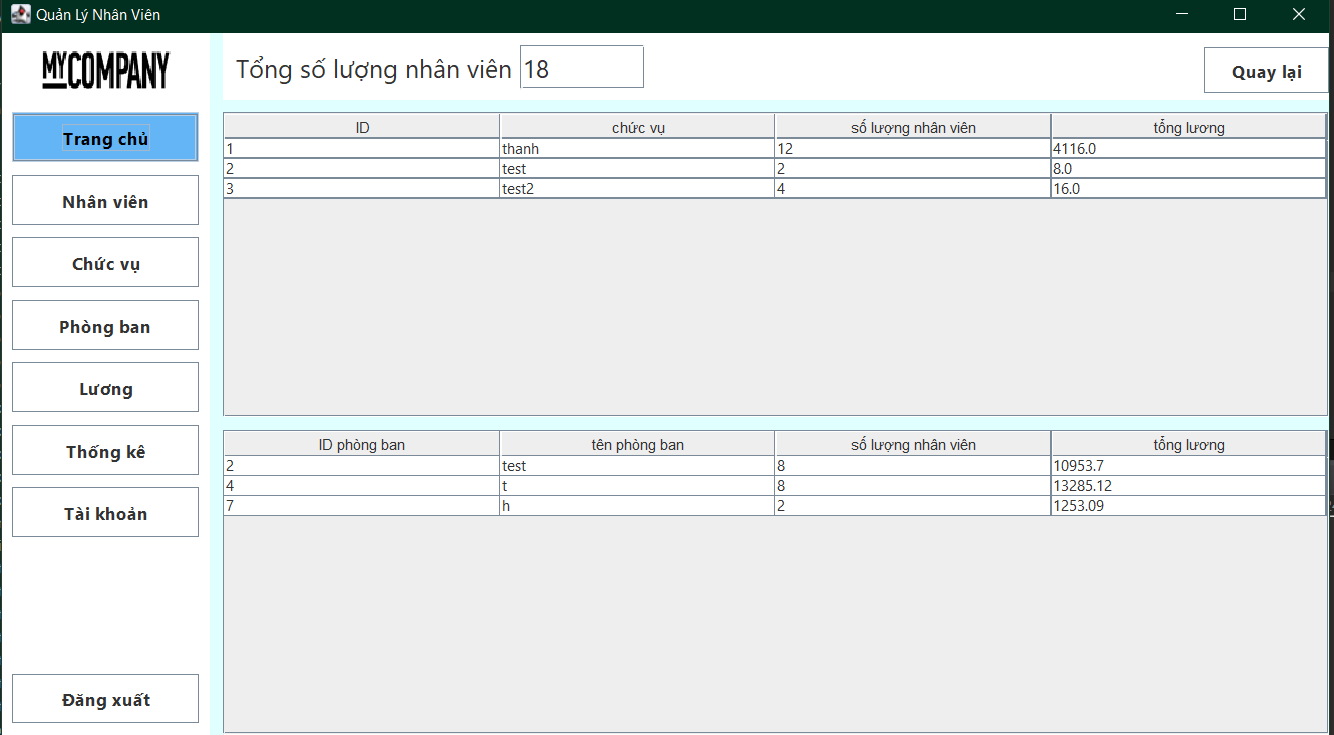
Hình 23: Giao diện tìm kiếm phòng ban



