



**HUTECH**  
Đại học Công nghệ Tp.HCM

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**

\_oOo\_

# **ĐỒ ÁN CHUYÊN NGÀNH**

## **WEBSITE QUẢN LÝ VIỆC LÀM**

**Ngành: Công Nghệ Thông Tin**

**Chuyên ngành: Công Nghệ Phần Mềm**

**Giảng viên hướng dẫn: Bùi Mạnh Toàn**

**Nhóm:**

**Sinh viên thực hiện:**

**Chu Thanh Việt**

**MSSV: 2180601812**

**Lớp: 21DTHA4**

**TP. Hồ Chí Minh, 2024**

## **LỜI CẢM ƠN**

Để hoàn thành báo cáo, trước hết nhóm em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Bùi Mạnh Toàn người đã tận tình hướng dẫn em trong suốt quá trình nghiên cứu vừa qua. Trong thời gian được thầy hướng dẫn, em không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần, thái độ làm việc nghiêm túc, hiệu quả. Đây là điều rất cần thiết cho em trong quá trình học tập và làm việc sau này.

Xin kính chúc tất cả mọi người dồi dào sức khỏe và tràn đầy nhiệt huyết trong cuộc sống cũng như công việc của mình.

**SINH VIÊN THỰC HIỆN ĐỒ ÁN**

## MỤC LỤC

MỤC LỤC .....	3
DANH MỤC TỪ VIẾT TẮT .....	6
DANH MỤC HÌNH ẢNH .....	Error! Bookmark not defined.
LỜI NÓI ĐẦU .....	7
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI .....	1
1.1. Lý do chọn đề tài .....	1
1.2. Mục đích và mục tiêu của đề tài .....	1
1.3. Phương pháp tiếp cận .....	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG.....	4
2.1. Hệ quản trị cơ sở dữ liệu MySQL Workbench.....	4
2.2. Tổng quan về kiến trúc của Java Persistence API (JPA).....	5
2.3. Tổng quan về Draw.io .....	7
2.4. Spring boot .....	8
2.5. Tổng quan về RESTful API.....	9
2.6. Tổng quan về HTML, CSS VÀ JAVASCRIPT .....	13
2.7. Tổng quan về postman .....	14
2.8. Tổng quan về React.....	15
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....	17
3.1. Các bước hoàn thành dự án.....	17
3.2 Tổng quan hệ thống.....	18
3.3 Liệt kê các tác nhân và usecase.....	18
3.3.1 Các tác nhân.....	18
3.3.2 Danh sách Usecase.....	18
3.4 Biểu đồ Usecase .....	22
3.4.1 Biểu đồ usecase tổng quát .....	22
3.4.2 Biểu đồ usecase Quản lý công ty .....	23
3.4.3. Biểu đồ usecase Quản lý danh sách người dùng .....	23
3.4.3 Biểu đồ usecase Quản lý hồ sơ ứng viên.....	24
3.4.4 Biểu đồ usecase quản lý quyền và vai trò .....	25
3.4.5 Biểu đồ usecase xác thực và đăng nhập.....	26
3.5 Đặc tả Usecase .....	26

3.5.1 Usecase quản lý thông tin của một công ty .....	26
3.5.2 Usecase quản lý danh sách người dùng.....	26
3.5.3 Usecase quản lý hồ sơ ứng viên trong hệ thống.....	27
3.5.4 Usecase quản lý quyền hạn và vai trò của người dùng trong.....	27
<b>3.6 Biểu đồ tuần tự .....</b>	<b>28</b>
3.6.1. Biểu đồ tuần tự quản lý thông tin công ty.....	28
3.6.2. Biểu đồ tuần tự quản lý người dùng .....	29
3.6.2. Biểu đồ tuần tự quản lý quyền hạn và vai trò .....	30
<b>3.7. Biểu đồ hoạt động .....</b>	<b>32</b>
3.7.1. Xem thông tin của một công ty.....	32
3.7.2. Xóa thông tin của công ty.....	33
3.7.3. Sửa thông tin công ty.....	33
3.7.4. Thêm thông tin cho công ty .....	34
3.7.5. Xem thông tin của người dùng.....	35
3.7.6. Xóa thông tin người dùng .....	36
3.7.7. Thêm thông tin người dùng.....	36
3.7.8. Sửa thông tin người dùng .....	37
3.7.9. Xem thông tin các quyền hạn và vai trò có trong hệ thống.....	38
3.7.10. Thêm mới một quyền hạn và gán quyền hạn cho vai trò có trong hệ thống.....	39
3.7.11. Chỉnh sửa một quyền trong hệ thống.....	40
3.7.12. Xóa bỏ một quyền hạn có trong hệ thống.....	41
3.7.13. Thiết kế cơ sở dữ liệu.....	41
<b>CHƯƠNG 4: KẾT QUẢ CÀI ĐẶT CHƯƠNG TRÌNH .....</b>	<b>45</b>
4.1. Môi trường thực nghiệm và cài đặt chương trình .....	45
4.2. Giao diện và chức năng dành cho người dùng chưa đăng ký tài khoản...	45
4.3. Giao diện và chức năng khi người dùng đăng nhập .....	48
4.4. Giao diện và chức năng cho người quản trị viên của hệ thống .....	51
<b>CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>55</b>
5.1. Ưu điểm.....	55
5.2. Nhược điểm.....	55
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>57</b>



## DANH MỤC TỪ VIẾT TẮT

Thuật ngữ/ Từ viết tắt	Cụm từ đầy đủ	Ý nghĩa
API	Application Programming Interface	Giao diện lập trình ứng dụng
CPU	Central Processing Unit	Bộ xử lý trung tâm
DOM	Document Object Model	Mô hình các đối tượng trong tài liệu HTML
Framework	Framework	Framework là phần mềm được phát triển và sử dụng bởi các nhà phát triển để xây dựng các ứng dụng
GUI	Graphical User Interface	Giao diện đồ họa người dùng
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
IDE	Integrated Development Environment	Môi trường phát triển tích hợp
IE	Internet Explorer	Một trình duyệt web
Module	Module	Một thành phần phần mềm hoặc một phần của chương trình
QA	Quality Assurance	Những công việc nhằm đảm bảo chất lượng của quy trình phát triển một hệ thống phần mềm
SDK	Software Development Kit	Bộ công cụ phát triển phần mềm
Test Case	Test Case	Ca kiểm thử
Test Script	Test Script	Kịch bản kiểm thử

## LỜI NÓI ĐẦU

Trong thời đại công nghệ số ngày nay, internet không chỉ là công cụ kết nối mọi người mà còn trở thành một công cụ quan trọng trong việc hỗ trợ các hoạt động kinh tế, xã hội, và đặc biệt là quản lý nguồn nhân lực. Thị trường việc làm ngày nay phát triển mạnh mẽ, nhu cầu kết nối giữa người tìm việc và nhà tuyển dụng ngày càng cao. Tuy nhiên, việc quản lý, tìm kiếm và tuyển dụng qua các phương pháp truyền thống gặp phải nhiều khó khăn và rất tốn kém thời gian, đôi khi không đạt được kết quả như mong đợi.

Nhận thấy nhu cầu cấp thiết đó, nhóm em đã chọn đề tài “Website Quản Lý Việc Làm” cho đồ án lần này. Đề tài ra đời với mục đích xây dựng một website trực tuyến hỗ trợ nhà tuyển dụng và người tìm việc dễ dàng tiếp cận với nhau, người tìm việc dễ dàng tìm kiếm công việc, còn nhà tuyển dụng sẽ dễ dàng quản lý công việc một cách hiệu quả hơn. Website sẽ cung cấp các chức năng quan trọng như đăng tin tuyển dụng, tìm kiếm việc làm, quản lý hồ sơ ứng viên,...

Chúng em hy vọng rằng sản phẩm lần này không những đáp ứng được yêu cầu của đồ án mà còn có tiềm năng phát triển và được sử dụng trong thực tế, góp phần giải quyết bài toán quản lý việc làm hiện nay.

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Lý do chọn đề tài

Trong bối cảnh kinh tế khó khăn hiện nay, cùng với thị trường việc làm cạnh tranh khốc liệt, nhu cầu tìm kiếm việc làm và tuyển dụng ngày càng trở nên cấp thiết. Do đó sự bùng nổ mạnh mẽ của công nghệ đã tạo ra cơ hội lớn để chúng ta có thể dễ dàng tiếp cận với nhiều cơ hội việc làm hơn. Tuy nhiên, các phương pháp tìm kiếm và tuyển dụng truyền thống hiện nay chưa đáp ứng đầy đủ các yêu cầu từ các bên tham gia, từ người tìm việc cho đến nhà tuyển dụng, dẫn đến tình trạng là thông tin đăng tuyển không đầy đủ, gây khó khăn trong việc tìm kiếm và lựa chọn ứng viên phù hợp cho vị trí cần tuyển dụng.

Một trong những vấn nạn nan giải mà thị trường lao động đang gặp phải là sự kết nối thiếu hiệu quả giữa người tìm việc và nhà tuyển dụng. Việc các công ty tuyển dụng không thể tiếp cận được nguồn nhân lực chất lượng hoặc người tìm việc không tìm được một công việc phù hợp.

Vì vậy chúng em chọn đề tài “Website Quản Lý Việc Làm” với mong muốn xây dựng một nền tảng trực tuyến nơi người tìm việc và nhà tuyển dụng có thể tìm thấy nhau. Website sẽ cho cả hai phía là người tìm việc có thể dễ dàng tìm kiếm và ứng tuyển vào các công việc phù hợp, trong khi đó nhà tuyển dụng có thể dễ dàng quản lý hồ sơ mà ứng viên đăng tải.

## 1.2. Mục đích và mục tiêu của đề tài

- **Mục đích của đề tài**

- Tạo lập một website kết nối giữa người tìm việc và nhà tuyển dụng. Xây dựng một trang web trực tuyến giúp người tìm việc có thể dễ dàng tìm kiếm công việc phù hợp với bản thân và nhà tuyển dụng có thể đăng tải thông tin tuyển dụng với mong muốn tìm kiếm ứng viên phù hợp.
- Website giúp nhà tuyển dụng quản lý hồ sơ ứng viên một cách dễ dàng, theo dõi quá trình tuyển dụng và lựa chọn ứng viên phù hợp với vị trí một cách nhanh chóng và chính xác tránh lãng phí nguồn lực.
- Website cung cấp thông tin tuyển dụng rõ ràng, giúp người tìm việc nắm rõ yêu cầu và mô tả công việc, đồng thời giúp nhà tuyển dụng tiếp cận được các ứng viên tiềm năng.



- **Mục tiêu của đề tài**

- Giúp người tìm việc có thể nhanh chóng tìm được công việc phù hợp với nhu cầu và kỹ năng của bản thân.
- Xây dựng giao diện website thân thiện gần gũi, dễ dàng sử dụng và tối ưu hóa cho cả người tìm việc và nhà tuyển dụng, giúp quá trình tuyển dụng trở nên đơn giản hóa và tiện lợi hơn.

- **Phạm vi chức năng**

Website xây dựng với nhiều chức năng hợp lý, dễ dàng sử dụng nhằm đem lại cho người sử dụng sự thuận tiện trong các thao tác truy cập và tìm kiếm thông tin.

- Đăng ký và đăng nhập người dùng: Cho phép người tìm việc và nhà tuyển dụng tạo tài khoản, đăng nhập để sử dụng các tính năng của website.
- Tìm kiếm việc làm: Cung cấp công cụ tìm kiếm việc làm thông minh, cho phép người tìm việc tìm kiếm các công việc theo các tiêu chí như địa điểm, kỹ năng.
- Đăng tin tuyển dụng: Nhà tuyển dụng có thể dễ dàng đăng tải các thông tin tuyển dụng, mô tả công việc, yêu cầu ứng viên và các thông tin liên quan.
- Rải CV: người tìm việc có thể dễ dàng rải cv vào các công việc phù hợp với nhu cầu của bản thân.

- **Phạm vi dữ liệu**

Nguồn dữ liệu dùng cho hệ thống có thể được cập nhật từ nhiều nguồn khác nhau, tuy nhiên trong điều kiện có thể và phạm vi của đề tài thì phạm vi dữ liệu của website quản lý việc làm:

- Tin tuyển dụng được đăng từ các công ty.
- Tin đăng từ ban quản trị hệ thống.
- Thông tin từ các website khác.

- **Môi trường phát triển**

- ❖ **Môi trường phát triển**

- CPU: Intel Core I5 hoặc cao hơn

- RAM: Tối thiểu 16GB
- SSD: Dung lượng trống tối thiểu 20GB
- ❖ Môi trường phần mềm
  - Hệ điều hành: Windows 10
  - Java spring version 3.0 trở lên
  - Công cụ phát triển: Visual Studio Code

### **1.3. Phương pháp tiếp cận**

Cách tiếp cận: Nghiên cứu các web trên các nền tảng công nghệ từ trước. Sử dụng các phương pháp nghiên cứu:

- Phương pháp đọc tài liệu.
- Phương pháp phân tích mẫu.
- Phương pháp sử dụng phần mềm và mô phỏng
- Phương pháp thực nghiệm làm những module nhỏ.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

### 2.1. Hệ quản trị cơ sở dữ liệu MySQL Workbench

#### **Giới Thiệu MySQL Workbench[1]:**

MySQL Workbench là một công cụ đồ họa mạnh mẽ được phát triển bởi Oracle, được thiết kế để làm việc với các máy chủ và cơ sở dữ liệu MySQL. Nó được biết đến như một phần mềm đa nền tảng, cho phép MySQL chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS. MySQL Workbench hỗ trợ đầy đủ cho MySQL Server 8.0. MySQL Workbench cung cấp một giao diện dễ sử dụng cho phép người dùng thực hiện các thao tác như quản lý kết nối, viết và thực thi truy vấn SQL, thiết kế cơ sở dữ liệu, cũng như giám sát và quản lý máy chủ MySQL.

#### **Các tính năng chính của MySQL Workbench bao gồm:**

**SQL Development:** Cung cấp công cụ để tạo và quản lý kết nối tới các máy chủ cơ sở dữ liệu MySQL. MySQL Workbench cho phép người dùng cấu hình các tham số kết nối và thực thi các truy vấn SQL thông qua trình chỉnh sửa SQL tích hợp sẵn.

**Data Modeling (Thiết kế cơ sở dữ liệu):** Cung cấp khả năng tạo mô hình cơ sở dữ liệu đồ họa, hỗ trợ việc đảo ngược và chuyển tiếp giữa mô hình cơ sở dữ liệu và cơ sở dữ liệu thực tế. Công cụ Table Editor của MySQL Workbench giúp chỉnh sửa các bảng, cột, chỉ mục, triggers, phân vùng, quyền hạn và các đối tượng cơ sở dữ liệu khác một cách dễ dàng.

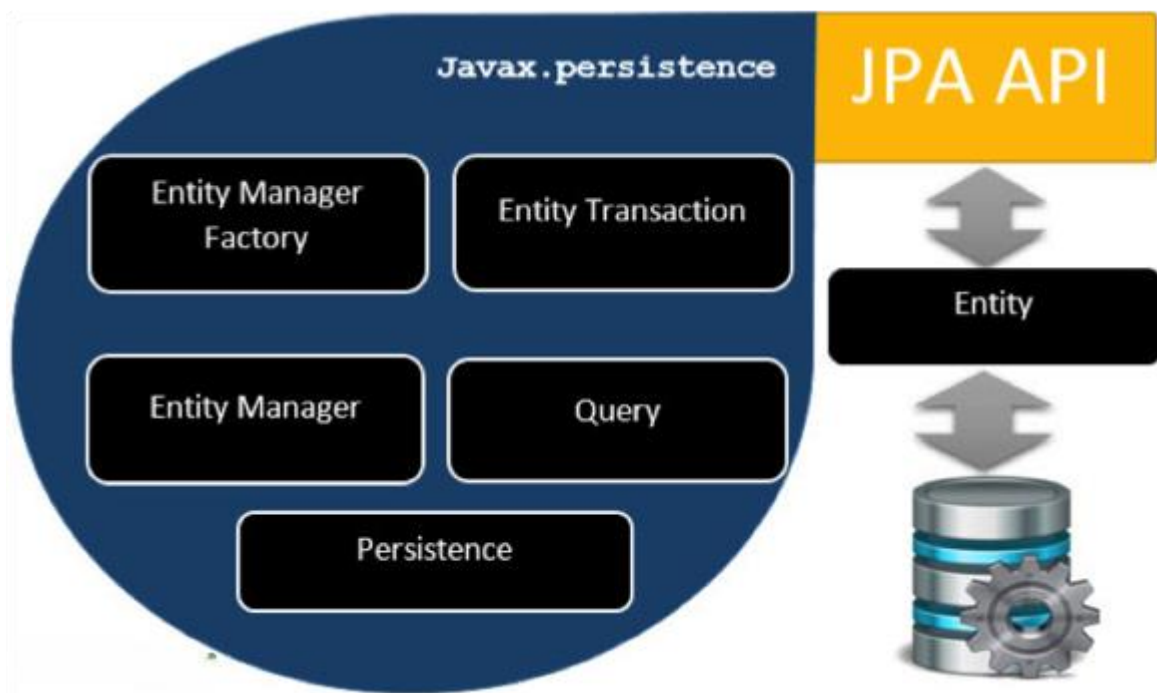
**Server Administration (Quản trị máy chủ):** Hỗ trợ quản trị các phiên bản MySQL thông qua việc quản lý người dùng, thực hiện sao lưu và khôi phục, kiểm tra dữ liệu audit, theo dõi sức khỏe của cơ sở dữ liệu và giám sát hiệu suất của máy chủ MySQL.

**Data Migration(di chuyển dữ liệu):** cho phép di chuyển dữ liệu từ các hệ quản trị cơ sở dữ liệu khác như Microsoft SQL Server, Microsoft access, Sybase ASE, SQLite, Postgresql,... và nhiều hệ quản trị cơ sở dữ liệu khác sang MySQL.

**MySQL Enterprise Support (Hỗ trợ doanh nghiệp MySQL):** Cung cấp các tính năng hỗ trợ cho các sản phẩm doanh nghiệp như MySQL Enterprise Backup, MySQL Firewall và MySQL audit.

## 2.2. Tổng quan về kiến trúc của Java Persistence API (JPA)

**Java Persistence API** là một nền tảng được sử dụng để làm việc với database thông qua cơ chế ánh xạ Object/Relational Mapping (ORM). Nhờ đó, có thể truy vấn, thao tác với database gián tiếp thông qua các đối tượng lập trình. Kiến trúc của Java Persistence API được minh họa như hình 2.2 dưới đây:



*Hình 2.2: Kiến trúc của Java Persistence API*

### **Entity**

Entity là các đối tượng đại diện cho một bảng (table) trong cơ sở dữ liệu. Thông thường, Entity được triển khai dưới dạng các lớp POJO (Plain Old Java Object), bao gồm các phương thức getter và setter đơn giản để truy cập và thay đổi giá trị của các thuộc tính.

Một số đặc điểm của Entity bao gồm:

- Entity có khả năng tương tác trực tiếp với cơ sở dữ liệu quan hệ.
- Mỗi Entity được xác định duy nhất bằng một định danh (ID), tương đương với khóa chính (primary key) trong bảng cơ sở dữ liệu.

- Entity hỗ trợ các giao dịch (transaction), giúp đảm bảo tính toàn vẹn dữ liệu trong quá trình thao tác.
- Giống như các lớp Java khác, Entity cũng có thể kế thừa từ các lớp cha.

### **EntityManager**

Entity Manager là một interface cung cấp các phương thức để thao tác với các Entity trong cơ sở dữ liệu. Nó đóng vai trò như là cầu nối giữa ứng dụng và cơ sở dữ liệu, giúp thực hiện các thao tác thêm sửa xóa dữ liệu.

Một số chức năng chính của EntityManager bao gồm:

- Persist: Dùng để lưu một thực thể (entity) mới vào cơ sở dữ liệu.
- Merge: Cập nhật các thay đổi của một thực thể vào cơ sở dữ liệu.
- Remove: Xóa một thực thể khỏi cơ sở dữ liệu.

### **Entity Manager Factory**

- EntityManagerFactory được dùng để tạo ra một đối tượng EntityManager

### **Persistence**

Một persistence định nghĩa một tập hợp các lớp Entity được quản lý bởi một đối tượng EntityManager trong ứng dụng

Lớp Persistence cung cấp các phương thức tĩnh(static) để lấy đối tượng EntityManagerFactory

### **EntityTransaction**

Một Transaction là một nhóm các thao tác, trong đó tất cả các thao tác phải được thực hiện thành công hoặc tất cả phải thất bại.

Một database transaction bao gồm một loạt các câu lệnh SQL, và các câu lệnh này sẽ được commit (xác nhận) hoặc rollback (hoàn tác) trong một đơn vị giao dịch.

EntityTransaction có mối quan hệ 1-1 với EntityManager. Mọi thao tác được khởi tạo qua EntityManager đều sẽ được thực hiện trong khuôn khổ một Transaction. EntityManager chịu trách nhiệm tạo ra đối tượng EntityTransaction.

## Query

Query là một interface mà mỗi nhà cung cấp JPA triển khai để tạo ra các đối tượng truy vấn, từ đó trả về kết quả phù hợp với các tiêu chí (criteria) đã được xác định.

### 2.3. Tổng quan về Draw.io

Draw.io là môi trường mô hình hóa tổng thể dưới dạng đồ họa và dễ dàng sử dụng. Công dụng Draw.io cung cấp [1]:

- Trình bày các mô hình bằng các ký hiệu đồ họa phù hợp với các phương pháp mô hình hóa trong thực tế như mô hình thực thể và mối kết hợp, mô hình quan hệ.
- Tự động tạo hồ sơ mô tả các đối tượng trên mô hình.
- Tự động tạo mã phát sinh Cơ Sở Dữ liệu và các chức năng xử lý từ mô hình đã xây dựng.

Vai trò của Draw.io việc thiết kế mô hình cơ sở dữ liệu:

- Trình bày mô hình ở dạng đồ họa.
- Kiểm tra tính hợp lệ của mô hình được thiết kế
- Phát sinh mô hình dữ liệu logic hay mô hình dữ liệu vật lý.

Sơ lược về thực thể, thuộc tính, liên kết

#### Thực thể

- Thực thể là khái niệm để chỉ một lớp các đối tượng có cùng đặc tính chung mà người ta muốn quản lý thông tin về nó. Ví dụ, sinh viên, hàng hóa, vật tư.

- Một đối tượng cụ thể trong thực thể được gọi là một cá thể (còn gọi là một thể hiện của thực thể). Ví dụ Lê Xuân Hưng là một cá thể của thực thể Sinh viên. Bánh kẹo là các thể của thực thể hàng hóa.

#### Thuộc tính thực thể

- Để mô tả thông tin về một thực thể người ta thường dựa vào các đặc trưng riêng của thực thể đó, Các đặc trưng đó được gọi là thuộc tính của thực thể. Ví dụ thực thể Sinh viên có các thuộc tính Mã sinh viên, Họ tên sinh viên, Ngày sinh, Địa chỉ, Trường, Khoa, Khóa sinh viên, Lớp.

## **Liên kết**

- Một liên kết là một sự ghép nối giữa hai hay nhiều thực thể. Phản ánh sự liên hệ giữa các thực thể Có 3 kiểu liên kết: liên kết một một (1-1), quan hệ nhiều nhiều (nn) và quan hệ một nhiều (1-n).
- Quan hệ 1-1: Hai thực thể A và B có mối quan hệ 1-1 nếu một thực thể kiểu A tương ứng với một thực thể kiểu B và ngược lại.
- Quan hệ 1-n: Hai thực thể A và B có mối quan hệ 1- n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và một thực thể của B chỉ tương ứng với một thực thể kiểu A.
- Quan hệ n-n: Hai thực thể A và B có mối quan hệ n-n nếu một thực thể kiểu A tương ứng với nhiều thực thể kiểu B và ngược lại.

## **2.4. Spring boot**

### **Giới thiệu về Spring boot.**

Spring boot là một framework được phát triển dựa trên Spring Framework với mục tiêu đơn giản hóa quá trình phát triển và cấu hình các ứng dụng Java. Được ra mắt lần đầu vào tháng 4 năm 2014, Spring Boot nhanh chóng trở thành sự lựa chọn hàng đầu của các lập trình viên trong sinh thái Java nhờ khả năng giảm thiểu quá trình cấu hình phức tạp và tối ưu hóa hiệu suất khi phát triển phần mềm.

Lý do tại sao Spring boot được ra đời ?

Trong những ngày đầu phát triển ứng dụng Java, việc cấu hình môi trường phát triển là một quy trình phức tạp, tốn thời gian và dễ gây ra những lỗi vặt. Mặc dù Spring Framework đã giúp đơn giản hóa phần nào với các khái niệm như inversion of Control (IOC) và Dependency injection (DI) nhưng các lập trình viên vẫn phải cấu hình rất nhiều thứ thông qua các tệp XML phức tạp.

Vì vậy spring boot ra đời là để thay đổi những khó khăn mà các lập trình viên đang gặp phải. Nó cung cấp các công cụ và thành phần được cấu hình sẵn, giúp đời sống lập trình của các lập trình viên nhàn hơn, phát triển và xây dựng ứng dụng với java một cách nhanh chóng và thuận tiện.

### **Ưu điểm tuyệt vời của Spring boot:**

- Spring Boot tự động cấu hình các thành phần cần thiết cho ứng dụng dựa trên các thư viện được sử dụng, giảm đi công sức và thời gian cấu để cấu hình thủ công.
- Cung cấp các gói phụ thuộc được cấu hình sẵn cho các loại ứng dụng khác nhau, như spring-boot-starter-web cho ứng dụng web hoặc spring-boot-starter-data-jpa sử dụng để tương tác với cơ sở dữ liệu
- Tích hợp sẵn máy chủ web Tomcat, Jetty và Undertow cho phép ứng dụng chạy hiệu quả hơn
- Spring Boot giúp các lập trình viên triển khai ứng dụng một cách nhanh chóng, nó phù hợp cho cả các dự án vừa và nhỏ lẫn các hệ thống lớn phức tạp.
- Các ứng dụng Spring Boot dễ dàng được mở rộng, bảo trì và có thể tích hợp với các dịch vụ, thư viện và công nghệ hiện đại.
- Giảm thiểu đi khó khăn khi tiếp cận ứng dụng cho lập các trình viên mới bằng cách cung cấp các công cụ đơn giản, dễ hiểu.

## **2.5. Tổng quan về RESTful API**

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy nhiên chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau. Nó cũng giống như bàn phím là thiết bị dùng để giao tiếp giữa người sử dụng và máy tính, API là một phần mềm giao tiếp giữa chương trình và hệ điều hành.

Những điểm nổi bật của API:

- Đây là một trong những framework mới sẽ giúp ích cho bạn trong việc xây dựng các HTTP service một cách rất đơn giản và nhanh chóng.
- Mã nguồn mở nên bạn có thể được sử dụng bởi bất kỳ một client nào hỗ trợ XML, JSON.
- Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma.
- Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS.
- Một kiểu kiến trúc vô cùng phù hợp dành cho các thiết bị trang bị bằng thông giới hạn như smartphone, tablet.
- Thường nó có định dạng dữ liệu là JSON, XML hoặc một kiểu dữ liệu bất kỳ.

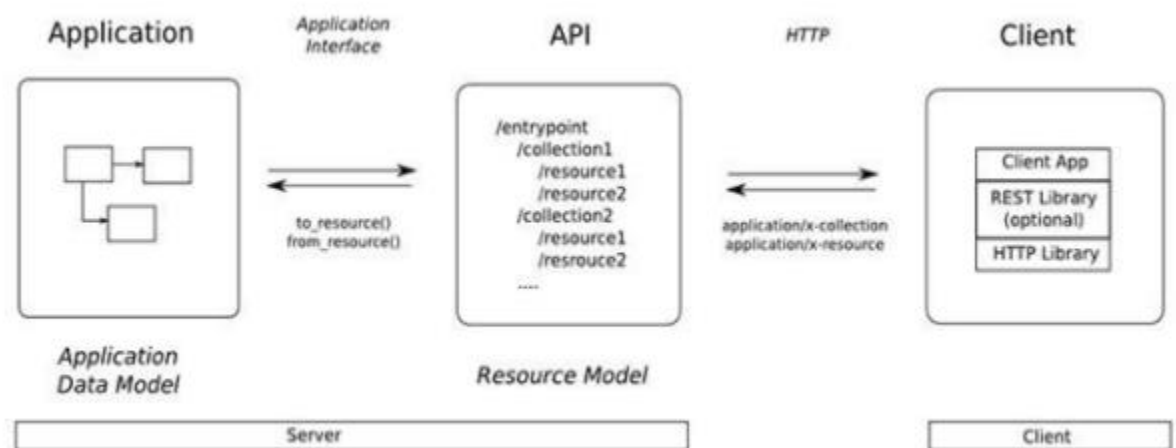


## REST For Web Developers

- Hiện nguyên lý REST và cấu trúc dữ liệu RESTful được biết đến rất rộng rãi trong giới lập trình, nhưng vẫn có nhiều người hay nhầm lẫn. Bản thân REST không phải là một loại công nghệ, mà là phương thức tạo API với nguyên lý tổ chức nhất định. Những nguyên lý này nhằm hướng dẫn lập trình viên tạo môi trường xử lý API request được toàn diện hơn.
- REST là viết tắt của Representational State Transfer. Giải thích đơn giản, REST là một loạt hướng dẫn và dạng cấu trúc dùng cho việc chuyển đổi dữ liệu. Thông thường, REST hay được dùng cho ứng dụng web, nhưng cũng có thể làm việc được với dữ liệu phân mềm.
- Tóm lại, RESTful API là các API theo cấu trúc REST.

### Đặc điểm

- Sự nhất quán trong cả API
- Tồn tại không trạng thái (ví dụ, không có server-side session)
- Sử dụng HTTP status code khi cần thiết
- Sử dụng URL endpoint với logical hierarchy



Hình 0.1 Cấu trúc REST

- Versioning trong URL chứ không phải trong HTTP header

REST là một phương thức nhỏ gọn. Nên rất được ưa chuộng cho dữ liệu HTTP. Cũng vì vậy nên REST dần phổ biến trên web, và được xem là lựa chọn “số một” cho phát triển API.

Có 4 lệnh dùng để truy cập RESTful API:

- GET để truy vấn object
- POST để tạo object mới
- PUT để sửa đổi hoặc thay thế một object
- DELETE để loại bỏ một object

Mỗi phương thức trên phải được API call thông qua để gửi chỉ thị cho server phải làm gì.

Sự khác nhau giữa Web Service, WCF, WCF REST, Web API

Hiện nay trong kỷ nguyên hậu PC, smartphone và tablet lên ngôi, nhu cầu xây dựng các ứng dụng mobile trên Windows Phone, iOS, Android đồng bộ dữ liệu đến server theo mô hình điện toán đám mây đang trở nên rất phổ biến. Nếu chọn Spring Boot để phát triển phía server, các lập trình viên có thể dễ dàng xây dựng các HTTP service và RESTful APIs mạnh mẽ, có khả năng phục vụ cho nhiều loại client khác nhau (bao gồm ứng dụng di động và web). Spring Boot cung cấp một nền tảng đơn giản, hiệu quả để xây dựng các dịch vụ web, hỗ trợ giao tiếp dữ liệu qua JSON, XML hoặc các định dạng dữ liệu khác, đồng thời tối ưu hóa hiệu suất và bảo mật cho ứng dụng.

Spring Boot Web API Service

- Spring Boot là một framework mạnh mẽ trong hệ sinh thái của Spring, giúp xây dựng các ứng dụng web và RESTful API dễ dàng và nhanh chóng.
- Spring Boot hỗ trợ giao thức HTTP và có dịch vụ RESTful, có thể trả về dữ liệu dưới dạng JSON hoặc XML, phù hợp cho các ứng dụng web hiện nay và cả mobile.
- Được tích hợp sẵn với các công cụ hỗ trợ REST như Spring Web và Spring Data, giảm đi thời gian và công việc cấu hình.
- Spring Boot có thể được triển khai và chạy trên các môi trường như Tomcat, Jetty, hoặc Undertow.

**Ưu điểm của Spring Boot Web Service:**

- Cấu hình đơn giản: Các dịch vụ RESTful có thể được triển khai một cách nhanh chóng mà không cần phải lo lắng về cấu hình phức tạp.

- Hỗ trợ JSON và XML: Có thể trả về dữ liệu trong nhiều định dạng khác nhau (JSON, XML), thích hợp cho các ứng dụng di động và web hiện nay.
- Tính mở rộng và dễ bảo trì: Spring boot hỗ trợ các tính năng như dependency injection, unit testing và IOC container, giúp cho việc mở rộng và bảo trì ứng dụng trở nên dễ dàng hơn.
- Tính tương thích cao: Có thể hoạt động với bất kỳ client nào hỗ trợ HTTP, bao gồm các trình duyệt web, ứng dụng di động, và các thiết bị khác.
- Spring Security: Đảm bảo an toàn bảo mật cho các ứng dụng web với các tính năng như authentication và authorization.
- Performance cao: Spring Boot tối ưu hóa hiệu suất của các dịch vụ RESTful và dễ dàng scale lên khi cần.

#### **Nhược điểm của Spring Boot Web Service:**

- Đôi khi khó cấu hình: Mặc dù Spring Boot có cấu hình tự động (auto-configuration), nhưng trong một số trường hợp, việc cấu hình có thể đòi hỏi lập trình viên có kiến thức sâu hơn về Spring.
- Đối với những người mới bắt đầu, việc tiếp xúc và làm quen với Spring Boot với các tính năng của nó phải dành rất nhiều thời gian nghiên cứu và học tập.

#### **Tại sao nên sử dụng Spring Boot Web API:**

- RESTful Web Services: Spring Boot hỗ trợ đầy đủ các tính năng của REST
- Đáp ứng nhiều loại client: Spring Boot có khả năng truyền dữ liệu cho các client khác nhau, bao gồm trình duyệt web, ứng dụng di động, máy tính bảng và các thiết bị IoT.
- Mở rộng dễ dàng: Spring Boot hỗ trợ việc phát triển các dịch vụ dễ dàng mở rộng khi ứng dụng phát triển lớn hơn theo thời gian.
- Cộng đồng hỗ trợ mạnh mẽ: Là một phần của hệ sinh thái Spring, Spring Boot nhận được sự hỗ trợ từ một cộng đồng lớn mạnh và tài liệu phong phú, giúp việc phát triển ứng dụng trở nên dễ dàng hơn.

## 2.6. Tổng quan về HTML, CSS VÀ JAVASCRIPT

- **HTML (Hypertext Markup Language) [2]**

HTML là chữ viết tắt của cụm từ HyperText Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML. Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

- **CSS (Cascading Style Sheets)**

CSS là từ viết tắt của Cascading Style Sheets: công dụng dùng để trang trí trang web và thường được gắn với các ngôn ngữ như HTML, PHP dùng làm nổi bật trang web và hình ảnh trang web. Trong Style Sheet này chứa những câu lệnh CSS. Mỗi câu lệnh CSS sẽ định dạng một phần nhất định của HTML, ví dụ như: font của chữ, đường viền, màu nền, căn chỉnh hình ảnh,...

- **JAVASCRIPT**

JavaScript là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu được tạo ra vào năm 1995 bởi Brendan Eich tại Netscape. Mục đích ban đầu để nâng cao khả năng tương tác của trang web với người sử dụng.

JavaScript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, thậm chí các trình duyệt trên thiết bị di động cũng hỗ trợ.

JavaScript là một ngôn ngữ lập trình dựa trên nguyên mẫu với cú pháp phát triển từ C. JavaScript có khái niệm từ khóa, do đó gần như JavaScript không được mở rộng. Cũng giống như C, JavaScript không có bộ xử lý xuất/nhập riêng. Trong khi C sử dụng thư viện xuất/nhập chuẩn, JavaScript dựa vào phần mềm ngôn ngữ được gắn vào để thực hiện xuất/nhập.

Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh

## 2.7. Tổng quan về Postman

### **Giới thiệu về Postman:**

Postman là một công cụ mạnh mẽ và rất phổ biến trong lĩnh vực phát triển và quản lý API (Application Programming Interfaces). Nó giúp đơn giản hóa quá trình tạo, kiểm thử, tài liệu hóa, mô phỏng, chia sẻ và giám sát các API, mang lại sự hiệu quả và tiện lợi cho các lập trình viên. Postman hoạt động dưới dạng ứng dụng desktop hoặc website trực tuyến, với giao diện thân thiện, dễ sử dụng và dễ làm quen.

### **Tại sao Postman lại quan trọng trong việc phát triển ứng dụng ?**

Trong thời đại của dữ liệu lên ngôi, các API đóng vai trò là cầu nối giữa các ứng dụng và các nguồn dữ liệu như mạng xã hội, dịch vụ điện toán đám mây và cơ sở dữ liệu. Postman không chỉ giúp các nhà phát triển kiểm tra, theo dõi và quản lý các API mà còn hỗ trợ việc tích hợp các công cụ mạnh mẽ như OpenAPI, GraphQL, SOAP và REST, chúng giúp cải thiện hiệu suất và chất lượng trong công việc. Postman là một công cụ tuyệt vời cho các đội nhóm phát triển làm việc với các dự án lớn, phức tạp, yêu cầu sự phối hợp chặt chẽ và quản lý giữa các phiên bản.

### **Các tính năng chính của Postman**

#### **API Repository:**

- Cung cấp không gian lưu trữ cho thông tin liên quan đến API, giúp dễ dàng tổ chức, quản lý cũng như kiểm soát các phiên bản API.
- Hỗ trợ các công cụ thiết kế API như OpenAPI, GraphQL, và RAML, tích hợp với các công cụ quản lý mã nguồn như Git để tự động hóa tài liệu và kiểm tra API.
- Cho phép chia sẻ API trong nội bộ hoặc công khai, đồng thời cung cấp công cụ tìm kiếm mạnh mẽ để dễ dàng truy cập thông tin.

#### **API Tools:**

- Hỗ trợ các giao thức như HTTP, REST, SOAP, GraphQL và WebSockets, với khả năng tự động phân tích phản hồi và định dạng dữ liệu.
- Thiết kế API: Hỗ trợ các định dạng như OpenAPI, GraphQL, RAML, và SOAP, giúp dễ dàng tạo và xác thực các đặc tả kỹ thuật của API.
- Tài liệu tự động: Tự động tạo tài liệu từ file OpenAPI, bao gồm mã mẫu ở nhiều ngôn ngữ khác nhau.

#### **API Governance:**

- Cung cấp các quy tắc thiết kế API để đảm bảo chất lượng và tính nhất quán trong toàn bộ vòng đời API.
- Hỗ trợ giám sát bảo mật, kiểm tra token truy cập và cung cấp bảng điều khiển phân tích trạng thái của API, giúp cải thiện quản lý vận hành và bảo mật.

### **Postman Integration:**

- Tích hợp các công cụ phát triển quan trọng như là AWS API Gateway, Github và Jenkins, giúp đơn giản hóa quy trình tạo và kiểm thử.

Postman là một công cụ không thể thiếu cho các nhà phát triển, Postman cung cấp các tính năng mạnh mẽ giúp tăng năng suất làm việc đáng kể, cải thiện hiệu quả và đảm bảo chất lượng trong mọi giai đoạn của vòng đời phát triển API. Với sự hỗ trợ đa dạng về giao thức và tích hợp, Postman đang đáp ứng đầy đủ các nhu cầu của cả các dự án nhỏ lẻ và các hệ thống lớn phức tạp.