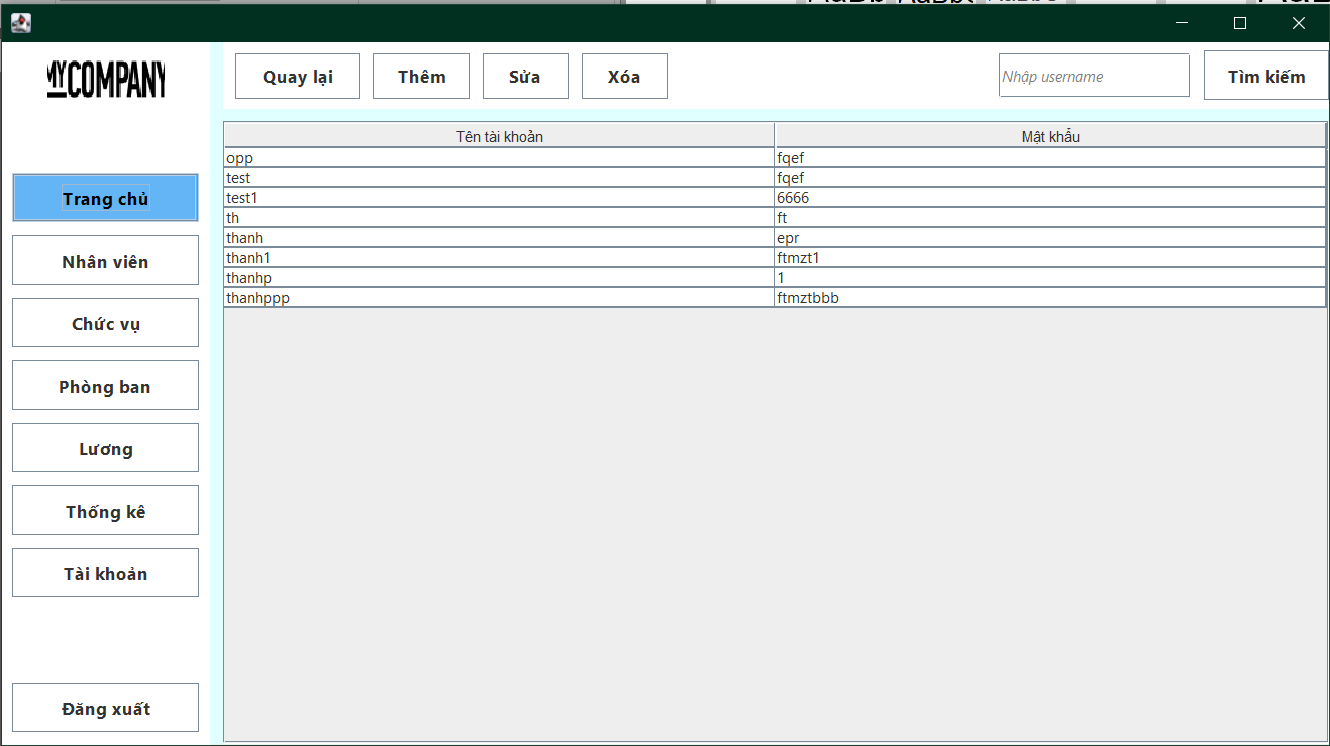
Hình 24:Giao diện quản lý lương, thông tin chi tiết nhân viên