## **2.8. Tổng quan về React**

### **Giới thiệu về React:**

React là một thư viện nổi tiếng dùng để xây dựng giao diện người dùng (UI). Được phát triển bởi Facebook, React cho phép tạo ra các thành phần UI có thể tái sử dụng và giúp quản lý trạng thái của ứng dụng một cách hiệu quả. Một trong những ưu điểm nổi bật của React là khả năng hoạt động không chỉ trên phía client mà còn có thể được render trên server, cho phép sự kết hợp linh hoạt giữa hai bên. React thực hiện so sánh sự khác biệt giữa giá trị hiện tại và giá trị trước đó trong quá trình render, từ đó chỉ cập nhật những thay đổi cần thiết nhất trên DOM.

### **Ưu điểm của ReactJS**

ReactJS có nhiều ưu điểm nổi bật, giúp nó trở thành lựa chọn phổ biến trong việc phát triển ứng dụng web hiện đại:

- **Hiệu quả cao:** ReactJS sử dụng Virtual DOM để cập nhật và hiển thị nhanh chóng giao diện người dùng. Khi có thay đổi xảy ra, React không hiển thị toàn bộ trang web mà chỉ cập nhật phần đã thay đổi của Virtual DOM. Điều này giúp cải thiện hiệu suất và tăng tốc độ tải trang, mang lại trải nghiệm mượt mà hơn cho người dùng.
- **Thành phần tái sử dụng:** ReactJS cho phép tạo các components có thể được tái sử dụng nhiều lần trong ứng dụng. Mỗi thành phần có thể được phát triển và kiểm thử một cách độc lập, cải thiện tính dễ bảo trì và tính nhất quán cho dự án. Các thành phần này cũng có thể được kết hợp với nhau để tạo ra các giao diện phức tạp.
- **Cộng đồng và hệ sinh thái mạnh mẽ:** ReactJS có một cộng đồng phát triển to lớn và hoạt động rất tích cực, góp phần cải tiến không ngừng của thư viện. Có rất nhiều tài liệu hướng dẫn, công cụ hỗ trợ trong cộng đồng giúp lập trình viên tìm kiếm thông tin và giải quyết vấn đề một cách dễ dàng.
- **Thân thiện với SEO:** hỗ trợ render phía server (server-side rendering), giúp các ứng dụng web dễ dàng được tìm thấy và lập chỉ mục các ứng dụng web. Nội dung có thể được render trước thành HTML, đảm bảo rằng các công cụ tìm kiếm có thể index hiệu quả. React cũng cho phép quản lý

metadata quan trọng cho SEO, tích hợp dữ liệu có cấu trúc và duy trì cấu trúc URL thân thiện.

### **Nhược điểm của ReactJS.**

- Độ phức tạp: ReactJS có thể khá phức tạp, đặc biệt là đối với người mới bắt đầu học. Việc học React đòi hỏi phải nắm vững nhiều khái niệm và hiểu rõ hệ sinh thái xung quanh của React.
- Thiếu tính nhất quán: ReactJS là một thư viện thay vì là một framework, điều này có nghĩa là nó sẽ cung cấp ít cấu trúc và mang đến nhiều sự linh hoạt hơn trong phát triển ứng dụng. Điều này cho phép các lập trình viên tự do lựa chọn cách cấu trúc dự án, quản lý state và xử lý routing, nhưng cũng dẫn đến cách thực thi code không theo tiêu chuẩn giữa các đội nhóm phát triển.
- React về mặt lưu trữ khá nặng nếu so với các framework khác như Angular. Trong khi đó Angular là một Framework hoàn chỉnh.

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1. Các bước hoàn thành dự án

*Bảng 3.1: Các bước hoàn thành dự án*

Tiến trình	Mô tả
Khởi tạo dự án	<p>Tìm hiểu các tài liệu liên quan đến dự án.</p> <p>Tìm hiểu cách thức hoạt động của kinh doanh và liên hệ đặt tour du lịch.</p> <p>Xem xét tính khả thi của dự án, các khía cạnh liên quan đến dự án.</p>
Lập kế hoạch	<p>Mô tả tổng quát quá trình thực hiện dự án.</p> <p>Ràng buộc dự án với các điều kiện chủ quan và khách quan. Xác định tính khả thi của dự án.</p> <p>Xác định rủi ro và các vấn đề phát sinh trong quá trình xây dựng dự án có thể gặp.</p> <p>Lập kế hoạch hệ thống mới.</p>
Thu thập yêu cầu	<p>Thu thập yêu cầu của khách hàng đối với hệ thống một cách chi tiết và chính xác.</p> <p>Khảo sát, tìm hiểu các hệ thống đã có thể đưa ra các yêu cầu, các chức năng cho hệ thống của mình.</p>
Phân tích và kiểm tra yêu cầu	<p>Phân tích các yêu cầu chức năng và phi chức năng.</p> <p>Kiểm tra tính khả thi của yêu cầu.</p> <p>Bổ sung các yêu cầu còn thiếu sót.</p>
Phân tích	<p>Mô hình hóa các yêu cầu hệ thống và đặc tả các chức năng.</p>
Thiết kế	<p>Thiết kế cơ sở dữ liệu. Thiết kế giao diện.</p>
Kiểm thử	<p>Kiểm thử cho từng giai đoạn của dự án từ lúc bắt đầu đến lúc kết thúc dự án.</p>
Cài đặt	<p>Xây dựng website cho kế hoạch đã đặt ra.</p> <p>Viết tài liệu hướng dẫn sử dụng hệ thống người dùng.</p>
Thực thi	<p>Đưa hệ thống vào hoạt động thử. Theo dõi hoạt động của hệ thống.</p>



## 3.2 Tổng quan hệ thống

Website quản lý việc làm là một ứng dụng web hiện đại nhằm kết nối các nhà tuyển dụng với người tìm việc một cách hiệu quả trong quản lý, đăng tải, tìm kiếm và hiển thị thông tin chi tiết của các công ty. Hệ thống được xây dựng trên tiêu chí dễ tiếp cận, dễ sử dụng, thông tin chi tiết, tìm kiếm nhanh gọn, trực quan, dễ quản lý và mức độ bảo mật cao.

## 3.3 Liệt kê các tác nhân và usecase

### 3.3.1 Các tác nhân

- Nhà tuyển dụng: Nhà tuyển dụng có thể quản lý thông tin công ty, đăng và quản lý các công việc cần tuyển dụng.
- Người tìm việc: Người tìm việc có thể quản lý hồ sơ, thông tin cá nhân và tài khoản, đăng ký tài khoản mới và đăng nhập vào hệ thống.
- Quản trị viên: Quản trị viên có thể quản lý các quyền truy cập, vai trò của người dùng, quản lý thông tin tài khoản của người dùng và quản lý các tệp tin trong hệ thống.
- Khách truy cập: Khách truy cập có thể tìm kiếm và xem thông tin các công việc, công ty và kỹ năng có trên hệ thống mà không cần đăng nhập.

### 3.3.2 Danh sách Usecase

- Quản lý công ty:
  - Thêm mới công ty (POST /api/v1/companies): Nhà tuyển dụng có thể tạo hồ sơ công ty mới bằng cách cung cấp thông tin mô tả chi tiết về công ty.
  - Cập nhật thông tin công ty (PUT /api/v1/companies): Nhà tuyển dụng có thể chỉnh sửa và cập nhật thông tin của công ty đã tồn tại.
  - Xóa công ty (DELETE /api/v1/companies/{id}): Xóa hồ sơ công ty khỏi hệ thống khi không còn hoạt động hoặc khi có yêu cầu.
  - Xem chi tiết công ty (GET /api/v1/companies/{id}): Xem thông tin chi tiết về một công ty cụ thể.
  - Liệt kê danh sách công ty (GET /api/v1/companies): Truy xuất và hiển thị danh sách tất cả các công ty có trong hệ thống.

– Quản lý công việc:

- Thêm mới công việc (POST /api/v1/jobs): Nhà tuyển dụng có thể tạo bài đăng tuyển dụng mới, cung cấp chi tiết về công việc cần tuyển.
- Cập nhật công việc (PUT /api/v1/jobs): Nhà tuyển dụng có thể chỉnh sửa thông tin của bài đăng tuyển dụng hiện có.
- Xóa công việc (DELETE /api/v1/jobs/{id}): Xóa bài đăng tuyển dụng khỏi hệ thống khi công việc đã được lấp đầy hoặc không còn nhu cầu tuyển dụng.
- Xem chi tiết công việc (GET /api/v1/jobs/{id}): Xem thông tin chi tiết về một công việc cụ thể.
- Liệt kê danh sách công việc (GET /api/v1/jobs): Truy xuất và hiển thị danh sách tất cả các công việc có trong hệ thống.

– Quản lý hồ sơ ứng viên:

- Thêm mới hồ sơ (POST /api/v1/resumes): Người tìm việc có thể tạo và tải lên hồ sơ của họ để ứng tuyển vào các công việc.
- Cập nhật hồ sơ (PUT /api/v1/resumes): Nhà tuyển dụng có thể chỉnh sửa và cập nhật trạng thái của hồ sơ.
- Xóa hồ sơ (DELETE /api/v1/resumes/{id}): Nhà tuyển dụng xóa hồ sơ khỏi hệ thống.
- Xem chi tiết hồ sơ (GET /api/v1/resumes/{id}): Nhà tuyển dụng xem thông tin chi tiết về một hồ sơ cụ thể.
- Liệt kê danh sách hồ sơ (GET /api/v1/resumes): Nhà tuyển dụng Truy xuất và hiển thị danh sách tất cả các hồ sơ có trong hệ thống.

– Quản lý người dùng:

- Thêm mới người dùng (POST /api/v1/users): Quản trị viên có thể thêm mới người dùng vào hệ thống.
- Cập nhật thông tin người dùng (PUT /api/v1/users): Quản trị viên có thể chỉnh sửa thông tin của người dùng hiện có.

- Xóa người dùng (DELETE /api/v1/users/{id}): Xóa tài khoản người dùng khỏi hệ thống khi không còn sử dụng hoặc khi có yêu cầu.
- Xem chi tiết người dùng (GET /api/v1/users/{id}): Xem thông tin chi tiết về một người dùng cụ thể.
- Liệt kê danh sách người dùng (GET /api/v1/users): Truy xuất và hiển thị danh sách tất cả các người dùng có trong hệ thống.

– Quản lý quyền và vai trò:

- Thêm mới quyền (POST /api/v1/permissions): Quản trị viên có thể thêm quyền mới vào hệ thống.
- Cập nhật quyền (PUT /api/v1/permissions): Quản trị viên có thể chỉnh sửa thông tin của các quyền hiện có.
- Xóa quyền (DELETE /api/v1/permissions/{id}): Xóa quyền khỏi hệ thống khi không còn cần thiết.
- Xem chi tiết quyền (GET /api/v1/permissions/{id}): Xem thông tin chi tiết về một quyền cụ thể.
- Liệt kê danh sách quyền (GET /api/v1/permissions): Truy xuất và hiển thị danh sách tất cả các quyền có trong hệ thống.
- Thêm mới vai trò (POST /api/v1/roles): Quản trị viên có thể thêm vai trò mới vào hệ thống.
- Cập nhật vai trò (PUT /api/v1/roles): Quản trị viên có thể chỉnh sửa thông tin của các vai trò hiện có.
- Xóa vai trò (DELETE /api/v1/roles/{id}): Xóa vai trò khỏi hệ thống khi không còn cần thiết.
- Xem chi tiết vai trò (GET /api/v1/roles/{id}): Xem thông tin chi tiết về một vai trò cụ thể.
- Liệt kê danh sách vai trò (GET /api/v1/roles): Truy xuất và hiển thị danh sách tất cả các vai trò có trong hệ thống.

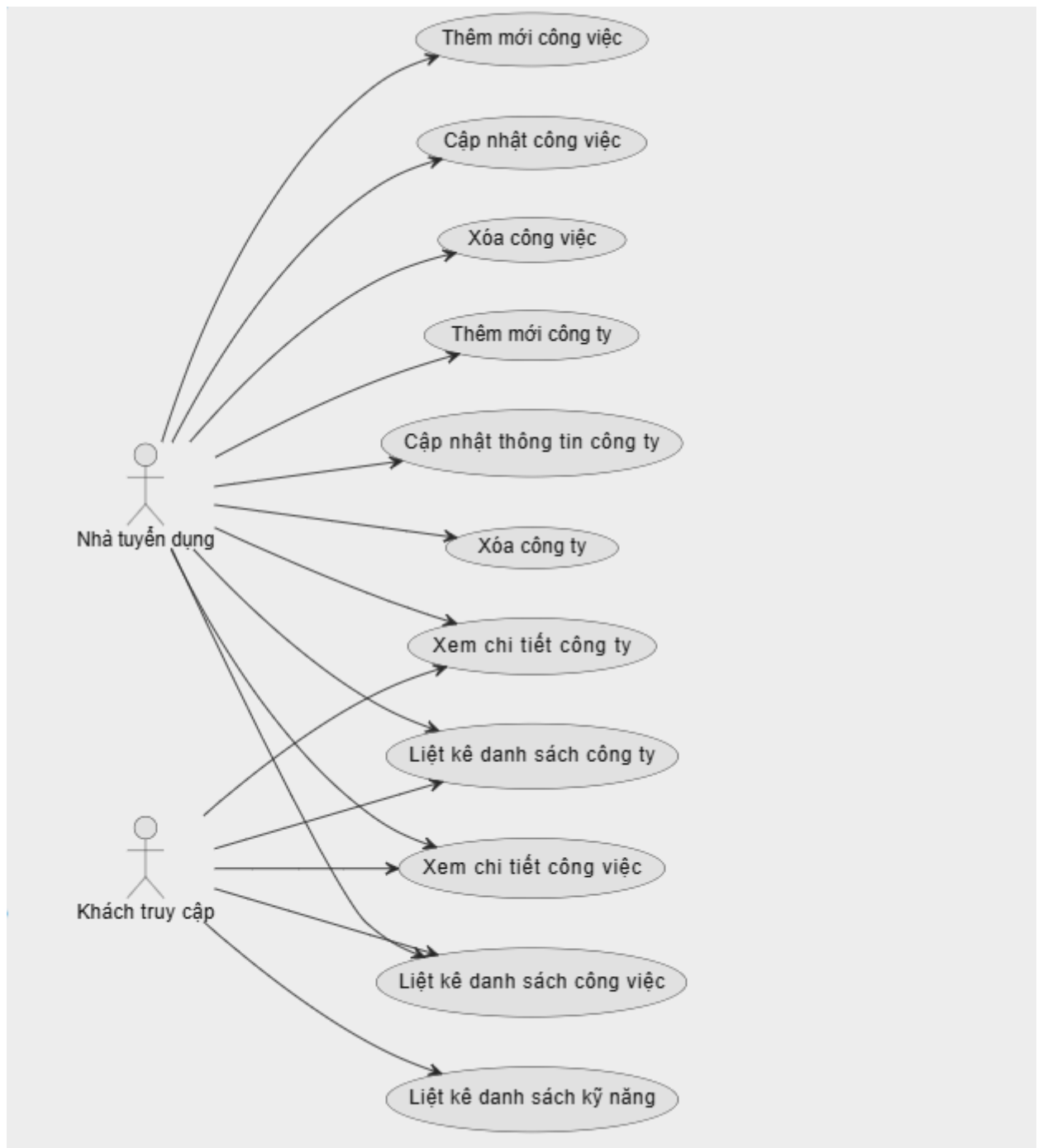
– Quản lý người đăng ký:

- Thêm mới người đăng ký (POST /api/v1/subscribers): Quản trị viên có thể thêm mới người đăng ký vào hệ thống.

- Cập nhật thông tin người đăng ký (PUT /api/v1/subscribers): Quản trị viên có thể chỉnh sửa thông tin của người đăng ký hiện có.
  - Xóa người đăng ký (DELETE /api/v1/subscribers/{id}): Xóa thông tin người đăng ký khỏi hệ thống khi không còn cần thiết.
  - Xem chi tiết người đăng ký (GET /api/v1/subscribers/{id}): Xem thông tin chi tiết về một người đăng ký cụ thể.
  - Liệt kê danh sách người đăng ký (GET /api/v1/subscribers): Truy xuất và hiển thị danh sách tất cả các người đăng ký có trong hệ thống.
- Quản lý tệp tin:
- Thêm mới tệp tin (POST /api/v1/files): Quản trị viên hoặc người dùng có thể tải lên các tệp tin mới vào hệ thống.
  - Liệt kê danh sách tệp tin (GET /api/v1/files): Truy xuất và hiển thị danh sách tất cả các tệp tin có trong hệ thống.
- Quản lý kỹ năng:
- Thêm mới kỹ năng (POST /api/v1/skills): Quản trị viên có thể thêm kỹ năng mới vào hệ thống.
  - Cập nhật kỹ năng (PUT /api/v1/skills): Quản trị viên có thể chỉnh sửa thông tin của các kỹ năng hiện có.
  - Xóa kỹ năng (DELETE /api/v1/skills): Xóa kỹ năng khỏi hệ thống khi không còn cần thiết.
  - Liệt kê danh sách kỹ năng (GET /api/v1/skills): Truy xuất và hiển thị danh sách tất cả các kỹ năng có trong hệ thống.
- Xác thực và Đăng nhập:
- Đăng nhập (POST /api/v1/auth/login): Người dùng có thể đăng nhập vào hệ thống bằng thông tin tài khoản của mình.
  - Đăng ký tài khoản (POST /api/v1/auth/register): Người dùng mới có thể đăng ký tài khoản mới để truy cập vào hệ thống.

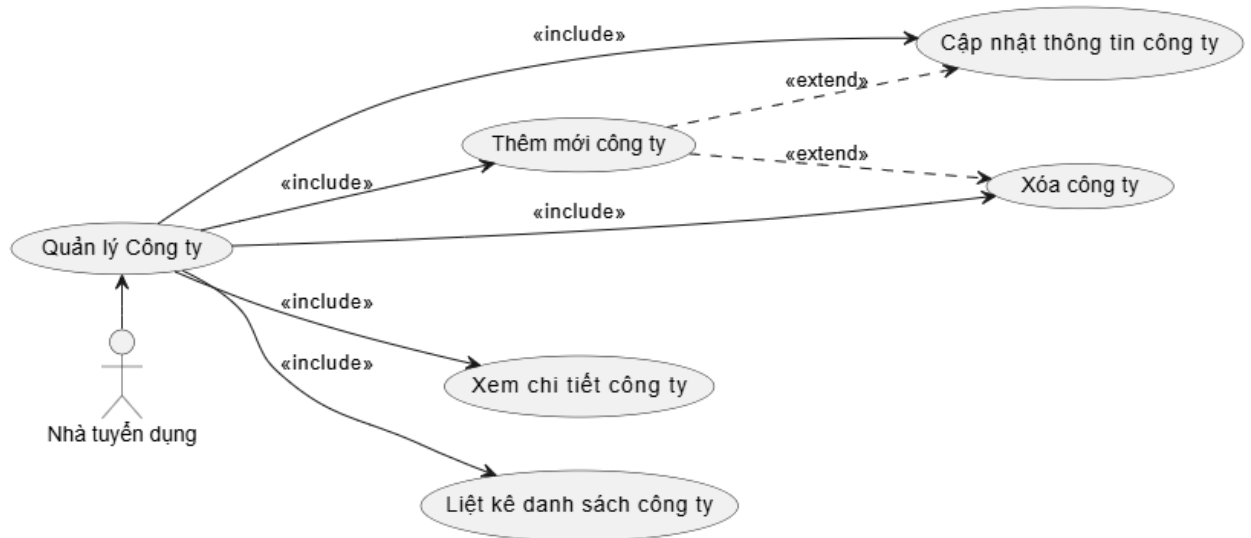
### 3.4 Biểu đồ Usecase

#### 3.4.1 Biểu đồ usecase tổng quát



Hình 3.1. Biểu đồ usecase tổng quát

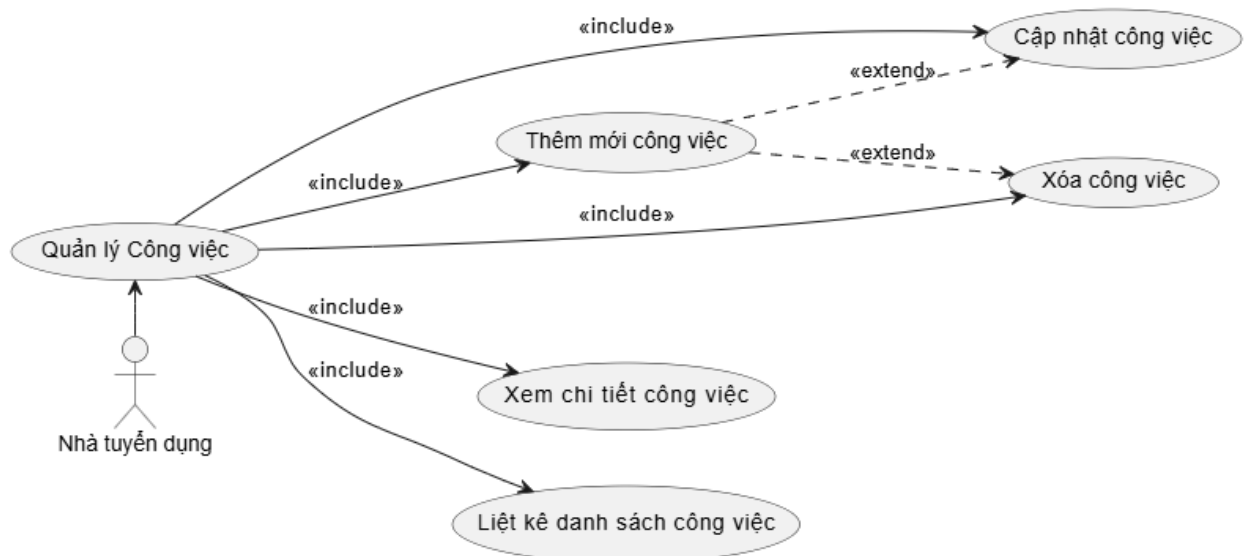
### 3.4.2 Biểu đồ usecase quản lý công ty



Hình 3.2. Biểu đồ usecase quản lý Công ty

Tóm tắt: Biểu đồ này mô tả nhà tuyển dụng có thể quản lý thông tin công ty, các tác vụ như thêm mới đến cập nhật và xóa thông tin công ty.

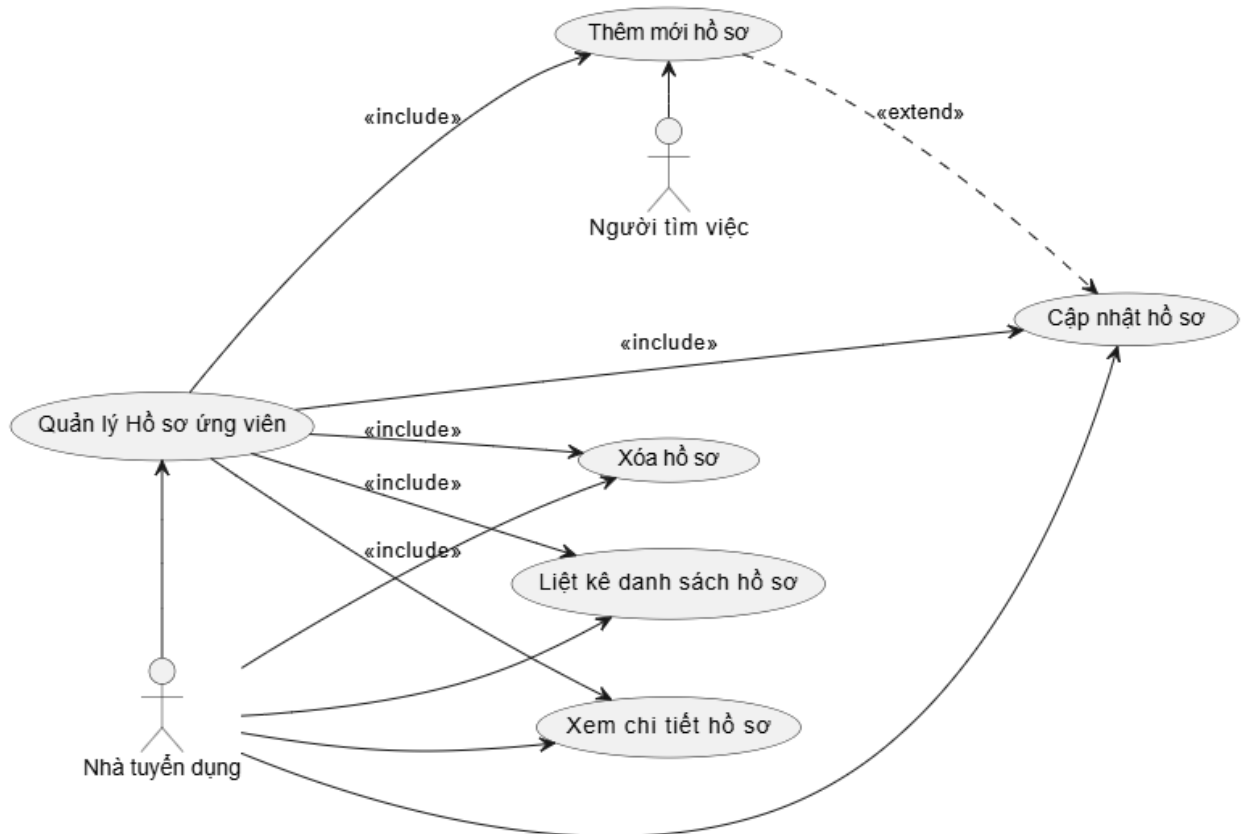
### 3.4.3. Biểu đồ usecase quản lý danh sách người dùng



Hình 3.3. Biểu đồ usecase quản lý công việc

Tóm tắt: Biểu đồ này mô tả nhà tuyển dụng tương tác với hệ thống để quản lý các công việc tuyển dụng một cách hiệu quả. Các tác vụ như thêm mới, cập nhật, xóa, xem chi tiết và liệt kê danh sách công việc đều được tích hợp trong quy trình quản lý công việc.

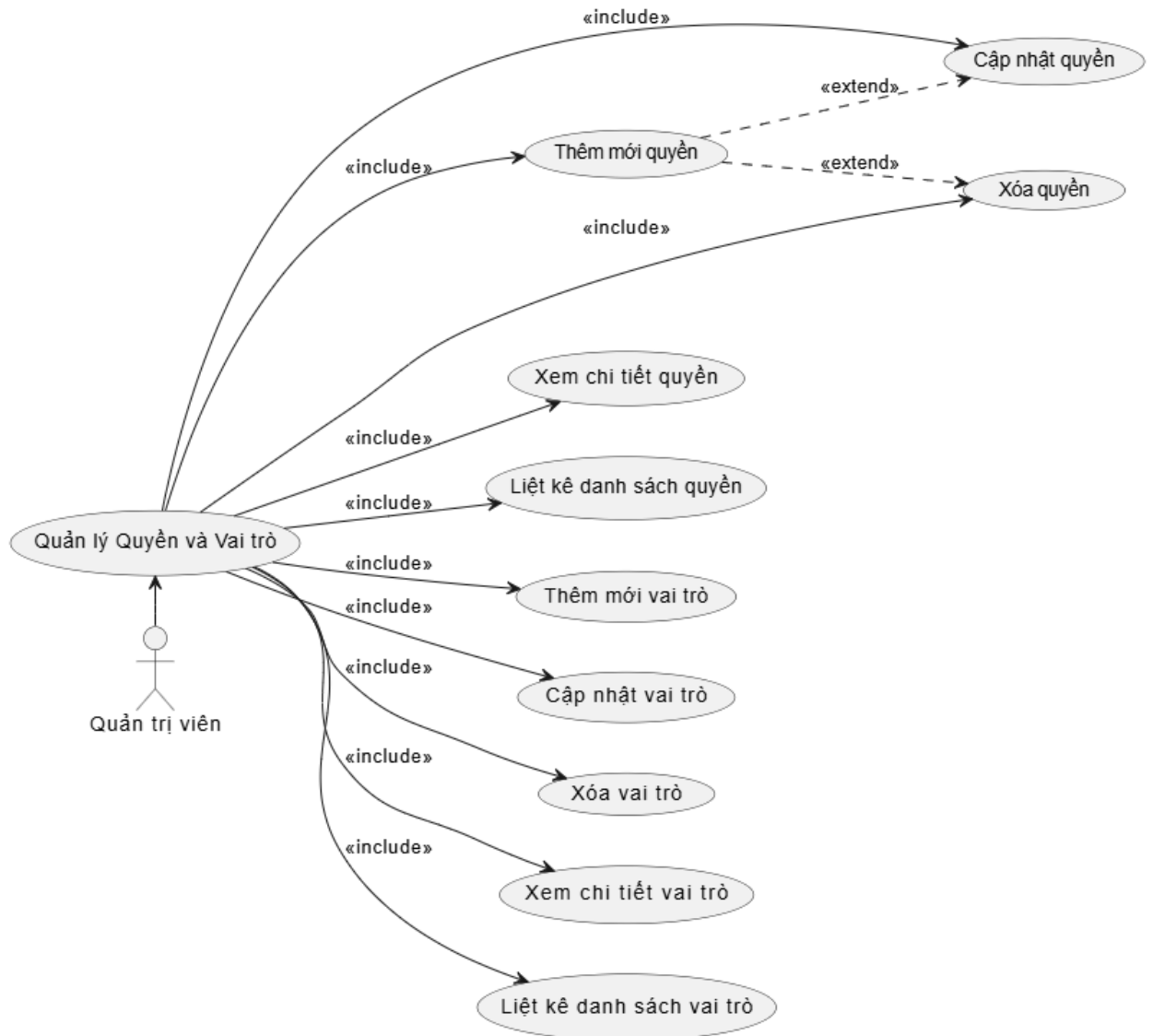
### 3.4.3 Biểu đồ usecase quản lý hồ sơ ứng viên



Hình 3.4. Biểu đồ usecase quản lý hồ sơ ứng viên

Tóm tắt: Biểu đồ này cung cấp một cái nhìn toàn diện về cách các tác vụ liên quan đến hồ sơ ứng viên được quản lý bởi các tác nhân khác nhau, từ đó đảm bảo rằng tất cả các tác vụ quan trọng đều được hỗ trợ một cách nhất quán và hiệu quả trong hệ thống quản lý việc làm.

### 3.4.4 Biểu đồ usecase quản lý quyền và vai trò

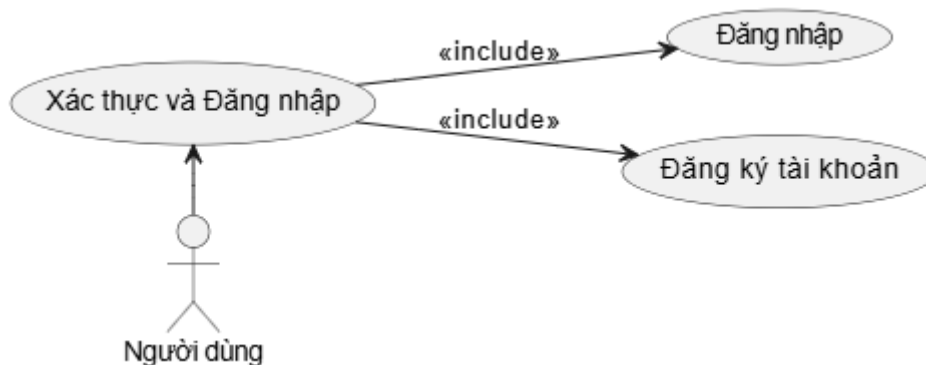


Hình 3.5. Biểu đồ usecase quản lý quyền và vai trò của người dùng

Tóm tắt: Biểu đồ này mô tả ca sử dụng của quản trị viên thực hiện các thao tác như tạo mới, cập nhật và xóa quyền hạn.



### 3.4.5 Biểu đồ usecase xác thực và đăng nhập



Hình 3.6. Biểu đồ usecase xác thực và đăng nhập

Tóm tắt: Biểu đồ này mô tả ca sử dụng của người dùng tương tác với hệ thống để thực hiện các tác vụ liên quan đến đăng nhập và đăng ký tài khoản. Cụ thể:

Người dùng có thể đăng nhập vào hệ thống bằng tài khoản đã được tạo trước đó.

Người dùng có thể đăng ký một tài khoản mới nếu như chưa có tài khoản.

## 3.5 Đặc tả Usecase

### 3.5.1 Usecase quản lý thông tin của một công ty

- Tóm tắt: Mô tả quy trình quản lý thông tin của một công ty.
- Dòng sự kiện chính:
  - o Use case này được sử dụng khi cần quản lý thông tin của các công ty trong hệ thống.
  - o Danh sách các công ty được hiển thị cho quản trị viên.
  - o Quản trị viên có thể bấm vào để xem thông tin chi tiết của từng công ty, lựa chọn kiểm duyệt hoặc bỏ kiểm duyệt để quyết định xem công ty nào được phép hiển thị trên hệ thống, hoặc xóa đi các công ty không hợp lệ.

### 3.5.2 Usecase quản lý danh sách người dùng

- Tóm tắt: Mô tả quy trình quản lý danh sách người dùng của hệ thống
- Dòng sự kiện chính:
  - o Ca này được sử dụng khi cần quản lý các thành viên trong hệ thống

- Danh sách người dùng được hiển thị cho người quản lý
- Người quản lý có thể thêm mới, xóa bỏ hoặc chỉnh sửa thông tin người dùng.

### **3.5.3 Usecase quản lý hồ sơ ứng viên trong hệ thống**

- Tóm tắt: Mô tả quy trình quản lý hồ sơ ứng viên của hệ thống
- Dòng sự kiện chính:
  - Use case này được sử dụng khi cần quản lý các hồ sơ ứng viên trong hệ thống.
  - Danh sách các hồ sơ ứng viên được hiển thị cho nhà tuyển dụng.
  - Nhà tuyển dụng có thể bấm vào để xem thông tin chi tiết của từng hồ sơ ứng viên, lựa chọn kiểm duyệt với trạng thái là Pending, Approved và Rejected.

### **3.5.4 Usecase quản lý quyền hạn và vai trò của người dùng**

- Tóm tắt: Mô tả quy trình quản lý quyền hạn và vai trò của hệ thống
- Dòng sự kiện chính:
  - Use case này được sử dụng khi cần quản lý các quyền hạn và vai trò trong hệ thống.
  - Danh sách các quyền hạn và vai trò hiện có sẽ được hiển thị cho quản trị viên.
  - Quản trị viên có thể bấm vào để xem thông tin chi tiết của từng quyền hạn và vai trò, lựa chọn kiểm duyệt hoặc bỏ kiểm duyệt để quyết định xem quyền hạn và vai trò nào được phép hiển thị trên hệ thống, hoặc xóa vĩnh viễn các quyền hạn và vai trò không hợp lệ.