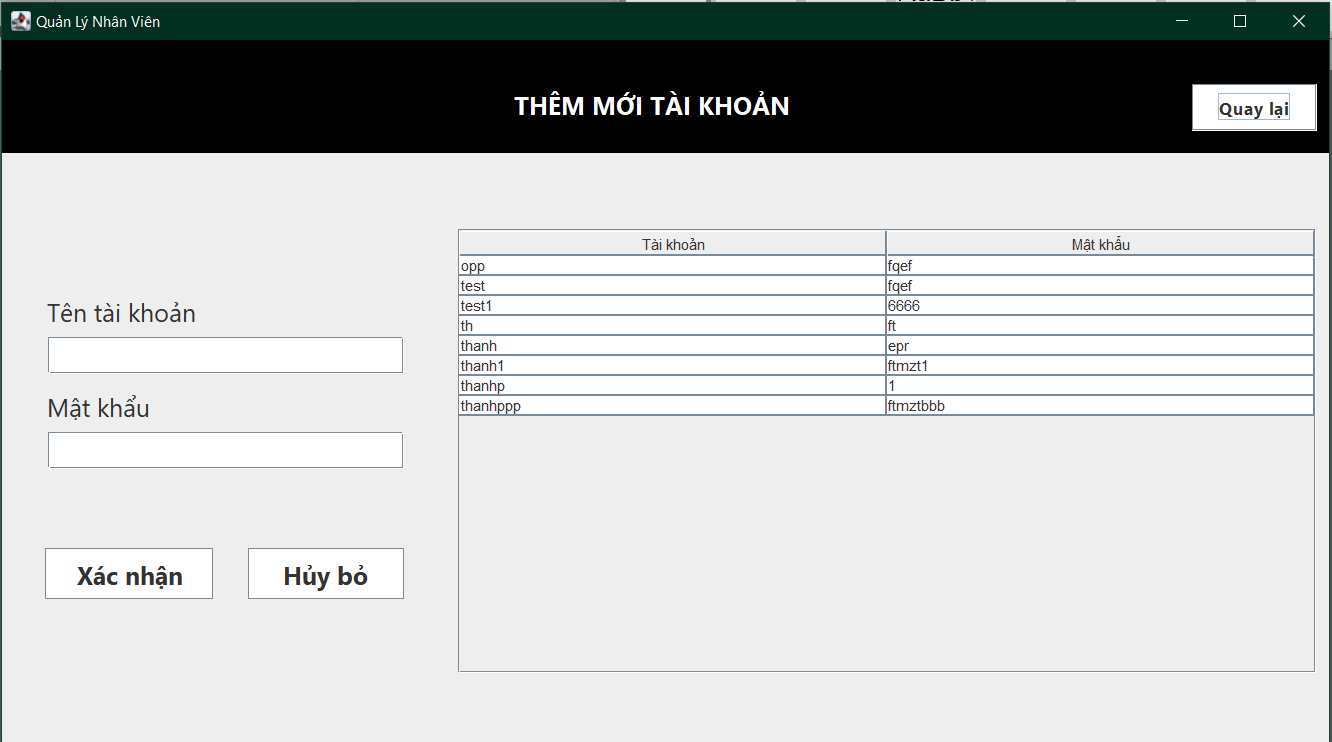
Hình 25: Giao diện sửa hệ số lương theo lương



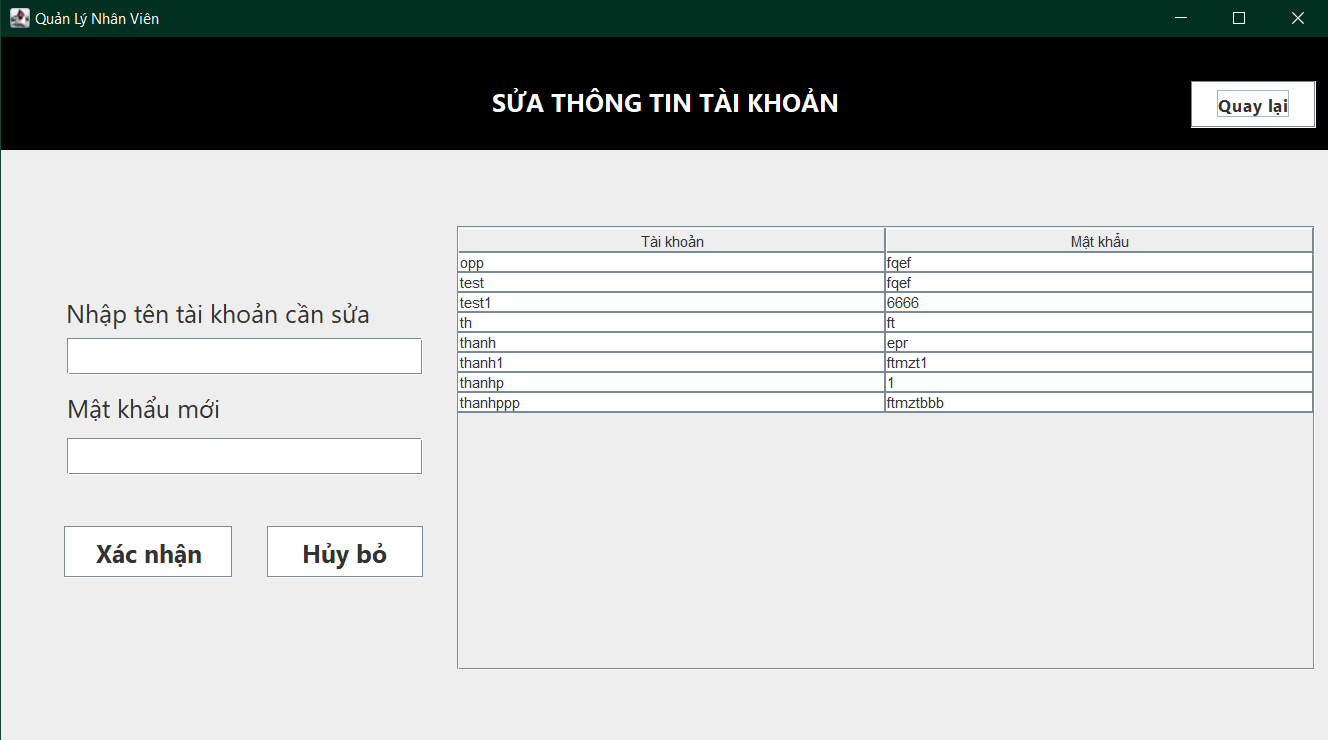
Hình 26: Giao diện thống kê lương, số lượng nhân viên



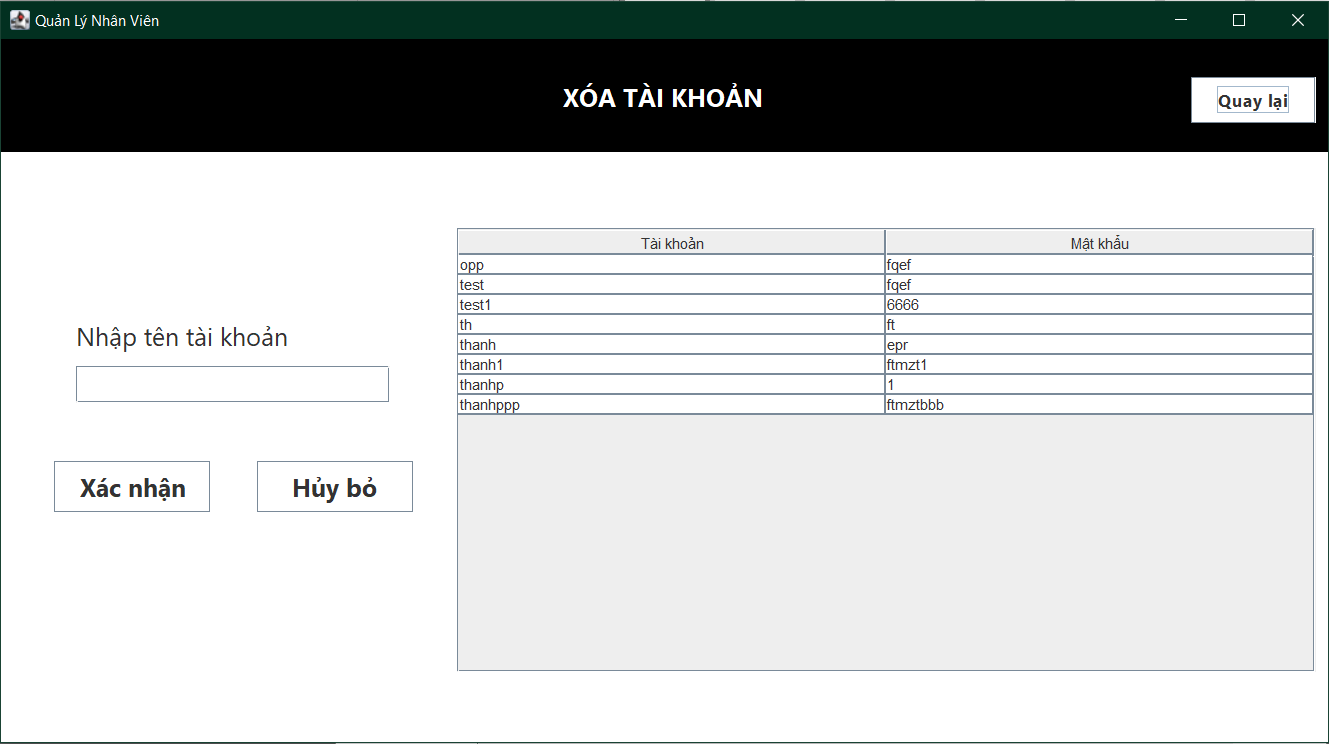
Hình 27: Giao diện quản lý tài khoản đăng nhập