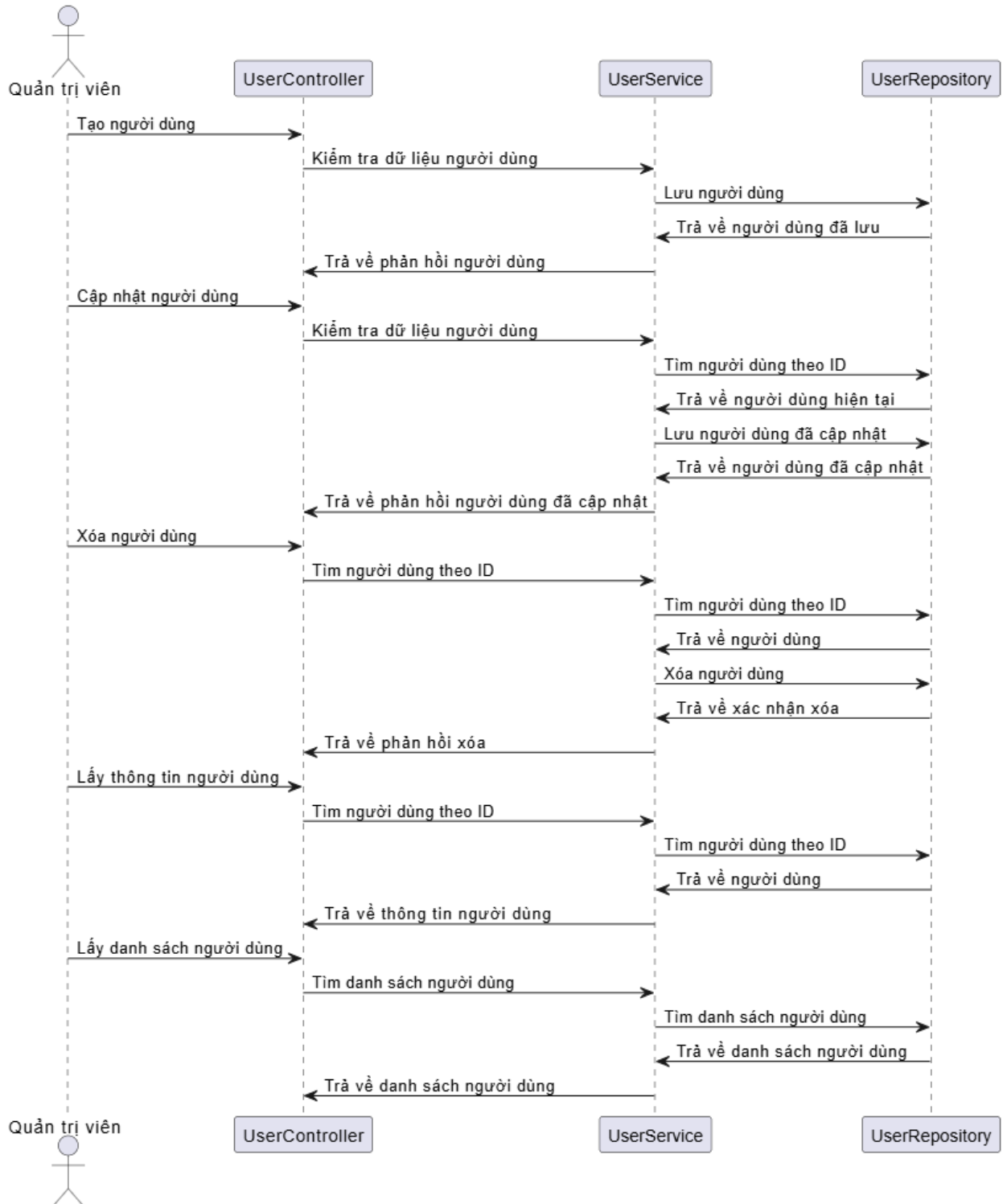
## 3.6 Biểu đồ tuần tự

### 3.6.1. Biểu đồ tuần tự quản lý thông tin công ty



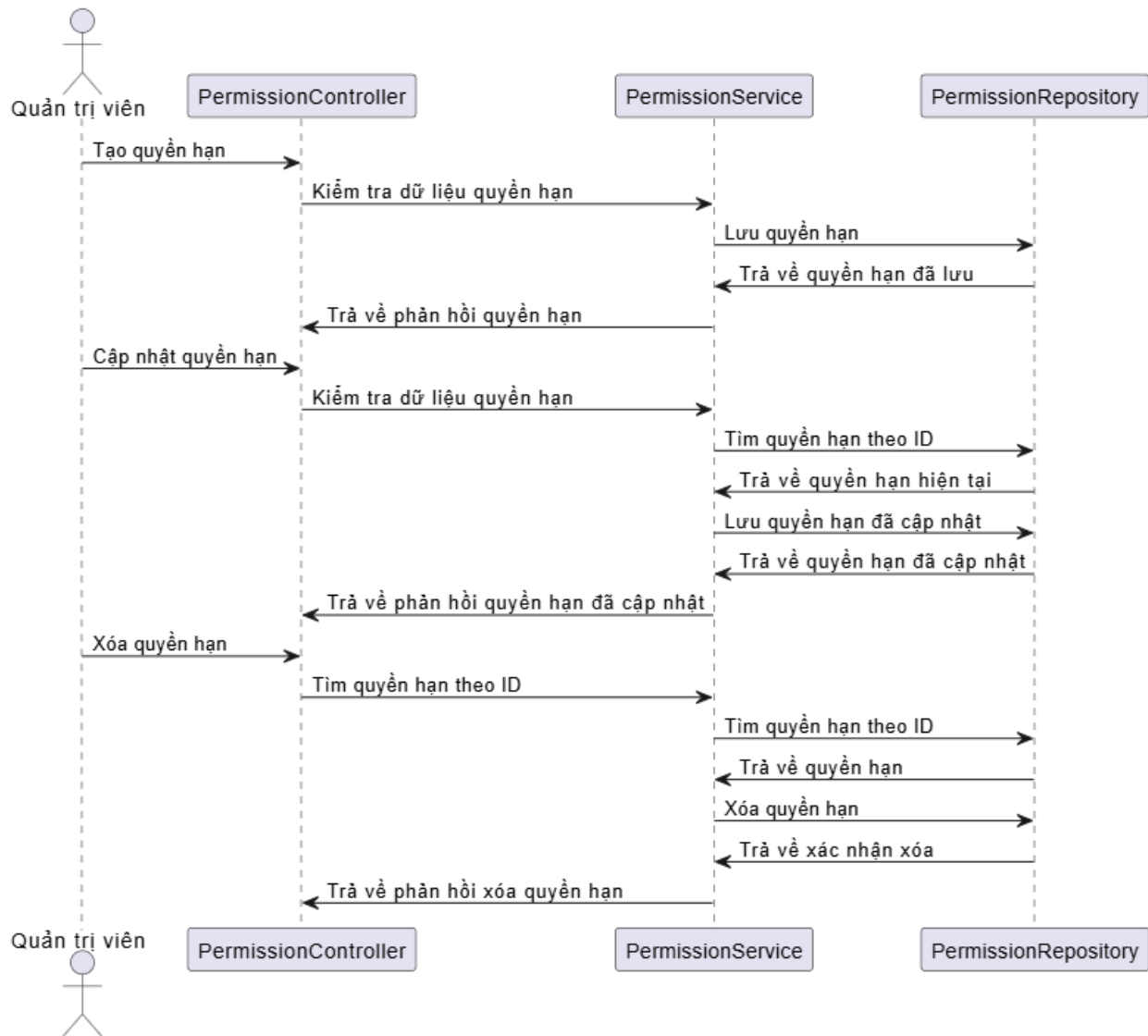
Hình 3.7: Biểu đồ tuần tự quản lý thông tin công ty.

### 3.6.2. Biểu đồ tuần tự quản lý người dùng

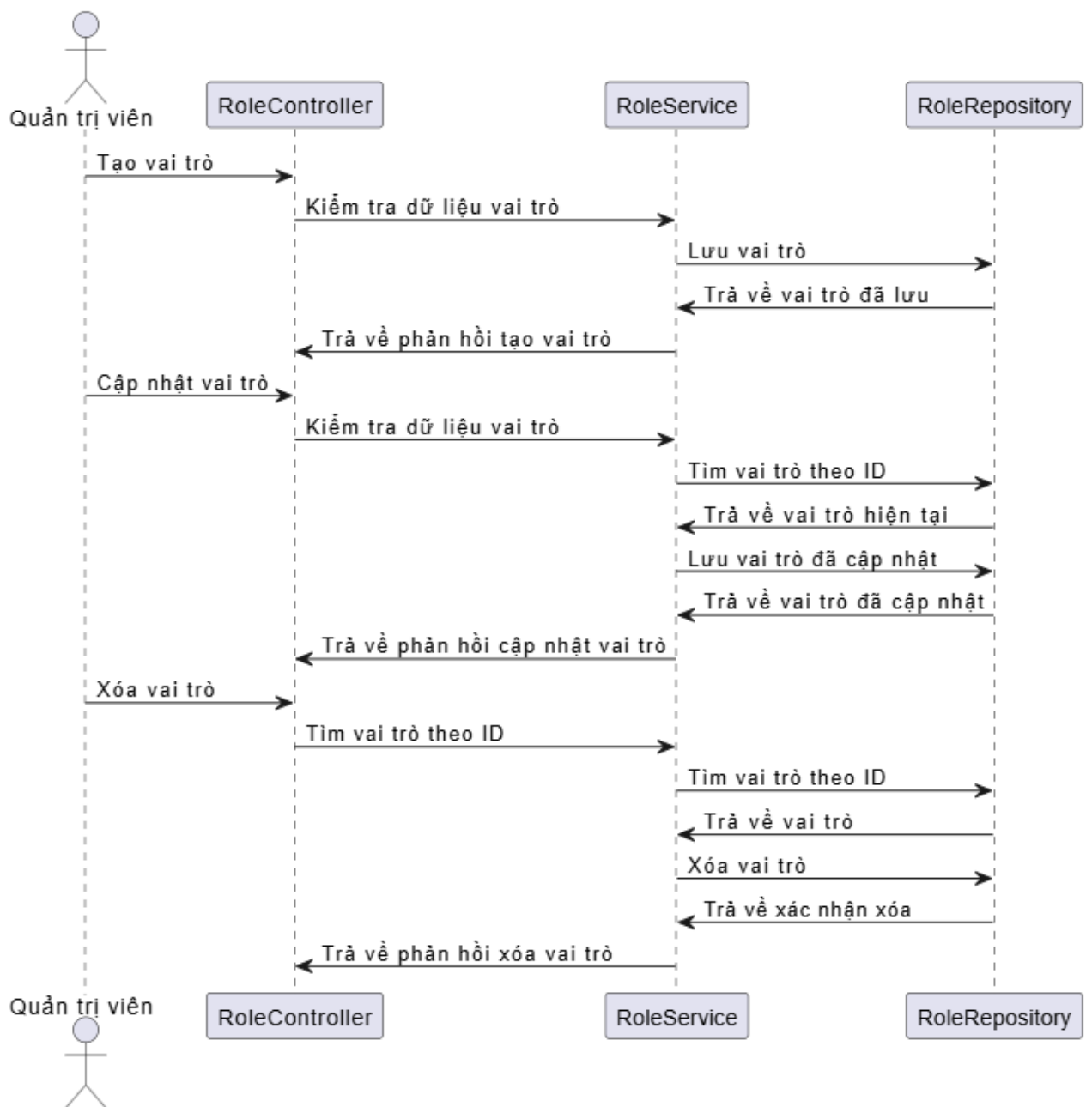


Hình 3.8: Biểu đồ tuần tự quản lý xem thông tin người dùng

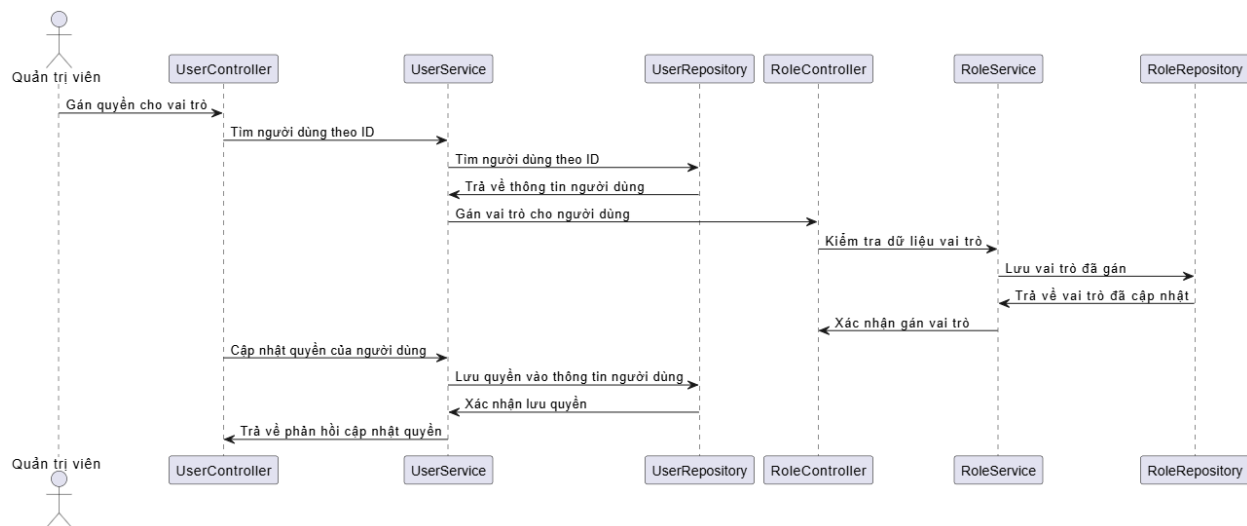
### 3.6.2. Biểu đồ tuần tự quản lý quyền hạn và vai trò



Hình 3.9: Biểu đồ tuần tự quản lý quyền hạn



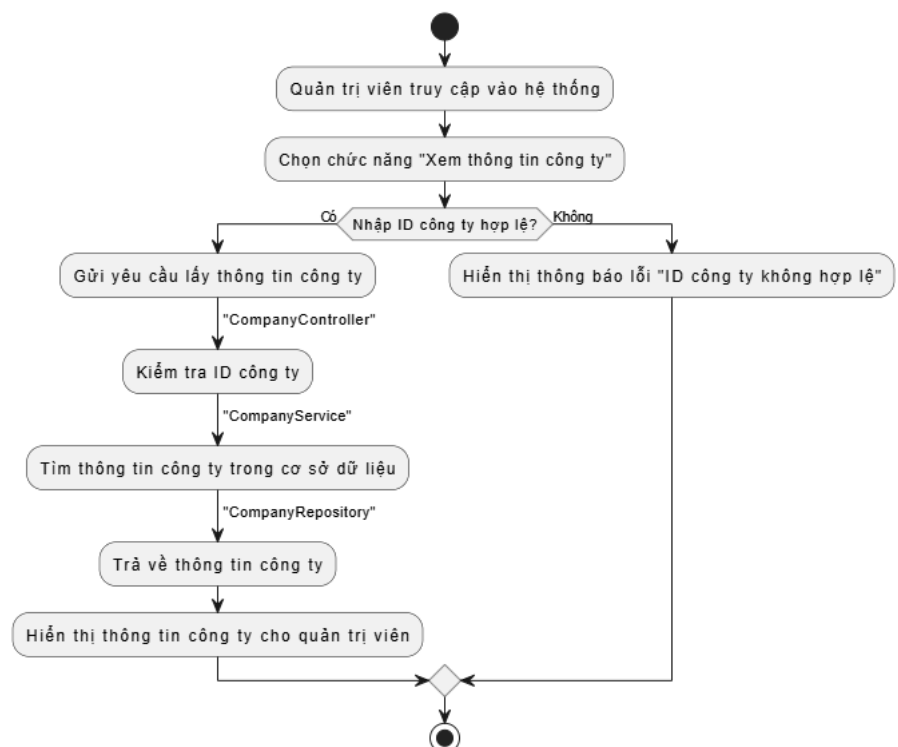
Hình 3.10: Biểu đồ tuần tự quản lý vai trò



Hình 3.11: Biểu đồ tuần tự quản lý quản lý vai trò và gán quyền cho người dùng

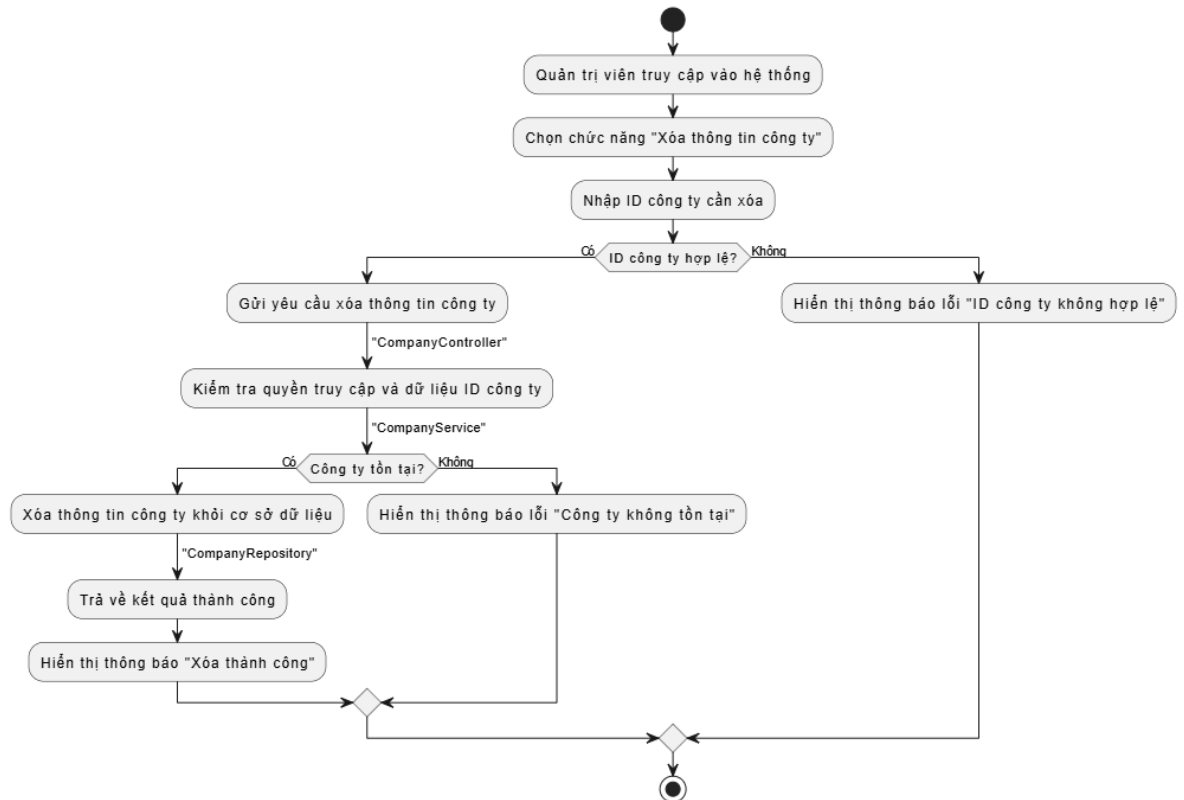
### 3.7. Biểu đồ hoạt động

#### 3.7.1. Xem thông tin của một công ty



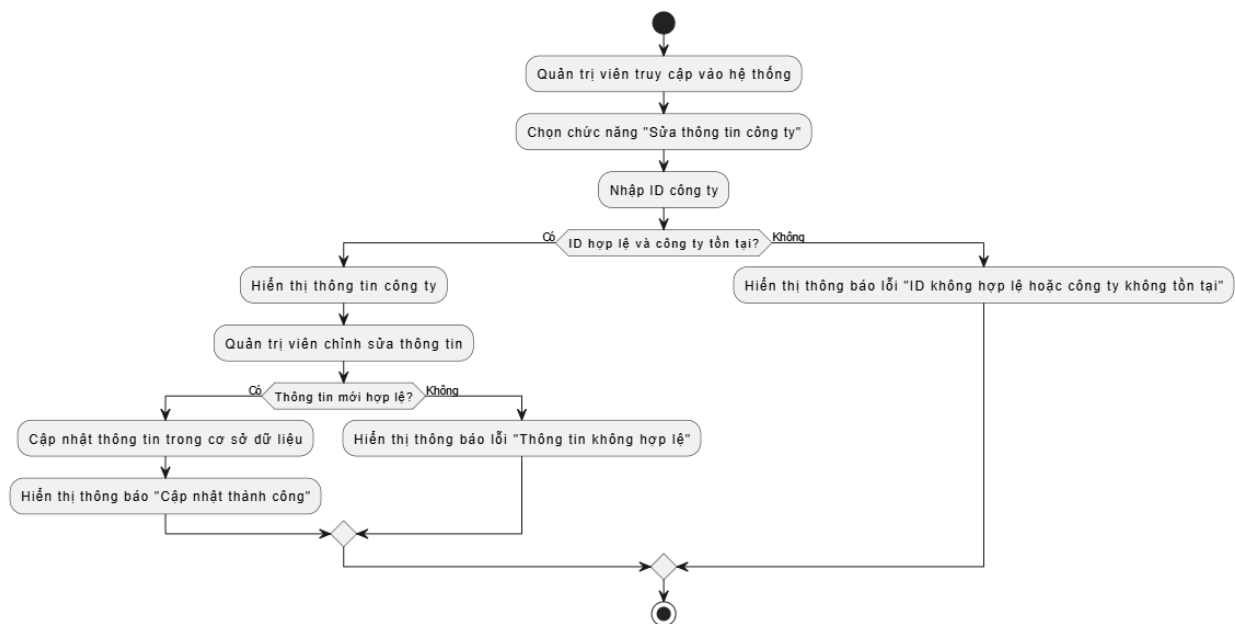
Hình 3.12: Biểu đồ hoạt động xem thông tin tour du lịch

### 3.7.2. Xóa thông tin của công ty



Hình 3.13: Biểu đồ hoạt động xóa thông tin công ty.

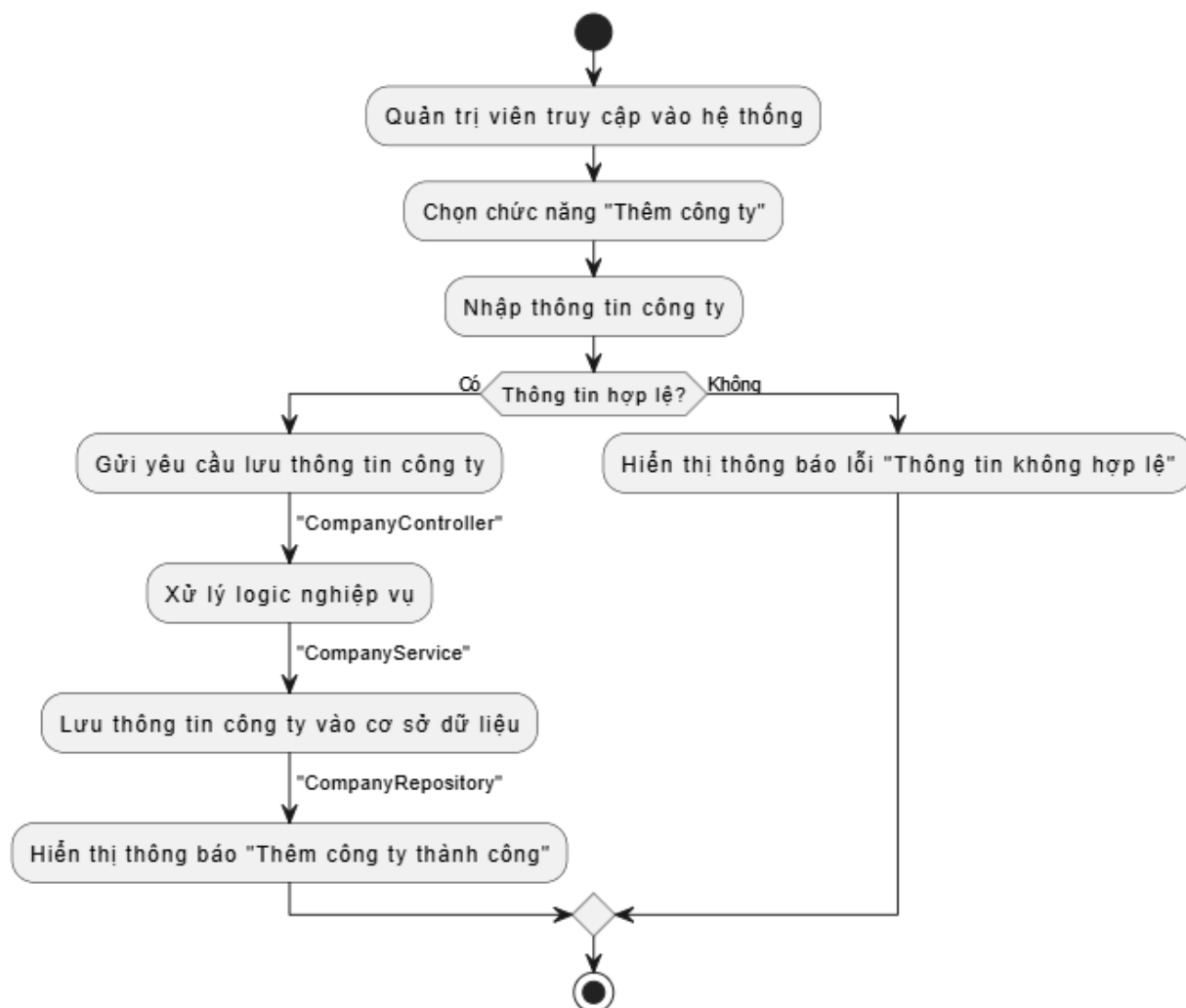
### 3.7.3. Sửa thông tin công ty



Hình 3.14: Biểu đồ hoạt động sửa thông tin công ty.