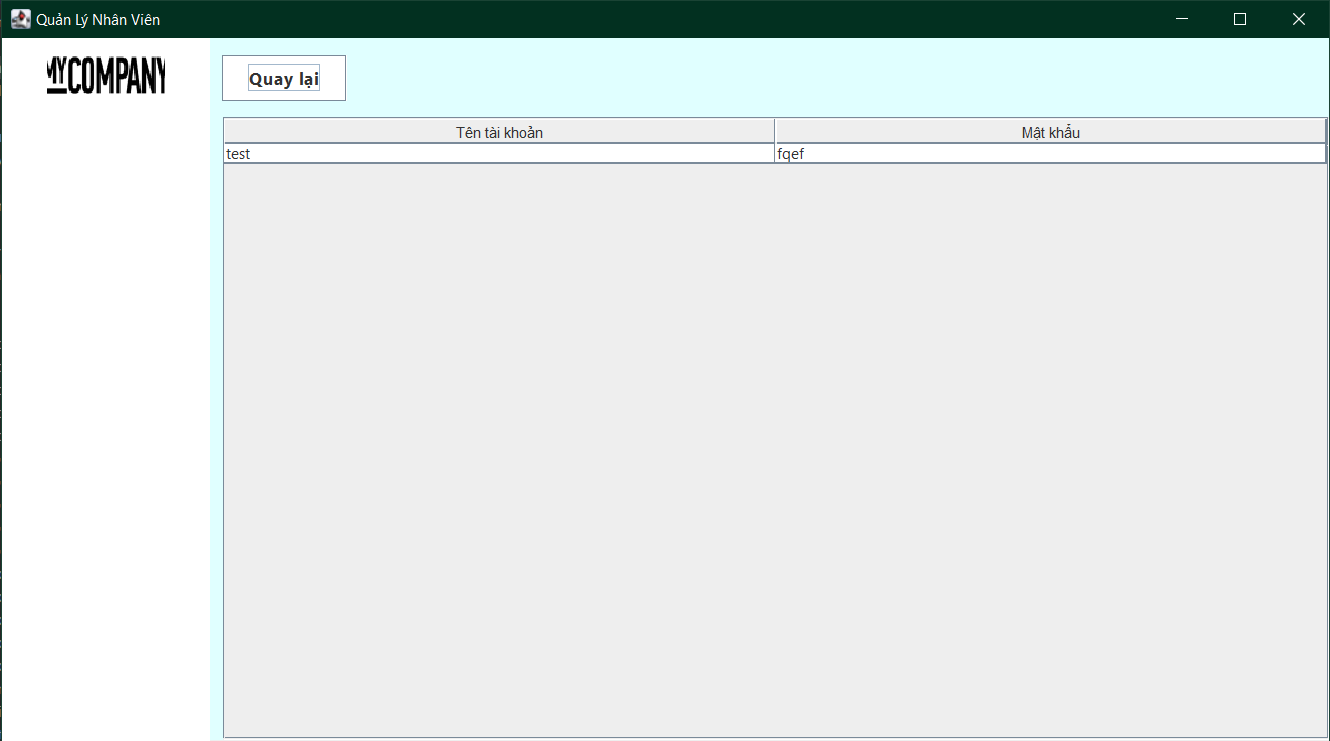
Hình 28: Giao diện thêm mới tài khoản



Hình 29: Giao diện sửa thông tin tài khoản



Hình 30: Giao diện xóa tài khoản



Hình 31: Giao diện tìm kiếm tài khoản

### **2.4.3. Kiểm tra thử nghiệm**

### Kế hoạch kiểm thử

Mục tiêu kiểm thử: Đảm bảo ứng dụng hoạt động đúng chức năng, giao diện thân thiện, hiệu suất tốt và an toàn bảo mật.