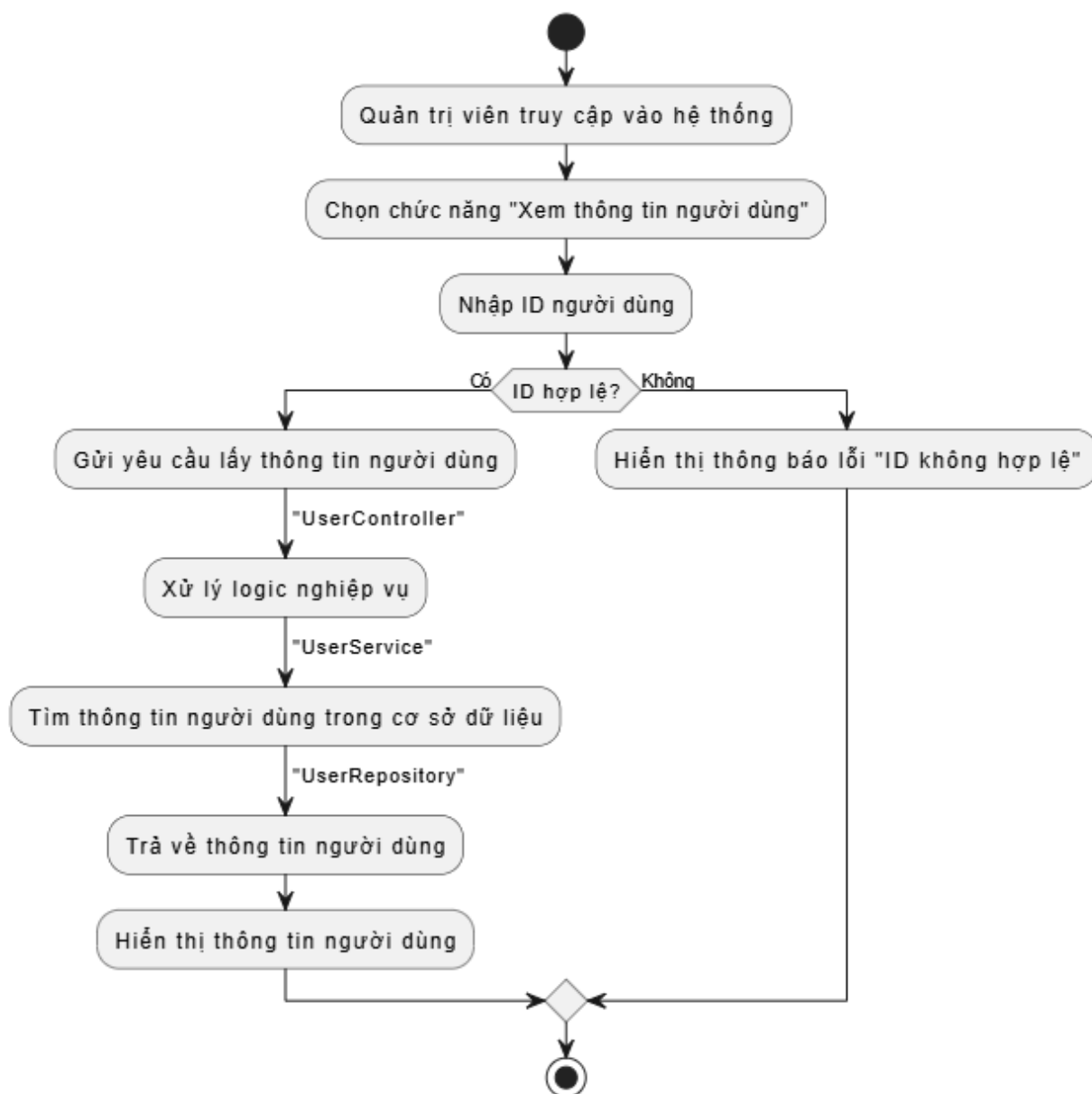


#### 3.7.4. Thêm thông tin cho công ty



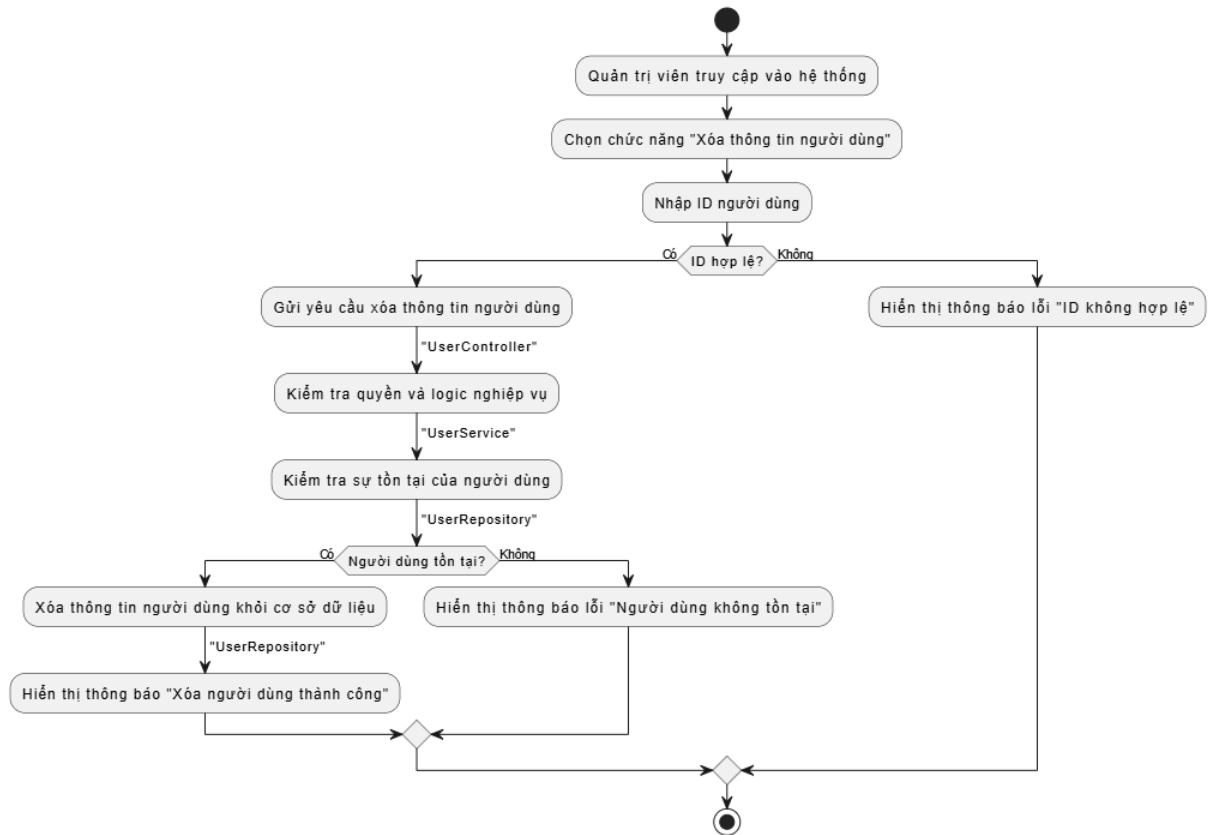
Hình 3.15: Biểu đồ hoạt động thêm thông tin công ty

### 3.7.5. Xem thông tin của người dùng



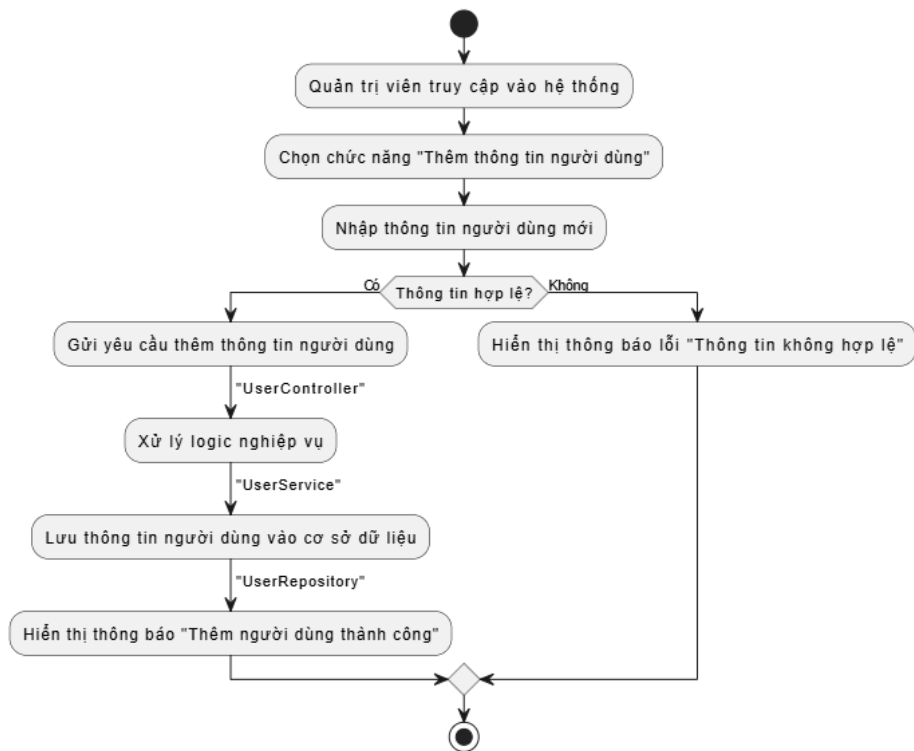
Hình 3.16: Biểu đồ hoạt động xem thông tin của người dùng

### 3.7.6. Xóa thông tin người dùng



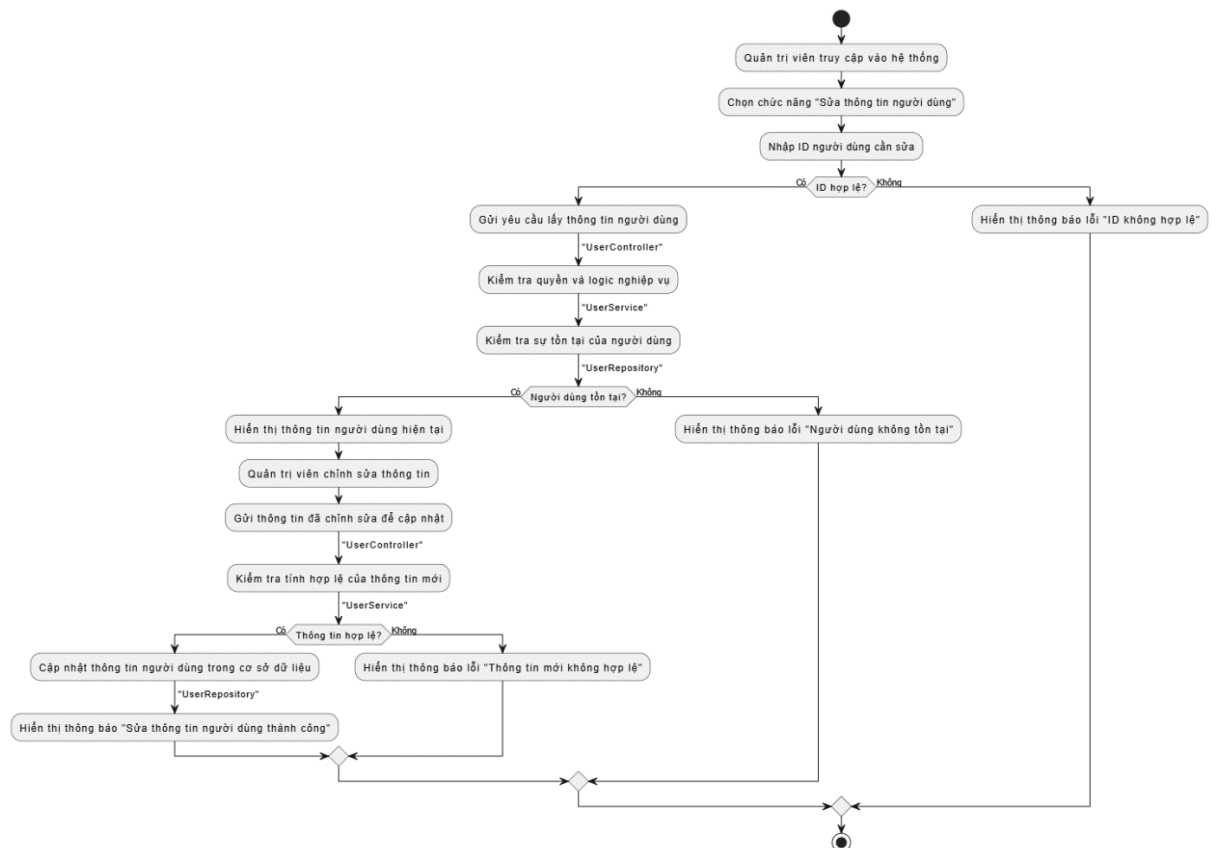
Hình 3.17: Biểu đồ hoạt động xóa thông tin người dùng

### 3.7.7. Thêm thông tin người dùng



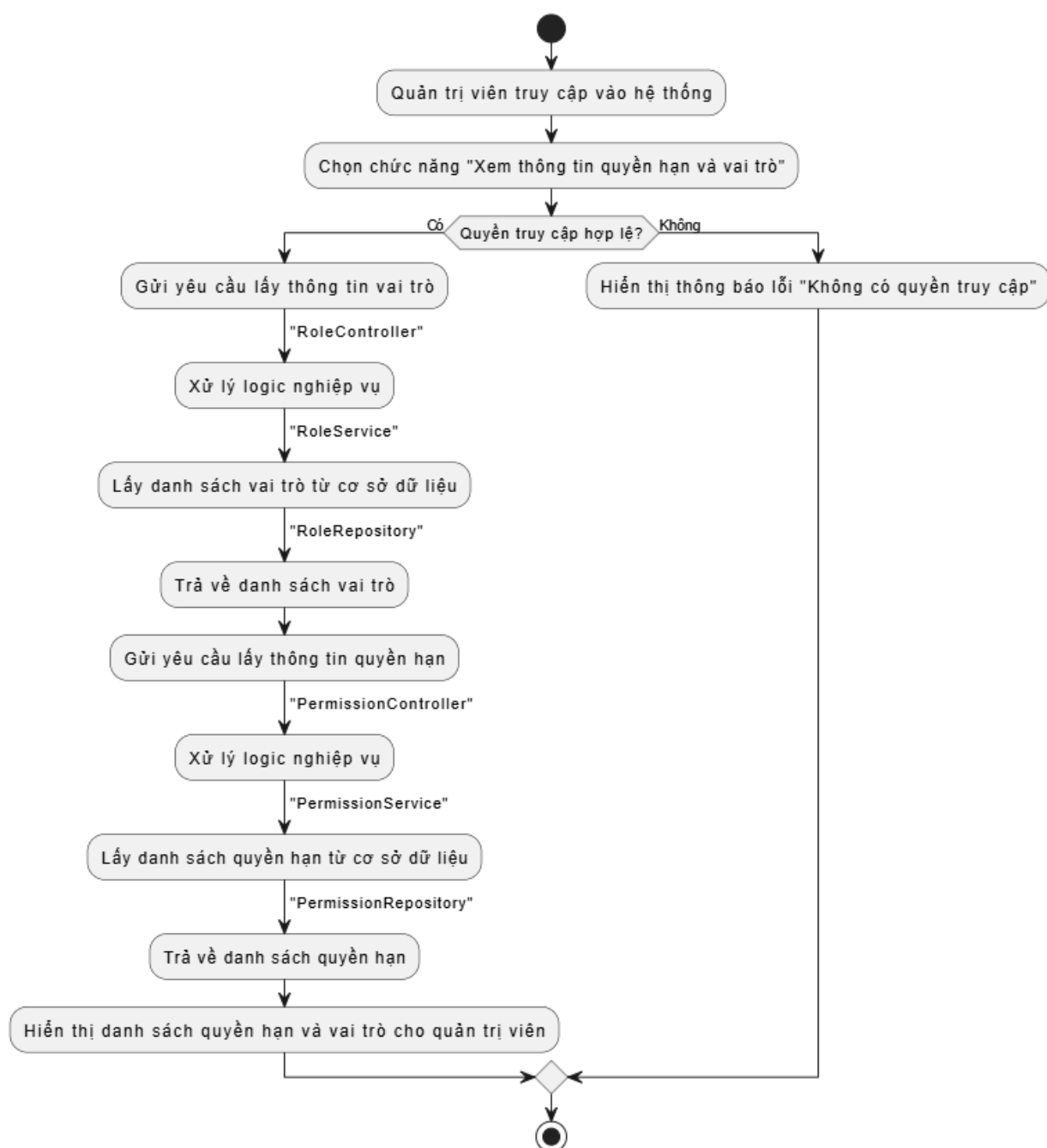
Hình 3.18: Biểu đồ hoạt động thêm thông tin người dùng