Các loại kiểm thử được thực hiện:

* Kiểm thử chức năng (Functional Testing)
* Kiểm thử giao diện người dùng (UI Testing)
* Kiểm thử tích hợp (Integration Testing)
* Kiểm thử hiệu năng (Performance Testing)
* Kiểm thử an toàn bảo mật (Security Testing)

1. Kết quả kiểm thử

2.1. Kiểm thử chức năng

* Kiểm tra các chức năng chính như quản lý thông tin nhân viên, quản lý phòng ban, chức vụ, tài khoản v.v.
* Các chức năng đều hoạt động đúng theo yêu cầu và mô tả.
* Xử lý các trường hợp đầu vào hợp lệ và không hợp lệ.
* Các thao tác C-R-U-D (Tạo, Đọc, Cập nhật, Xóa) dữ liệu hoạt động chính xác.

2.2. Kiểm thử giao diện người dùng

* Giao diện được thiết kế rõ ràng, dễ hiểu và thân thiện với người dùng.
* Các thành phần giao diện như nút nhấn, văn bản, biểu tượng hiển thị đúng.
* Tương tác người dùng như click, nhập liệu, điều hướng diễn ra trơn tru.

2.3. Kiểm thử tích hợp

* Các module/thành phần của ứng dụng tích hợp và hoạt động đúng khi kết hợp với nhau.
* Luồng dữ liệu và trao đổi thông tin giữa các thành phần diễn ra chính xác.

2.4. Kiểm thử hiệu năng

* Ứng dụng đáp ứng tốc độ phản hồi và thời gian xử lý tương đối nhanh.
* Sử dụng tài nguyên hệ thống (bộ nhớ, CPU) trong giới hạn chấp nhận được.
* Ứng dụng vẫn hoạt động ổn định khi có nhiều người dùng truy cập cùng lúc.

2.5. Kiểm thử an toàn bảo mật

* Phân quyền truy cập người dùng được thiết lập đúng.
* Dữ liệu nhân viên được bảo vệ an toàn.

1. Đánh giá và kết luận

* Dự án quản lý nhân sự đáp ứng các yêu cầu chức năng và phi chức năng đã đề ra.
* Giao diện thân thiện, dễ sử dụng, các chức năng hoạt động ổn định.
* Hiệu năng và tính bảo mật của ứng dụng đạt yêu cầu.
* Dự án có thể được triển khai và đưa vào sử dụng.

**CHƯƠNG 3: TỔNG KẾT**

## **3.1. Nội dung đã thực hiện**

Phân tích nội dung cần thực hiện, các chức năng, dữ liệu liên quan để xây dựng

hệ thống. Phân tích và đặc tả qua đó xác định những tác nhân, những use case của hệ thống. Áp dụng các kiến thức các thuật toán cùng ngôn ngữ lập trình java để áp dụng hoàn thành các chức năng như yêu cầu của hệ thống. Áp dụng giao diện đồ họa GUI vào thiết kế giao diện, form của hệ thống. Thành thạo Collections, bắt lỗi xử lý ngoại lệ, đọc lưu dữ liệu vào cơ sở dữ liệu, thiết kế giao diện hệ thống. Phân chia công việc giữa các thành viên theo khả năng lập trình đảm bảo kết quả được thống nhất, hiệu quả.

Sau bài tiểu luận các thành viên trong nhóm rút ra được những bài học

● Tổ chức, phân chia công việc sao cho phù hợp với mỗi người mà vẫn đảm bảo

tính hiệu quả và thống nhất cho bài tiểu luận.

● Học hỏi, giúp đỡ nhau trong việc hoàn thành chương trình hệ thống.

● Tổ chức một cách liên kết, trao đổi, họp để phân chia công việc cũng như bàn

cách giải quyết khi gặp vấn đề trong việc xây dựng hệ thống.

## **3.2. Kết quả đạt được**

1. Giao Diện Người Dùng (UI/UX):
   * Thiết kế giao diện người dùng trực quan và thân thiện: Ứng dụng có giao diện dễ sử dụng, hỗ trợ tốt cho người dùng cuối.
   * Phát triển các màn hình chính: Bao gồm màn hình đăng nhập, dashboard, quản lý nhân viên, quản lý lương thưởng, quản lý phòng ban, v.v.
2. Chức Năng Quản Lý Nhân Sự:
   * Quản lý thông tin nhân viên: Thêm mới, chỉnh sửa, xóa thông tin nhân viên; quản lý ngày làm việc.
   * Quản lý phòng ban: Ghi nhận được mã phòng,tên phòng,số điện thoại và địa chỉ.
   * Quản lý chức vụ: Ghi nhận được mã chức vụ, tên chức vụ, lương của chức vụ.
   * Quản lý lương: Tính toán lương, thưởng, các khoản phụ cấp và khấu trừ; phát hành phiếu lương.
   * Quản lý tài khoản: Thêm mới, chỉnh sửa, xóa tài khoản.
3. Phản Hồi và Điều Chỉnh:
   * Thu thập phản hồi từ người dùng: Dựa trên phản hồi của người dùng cuối để điều chỉnh và cải tiến ứng dụng.
   * Cải tiến và cập nhật: Định kỳ cập nhật các tính năng mới và sửa lỗi để nâng cao trải nghiệm người dùng.
4. Tính Năng Bổ Trợ:
   * Báo cáo và thống kê: Tạo các báo cáo về nhân sự, lương thưởng; biểu đồ thống kê tình hình nhân sự.

**3.3. Hướng phát triển**

Qua việc xây dựng và phát triển hệ thống lần này nhóm chúng em nhận thấy việc phân tích cũng như định hướng cho phần mềm theo yêu cầu đòi hỏi các thành viên phải tìm hiểu kĩ cũng như đóng góp, xây dựng theo đúng hướng chức năng của người dùng, về cách thức xây dựng hệ thống bám sát theo những kiến thức đã học được ở bộ môn lập trình Java. Ban đầu sau khi nhận đề tài chúng em gặp khó khăn trong việc xác định những công việc cần làm và phân chia với các thành viên sao cho phù hợp nhất nhưng sau một thời gian các thành viên trong nhóm thảo luận, trao đổi cũng như đóng góp ý kiến nhóm đã xác định được hướng đi cần thiết để xây dựng hệ thống sao cho hiệu quả nhất. Sau khi hoàn thiện hệ thống tất cả các thành viên trong nhóm đã hiểu hơn về phần thiết kế giao diện đồ họa cũng như biết các bước cần thiết từ khâu lên ý tưởng đến xây dựng hệ thống hoàn chỉnh. Do khả năng lập trình còn một số hạn chế cũng như việc chênh lệch trong khả năng lập trình giữa các thành viên trong nhóm nên không tránh khỏi những sai sót trong hệ thống. Kính mong thầy cô và các bạn có thể đóng góp ý kiến để nhóm chúng em có thể khắc phục những tồn tại và phát triển hệ thống được bảo đảm hơn.

## **3.4. Tính khả thi và mở rộng**

1. Java Swing:
   * Java Swing là một phần của Java Foundation Classes (JFC) và được sử dụng rộng rãi để phát triển các ứng dụng desktop với giao diện người dùng đồ họa (GUI).
   * Swing cung cấp một bộ công cụ phong phú cho việc tạo giao diện người dùng, bao gồm các thành phần như button, text field, table, và tree, giúp dễ dàng xây dựng các ứng dụng quản lý nhân sự.
2. Phát Triển Nhanh và Dễ Dàng:
   * Java Swing có tài liệu phong phú và nhiều ví dụ minh họa, giúp lập trình viên dễ dàng bắt đầu và phát triển ứng dụng.
   * Swing hỗ trợ thiết kế giao diện theo mô hình MVC (Model-View-Controller), giúp tách biệt logic xử lý và giao diện người dùng, tạo điều kiện thuận lợi cho việc bảo trì và mở rộng.
3. Khả Năng Tương Thích Cao:
   * Ứng dụng Swing chạy trên MVC(Model-View-Controller) nên có tính đa nền tảng cao, chạy được trên nhiều hệ điều hành như Windows, macOS, và Linux.
4. Hiệu Suất và Bảo Mật:
   * Java là một ngôn ngữ hiệu suất cao, và Swing tận dụng được sức mạnh của Java để cung cấp trải nghiệm người dùng mượt mà.
   * Hệ thống bảo mật của Java giúp bảo vệ dữ liệu người dùng và ngăn chặn các mối đe dọa từ bên ngoài.