### 3.7.8. Sửa thông tin người dùng



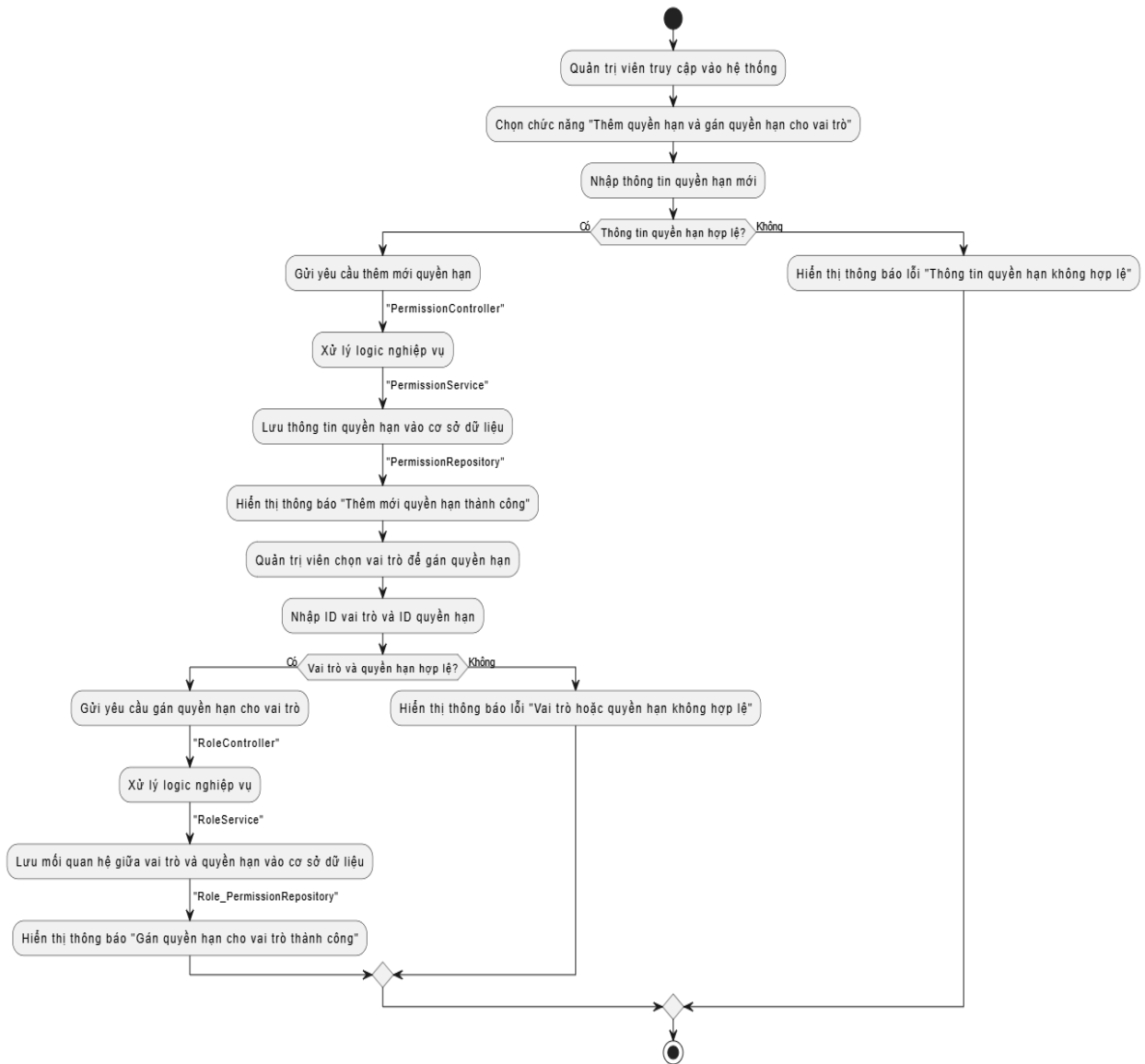
Hình 3.19: Biểu đồ hoạt động chỉnh sửa thông tin người dùng

**3.7.9. Xem thông tin các quyền hạn và vai trò có trong hệ thống**



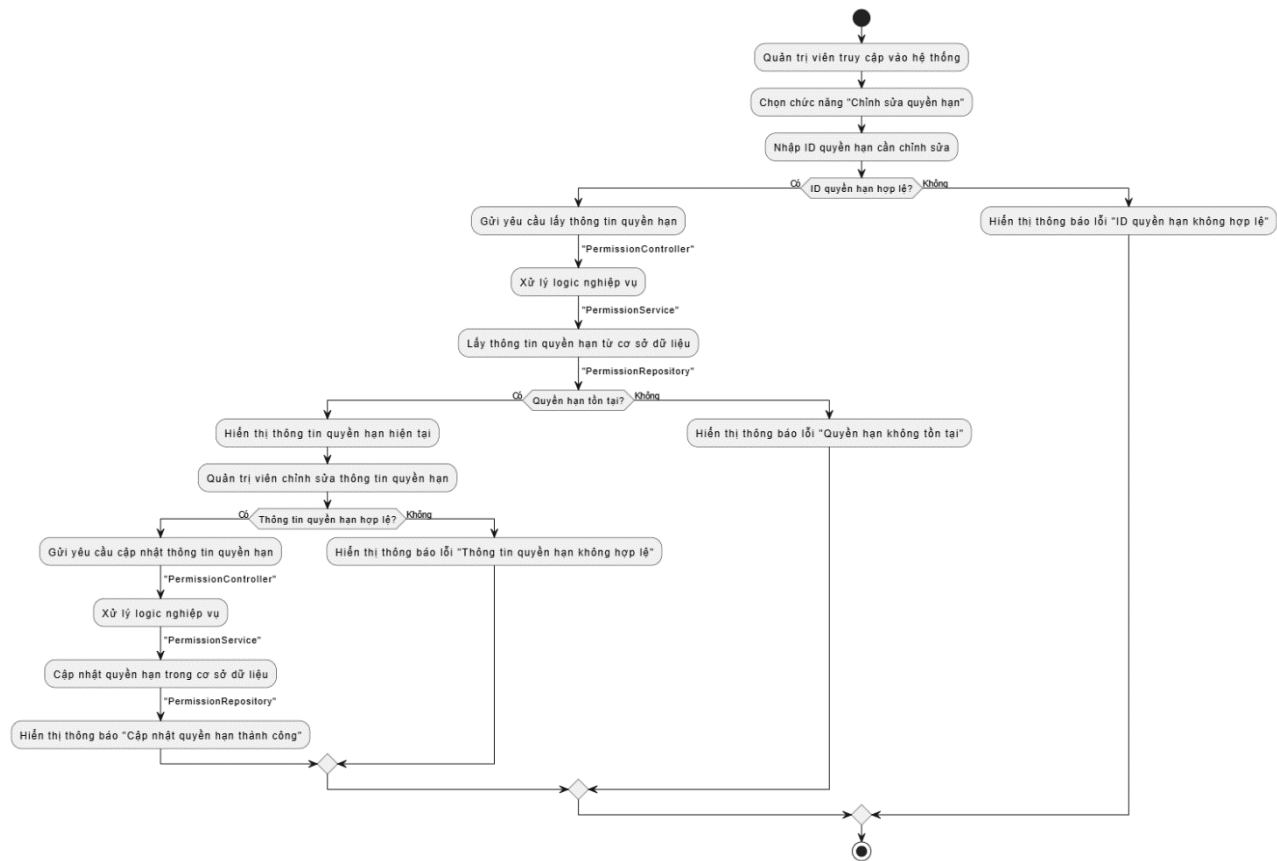
Hình 3.20: Biểu đồ hoạt động xem thông tin quyền hạn và vai trò có trong hệ thống

### 3.7.10. Thêm mới một quyền hạn và gán quyền hạn cho vai trò có trong hệ thống



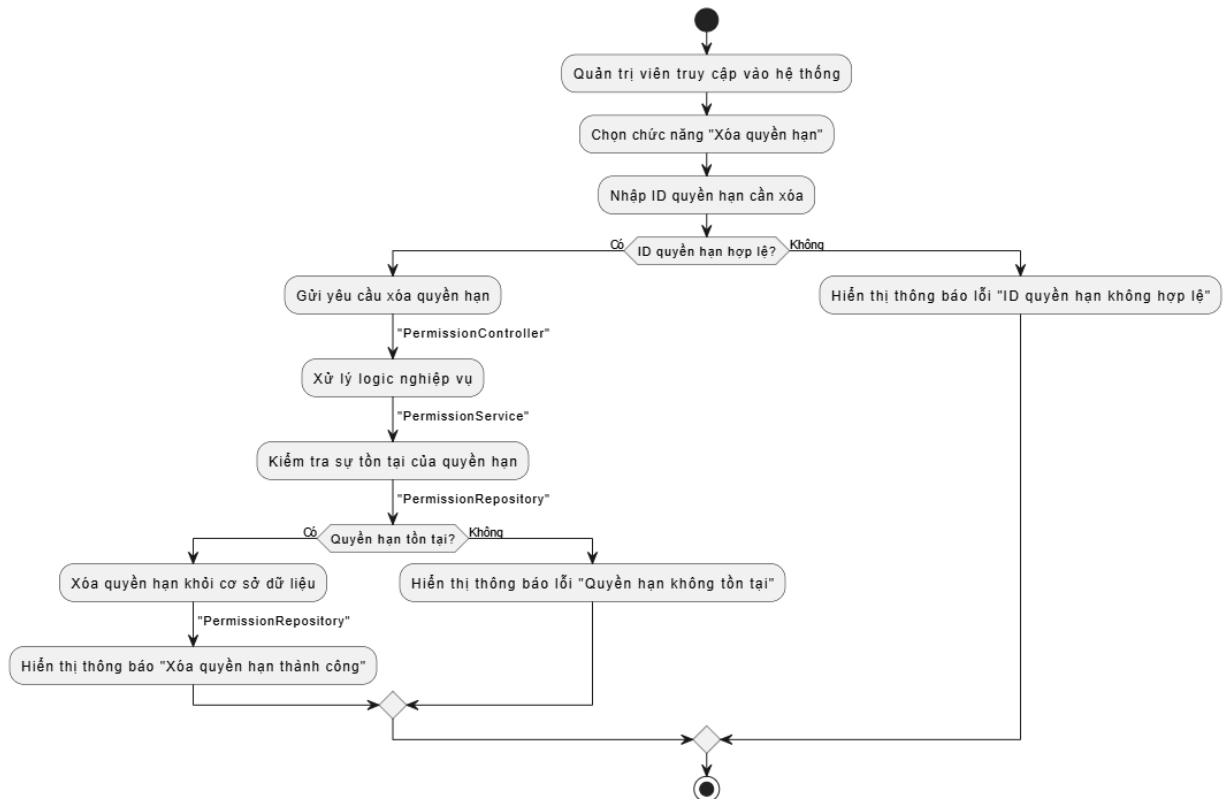
Hình 3.21: Biểu đồ hoạt động thêm mới một quyền hạn và gán quyền hạn cho vai một trò có trong hệ thống

### 3.7.11. Chỉnh sửa một quyền trong hệ thống.



Hình 3.22: Biểu đồ hoạt động chỉnh sửa một quyền hạn có trong hệ thống

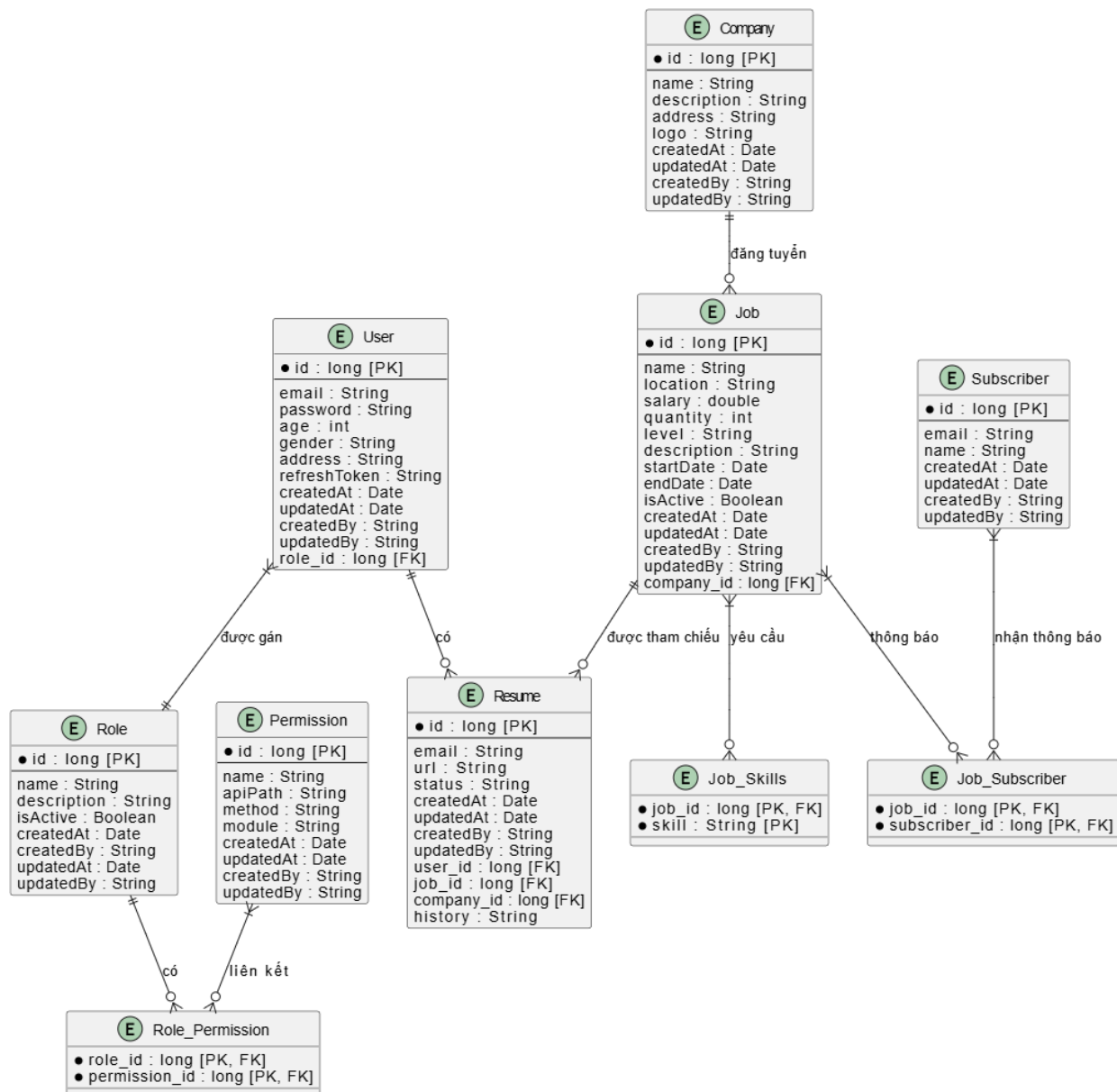
### 3.7.12. Xóa bỏ một quyền hạn có trong hệ thống.



Hình 3.23: Biểu đồ hoạt động chỉnh sửa một quyền hạn có trong hệ thống

### 3.7.13. Thiết kế cơ sở dữ liệu.





Hình 3.24: Biểu đồ ERD

Bảng User chứa thông tin về người dùng trong hệ thống. Bảng này bao gồm các trường sau:

- Id (Mã): Mã định danh duy nhất của người dùng.
- Email (Email): Địa chỉ email của người dùng.
- Password (Mật khẩu): Mật khẩu được mã hóa.
- Age (Tuổi): Tuổi của người dùng.
- Gender (Giới tính): Giới tính của người dùng.
- Address (Địa chỉ): Địa chỉ của người dùng.

- RefreshToken (Mã làm mới): Mã token dùng để làm mới phiên đăng nhập.
- CreatedAt (Ngày tạo): Ngày tạo bản ghi.
- UpdatedAt (Ngày sửa): Ngày cập nhật bản ghi.
- CreatedBy (Tạo bởi): Người tạo bản ghi.
- UpdatedBy (Sửa bởi): Người sửa bản ghi.
- RoleId (Mã vai trò): Mã vai trò của người dùng (khóa ngoại liên kết với bảng Role).

Bảng Role chứa thông tin về vai trò trong hệ thống. Bảng này bao gồm các trường sau:

- Id (Mã): Mã định danh duy nhất của vai trò.
- Name (Tên vai trò): Tên của vai trò.
- Description (Mô tả): Mô tả chi tiết về vai trò.
- IsActive (Hoạt động): Trạng thái hoạt động của vai trò.
- CreatedAt (Ngày tạo): Ngày tạo bản ghi.
- CreatedBy (Tạo bởi): Người tạo bản ghi.
- UpdatedAt (Ngày sửa): Ngày cập nhật bản ghi.
- UpdatedBy (Sửa bởi): Người sửa bản ghi.

Bảng Permission chứa thông tin về quyền trong hệ thống. Bảng này bao gồm các trường sau:

- Id (Mã): Mã định danh duy nhất của quyền.
- Name (Tên quyền): Tên của quyền.
- ApiPath (Đường dẫn API): Đường dẫn API liên quan đến quyền.
- Method (Phương thức): Phương thức HTTP được áp dụng (GET, POST, PUT, DELETE).
- Module (Module): Module mà quyền này thuộc về.
- CreatedAt (Ngày tạo): Ngày tạo bản ghi.
- UpdatedAt (Ngày sửa): Ngày cập nhật bản ghi.
- CreatedBy (Tạo bởi): Người tạo bản ghi.
- UpdatedBy (Sửa bởi): Người sửa bản ghi.

Bảng Company chứa thông tin về các công ty. Bảng này bao gồm các trường sau:

- Id (Mã): Mã định danh duy nhất của công ty.
- Name (Tên công ty): Tên của công ty.
- Description (Mô tả): Mô tả chi tiết về công ty.
- Address (Địa chỉ): Địa chỉ của công ty.
- Logo (Logo): Đường dẫn tới logo của công ty.
- CreatedAt (Ngày tạo): Ngày tạo bản ghi.
- UpdatedAt (Ngày sửa): Ngày cập nhật bản ghi.
- CreatedBy (Tạo bởi): Người tạo bản ghi.
- UpdatedBy (Sửa bởi): Người sửa bản ghi.

Bảng Job chứa thông tin về các công việc được đăng tuyển bởi các công ty. Bảng này bao gồm các trường sau:

- Id (Mã): Mã định danh duy nhất của công việc.

- Name (Tên công việc): Tên của công việc.
- Location (Địa điểm): Địa điểm làm việc.
- Salary (Mức lương): Mức lương của công việc.
- Quantity (Số lượng): Số lượng tuyển dụng.
- Level (Cấp bậc): Cấp bậc của công việc.
- Description (Mô tả): Mô tả chi tiết về công việc.
- StartDate (Ngày bắt đầu): Ngày bắt đầu công việc.
- EndDate (Ngày kết thúc): Ngày kết thúc công việc.
- IsActive (Hoạt động): Trạng thái hoạt động của công việc.
- CreatedAt (Ngày tạo): Ngày tạo bản ghi.
- UpdatedAt (Ngày sửa): Ngày cập nhật bản ghi.
- CreatedBy (Tạo bởi): Người tạo bản ghi.
- UpdatedBy (Sửa bởi): Người sửa bản ghi.
- CompanyId (Mã công ty): Mã công ty liên kết (khóa ngoại liên kết với bảng Company).

Bảng Resume chứa thông tin về các hồ sơ ứng tuyển của người dùng. Bảng này bao gồm các trường sau:

- Id (Mã): Mã định danh duy nhất của hồ sơ.
- Email (Email): Địa chỉ email của ứng viên.
- Url (Đường dẫn): Đường dẫn tới hồ sơ.
- Status (Trạng thái): Trạng thái của hồ sơ.
- CreatedAt (Ngày tạo): Ngày tạo bản ghi.
- UpdatedAt (Ngày sửa): Ngày cập nhật bản ghi.
- CreatedBy (Tạo bởi): Người tạo bản ghi.
- UpdatedBy (Sửa bởi): Người sửa bản ghi.
- UserId (Mã người dùng): Mã người dùng liên kết (khóa ngoại liên kết với bảng User).
- JobId (Mã công việc): Mã công việc liên kết (khóa ngoại liên kết với bảng Job).
- CompanyId (Mã công ty): Mã công ty liên kết (khóa ngoại liên kết với bảng Company).
- History (Lịch sử): Lịch sử trạng thái hồ sơ.

Bảng Subscriber chứa thông tin về các ứng viên theo dõi công việc. Bảng này bao gồm các trường sau:

- Id (Mã): Mã định danh duy nhất của người theo dõi.
- Email (Email): Địa chỉ email của người theo dõi.
- Name (Tên): Tên của người theo dõi.
- CreatedAt (Ngày tạo): Ngày tạo bản ghi.
- UpdatedAt (Ngày sửa): Ngày cập nhật bản ghi.
- CreatedBy (Tạo bởi): Người tạo bản ghi.
- UpdatedBy (Sửa bởi): Người sửa bản ghi.