### **Tính Mở Rộng**

1. Kiến Trúc Phân Lớp:
   * Thiết kế ứng dụng theo kiến trúc phân lớp, với các lớp Presentation, Business Logic, và Data Access, giúp dễ dàng mở rộng và bảo trì ứng dụng.
   * Sử dụng các mô hình thiết kế như MVC, DAO (Data Access Object) để quản lý logic nghiệp vụ và tương tác với cơ sở dữ liệu.
2. Tích Hợp Công Nghệ Mới:
   * Dễ dàng tích hợp các thư viện Java khác, chẳng hạn như Hibernate cho ORM (Object-Relational Mapping) và Spring Framework cho quản lý các nghiệp vụ phức tạp.
   * Khả năng tích hợp với các dịch vụ web (RESTful APIs) để mở rộng chức năng và kết nối với các hệ thống khác.
3. Mở Rộng Chức Năng:
   * Thêm các module hoặc plugin mới để mở rộng chức năng của ứng dụng quản lý nhân sự, chẳng hạn như quản lý đào tạo, đánh giá hiệu suất, và quản lý dự án.
   * Thiết kế giao diện người dùng linh hoạt, cho phép tùy chỉnh và thêm mới các màn hình và chức năng theo yêu cầu của doanh nghiệp.
4. Quản Lý Dữ Liệu Lớn:
   * Sử dụng các công nghệ như Apache Kafka và Apache Hadoop để xử lý và phân tích dữ liệu lớn, giúp cung cấp các báo cáo và dự đoán thông minh.
   * Tối ưu hóa cơ sở dữ liệu và sử dụng caching (bộ nhớ đệm) để cải thiện hiệu suất ứng dụng khi xử lý lượng dữ liệu lớn.
5. Hỗ Trợ Đa Ngôn Ngữ và Đa Văn Hóa:
   * Ứng dụng có thể được quốc tế hóa (i18n) để hỗ trợ nhiều ngôn ngữ và văn hóa khác nhau, giúp dễ dàng triển khai ở nhiều quốc gia.
6. Phân Quyền và Bảo Mật:
   * Tích hợp các hệ thống quản lý quyền truy cập và xác thực người dùng để đảm bảo chỉ những người dùng có quyền mới có thể truy cập vào các chức năng quan trọng.
   * Sử dụng mã hóa để bảo vệ dữ liệu nhạy cảm và tuân thủ các quy định bảo mật dữ liệu như GDPR.

### **Kết Luận**

Java Swing là một lựa chọn khả thi và mạnh mẽ để xây dựng ứng dụng quản lý nhân sự với giao diện người dùng đồ họa. Với tính khả thi cao và khả năng mở rộng linh hoạt, ứng dụng có thể phát triển và đáp ứng các nhu cầu đa dạng của doanh nghiệp. Việc sử dụng Java Swing cùng với các công nghệ và kiến trúc hiện đại giúp đảm bảo rằng ứng dụng có thể duy trì và nâng cấp một cách dễ dàng trong tương lai.

# **TÀI LIỆU THAM KHẢO**

[1] Đoàn Văn Ban; *Lập trình hướng đối tượng với Java*; NXB khoa học và kỹ thuật, Hà Nội 2005

[2] Đoàn Văn Ban; *Lập trình Java nâng cao*; NXB khoa học và kỹ thuật, Hà Nội 2006

[3] Nguyễn Bá Nghiễn; *Lập trình java*; NXB thông tin và truyền thông 2018.