## CHƯƠNG 4: KẾT QUẢ CÀI ĐẶT CHƯƠNG TRÌNH

### 4.1. Môi trường thực nghiệm và cài đặt chương trình

#### Phần cứng

CPU: Intel Core i5 hoặc cao hơn.

RAM: 8GB (khuyến nghị 16GB để xử lý mượt mà hơn).

Ổ cứng: SSD với dung lượng trống ít nhất 50GB.

#### Phần mềm

Hệ điều hành Windows 10/11 (64-bit).

Java Development Kit (JDK): Phiên bản 17

Cơ sở dữ liệu: MySQL phiên bản 8.0 hoặc cao hơn.

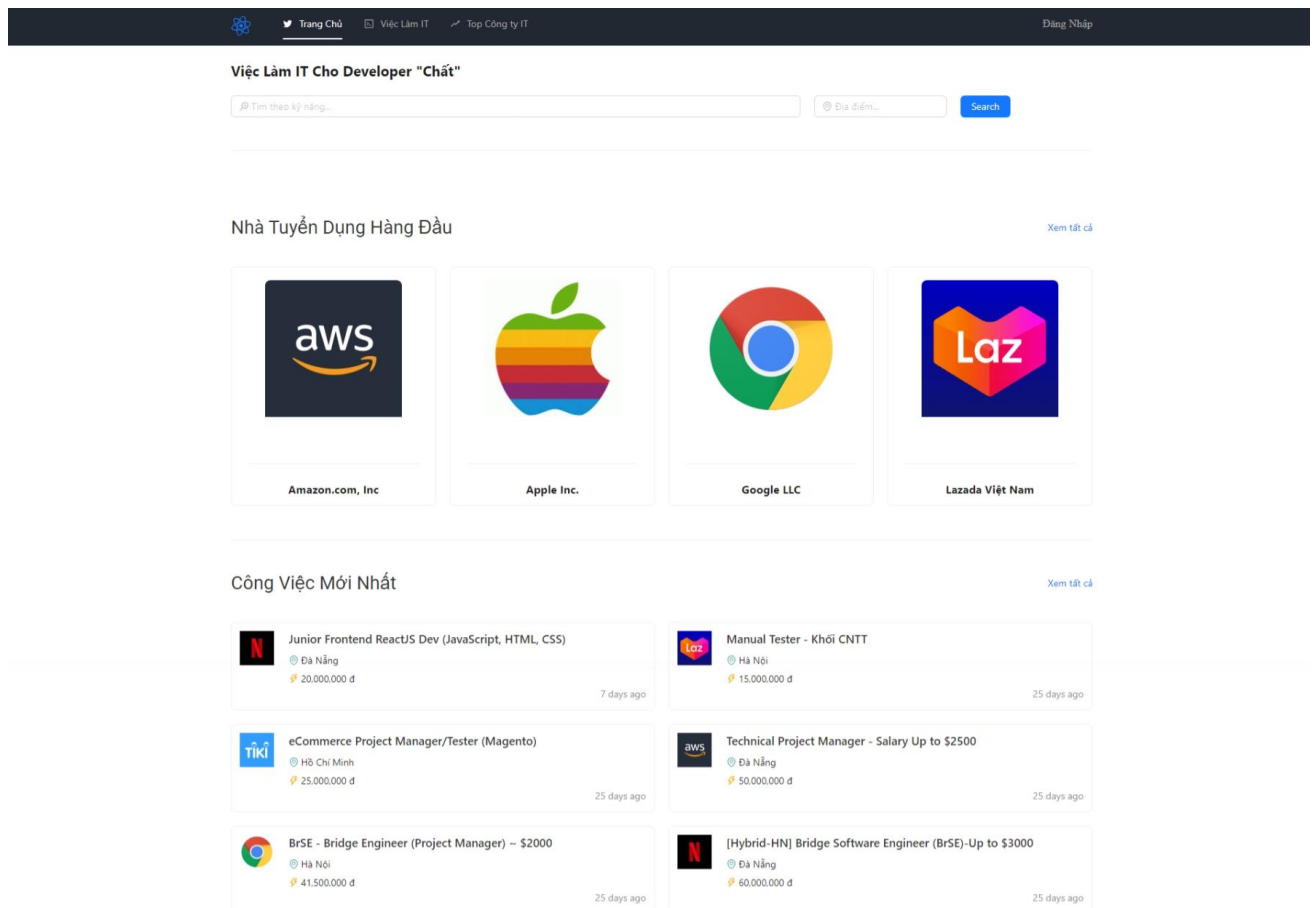
#### Công cụ hỗ trợ

Postman: Kiểm thử API.

Git: Quản lý mã nguồn

Node.js

### 4.2. Giao diện và chức năng dành cho người dùng chưa đăng ký tài khoản



Hình 4.1: Giao diện trang chủ

(người dùng khi chưa đăng nhập có thể xem thông tin các công ty và công việc đang được đăng tuyển )

Google LLC

Mountain View, California, Hoa Kỳ

Google LLC

**Google LLC** (n.d. "Công ty TNHH Google") là một công ty công nghệ đa quốc gia của Mỹ, chuyên về các dịch vụ và sản phẩm liên quan đến Internet, bao gồm các công nghệ quảng cáo trực tuyến, công cụ tìm kiếm, điện toán đám mây, phần mềm và phần cứng. Đây được coi là một trong những công ty công nghệ Big Four, cùng với Amazon, Apple và Facebook.

Google được thành lập vào năm 1998 bởi Larry Page và Sergey Brin trong khi họ là nghiên cứu sinh đã có bằng tiến sĩ tại Đại học Stanford ở California. Họ cùng nhau sở hữu khoảng 14% cổ phần và kiểm soát 56% quyền biểu quyết của cổ đông thông qua cổ phiếu ưu đãi. Họ đã hợp nhất Google thành một công ty tư nhân vào ngày 4 tháng 9 năm 1998. Một đợt chào bán công khai lần đầu (IPO) diễn ra vào ngày 19 tháng 8 năm 2004 và Google chuyển đến trụ sở chính tại Mountain View, California với tên Googleplex. Vào tháng 8 năm 2015, Google đã công bố kế hoạch tổ chức lại công ty với tư cách là một tập đoàn có tên là Alphabet Inc. Google là công ty con hàng đầu của Alphabet và sẽ tiếp tục là công ty ô tô và lợi ích Internet của Alphabet. Sundar Pichai được bổ nhiệm làm CEO của Google, thay thế Larry Page trở thành CEO của Alphabet.

Sự phát triển nhanh chóng của công ty kể từ khi thành lập đã kích hoạt một chuỗi các sản phẩm, mua lại để sáp nhập và hợp tác ngoài công cụ tìm kiếm cốt lõi của Google (Google Tìm kiếm). Nó cung cấp các dịch vụ được thiết kế cho công việc và năng suất (Google Docs, Google Sheets và Google Slides), email (Gmail/Inbox), lịch và quản lý thời gian (Lịch Google), lưu trữ đám mây (Google Drive), mạng xã hội (Google+), nhắn tin và trò chuyện video trực tiếp (Google Allo, Duo, Hangouts), dịch ngôn ngữ (Google Dịch), lập bản đồ và điều hướng (Google Maps, Waze, Google Earth, Chế độ xem phố), chia sẻ video (YouTube), ghi chú (Google Keep) và tổ chức và chỉnh sửa ảnh (Google Ảnh). Công ty dẫn đầu sự phát triển của hệ điều hành di động Android, trình duyệt web Google Chrome và Chrome OS, một hệ điều hành nhẹ dựa trên trình duyệt Chrome. Google đã ngày càng chuyển sang phần cứng: từ năm 2010 đến 2015, nó hợp tác với các nhà sản xuất điện tử lớn trong việc sản xuất các thiết bị Nexus của mình và đã phát hành nhiều sản phẩm phần cứng vào tháng 10 năm 2016, bao gồm điện thoại thông minh Google Pixel, loa thông minh Google Home, bộ định tuyến không dây Google Wifi và Daydream-tai nghe thực tế ảo. Google cũng đã thử nghiệm trở thành nhà cung cấp dịch vụ Internet (Google Fiber, Project Fi và Google Station).

Google.com là trang web được truy cập nhiều nhất trên thế giới. Một số dịch vụ khác của Google cũng nằm trong top 100 trang web được truy cập nhiều nhất trên thế giới, bao gồm YouTube và Blogger. Google là thương hiệu có giá trị nhất thế giới tính đến năm 2017 nhưng đã nhận được sự chỉ trích đáng kể liên quan đến các vấn đề như lo ngại về quyền riêng tư, tránh thuế, chống độc quyền, kiểm duyệt và trung lập trong tìm kiếm. Tuyên bố sử dụng của Google là "tuyệt đối không tin cậy của thế giới", và khẩu hiệu không chính thức là "Don't be evil" (Đừng trở nên xấu xa) cho đến khi cụm từ này được xóa khỏi quy tắc ứng xử của công ty vào khoảng tháng 5 năm 2018, nhưng lại được đưa vào trở lại ngày 31 tháng 7 năm 2018.<sup>[1][4]</sup>

Hình 4.2: Giao diện xem chi tiết thông tin của một công ty (người dùng có thể xem thông tin các công ty)

Trang ChủViệc Làm ITTop Công ty ITĐăng Nhập

Việc Làm IT Cho Developer "Chất"

🔍 Tìm theo kỹ năng...

📍 Địa điểm...

Search

Công Việc Mới Nhất

N

Junior Frontend ReactJS Dev (JavaScript, HTML, CSS)

📍 Đà Nẵng

💰 20.000.000 đ

7 days ago

L

Manual Tester - Khối CNTT

📍 Hà Nội

💰 15.000.000 đ

25 days ago

T

eCommerce Project Manager/Tester (Magento)

📍 Hồ Chí Minh

💰 25.000.000 đ

25 days ago

aws

Technical Project Manager - Salary Up to \$2500

📍 Đà Nẵng

💰 50.000.000 đ

25 days ago

Google

BrSE - Bridge Engineer (Project Manager) - \$2000

📍 Hà Nội

💰 41.500.000 đ

25 days ago

N

[Hybrid-HN] Bridge Software Engineer (BrSE)-Up to \$3000

📍 Đà Nẵng

💰 60.000.000 đ

25 days ago

<

1

2

>

46

*Hình 4.3: Giao diện việc làm  
(tại đây người dùng có thể xem thông tin các công việc đang được đăng tuyển)*

Trang ChủViệc Làm ITTop Công ty ITĐăng Nhập

Junior Frontend ReactJS Dev (JavaScript, HTML, CSS)

Apply Now

VUEJSANGULAR

20,000,000 đ

Đà Nẵng

7 days ago

Netflix Inc

Mô Tả Công Việc

Xây dựng phần hệ front-end (giao diện người dùng) các tính năng được yêu cầu từ team phát triển sản phẩm cùng như từ người dùng, cụ thể:

- Tham gia vào việc phát triển sản phẩm của công ty: lập trình front-end, chuyển các file thiết kế (PSD, Figma, Sketch, Invision,...) sang HTML, CSS, IOS, Android.
- Bảo trì, nâng cấp cho front-end sản phẩm của công ty.
- Nghiên cứu và áp dụng các công nghệ mới để tối đa hóa hiệu quả phát triển sản phẩm.
- Phối hợp và hỗ trợ với team back-end một cách chủ động và chặt chẽ để nâng cao trải nghiệm người dùng trên từng điểm tiếp xúc.

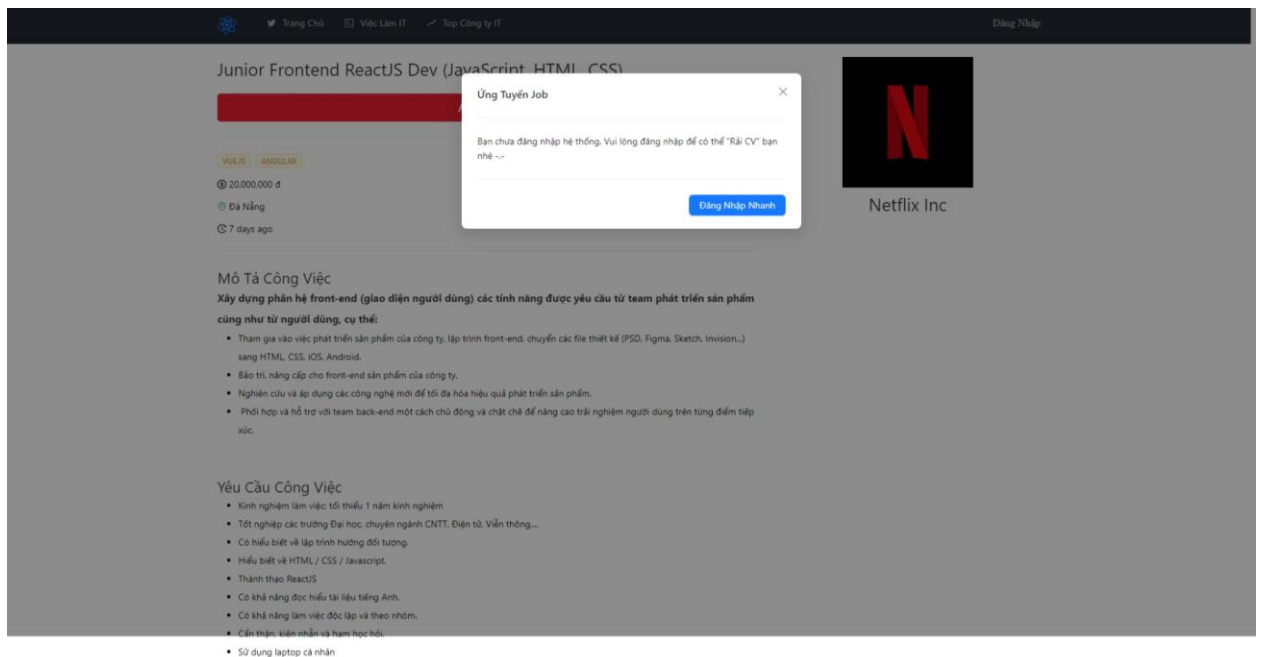
Yêu Cầu Công Việc

- Kinh nghiệm làm việc tối thiểu 1 năm kinh nghiệm
- Tốt nghiệp các trường Đại học, chuyên ngành CNTT, Điện tử, Viễn thông...
- Có hiểu biết về lập trình hướng đối tượng.
- Hiểu biết về HTML / CSS / Javascript.
- Thành thạo ReactJS
- Có khả năng đọc hiểu tài liệu tiếng Anh.
- Có khả năng làm việc độc lập và theo nhóm.
- Cẩn thận, kiên nhẫn và ham học hỏi.
- Sử dụng laptop cá nhân

Tại Sao Bạn Sẽ Yêu Thích Làm Việc Tại Đây

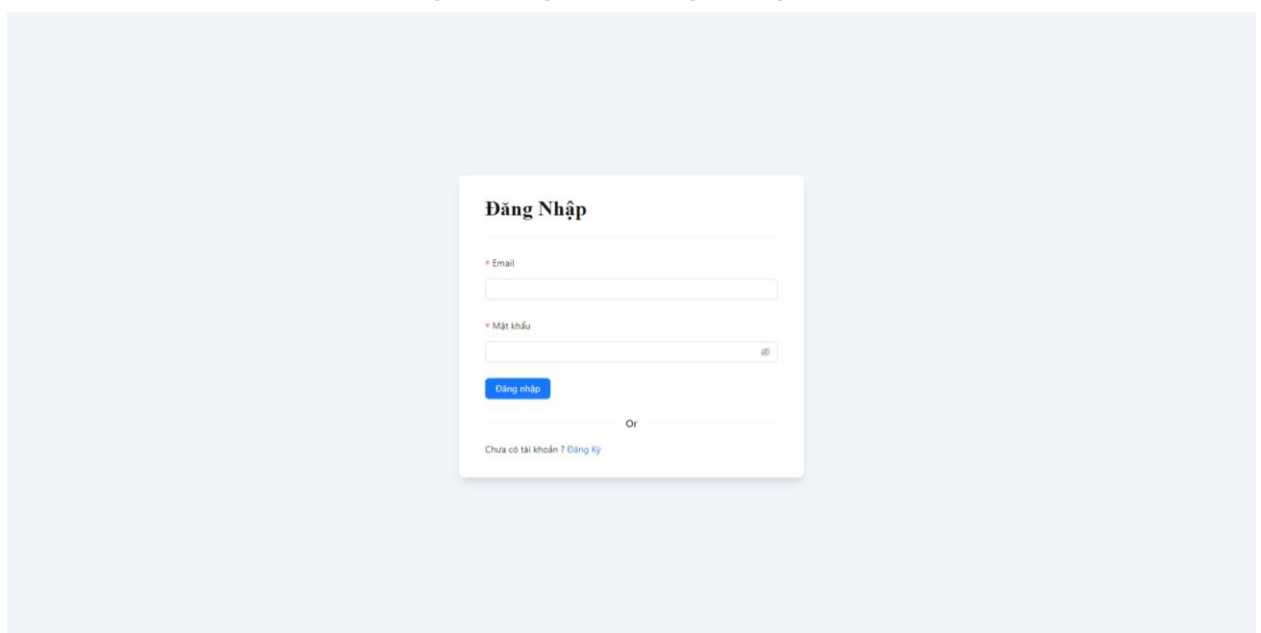
- Chế độ lương - thưởng: Mức lương cạnh tranh & hấp dẫn, Lương tháng 13.
- Đánh giá năng lực: xem xét điều chỉnh lương 1 lần 1 năm.
- Thời gian làm việc: Từ thứ Hai đến thứ Sáu: 09h00 - 18h00.
- Company trip, Team Building, Year end party, Party team hàng tháng đi ăn nhậu với nhau (được công ty hỗ trợ)
- Mỗi tháng có meeting tổng kết kinh doanh, ăn uống, giao lưu với nhân viên mới.
- Tham gia các nhóm thể thao của công ty: Đá bóng, cầu lông.
- Môi trường làm việc mở, sẵn sàng đón nhận ý kiến từ tất cả nhân viên, thân thiện, năng động, đa dạng và thăng tiến.

*Hình 4.4: Giao diện của một công việc đang được đăng tuyển  
(người dùng vào trang này để đọc mô tả công việc và ứng tuyển vào công việc)*



*Hình 4.5: Giao diện khi ứng tuyển một công việc  
(nếu người dùng chưa đăng nhập, hệ thống sẽ yêu cầu người dùng đăng nhập để  
được rải CV )*

### 4.3. Giao diện và chức năng khi người dùng đăng nhập



*Hình 4.6: Giao diện form đăng nhập  
(Người dùng đăng nhập bằng tài khoản các nhân để truy cập trang web )*

**Đăng Ký Tài Khoản**

\* Họ tên

\* Email

\* Mật khẩu

\* Tuổi

\* Giới tính

\* Địa chỉ

Đăng ký

Or

Đã có tài khoản? [Đăng Nhập](#)

*Hình 4.7: Giao diện đăng ký  
(đăng ký một tài khoản nếu người dùng chưa có tài khoản)*

**Ứng Tuyển Job**

Bạn đang ứng tuyển công việc **Junior Frontend ReactJS Dev (JavaScript, HTML, CSS)** tại **Netflix Inc**

Email

Upload file CV

Tải lên CV của bạn (Hỗ trợ \*.doc, \*.docx, \*.pdf, and < 5MB)

RÚ.pdf

[Rải CV Nào](#)

**Junior Frontend ReactJS Dev (JavaScript, HTML, CSS)**

20.000.000 đ

Đã Nẵng

7 days ago

**Mô Tả Công Việc**

Xây dựng phần hệ front-end (giao diện người dùng) như từ người dùng, cụ thể:

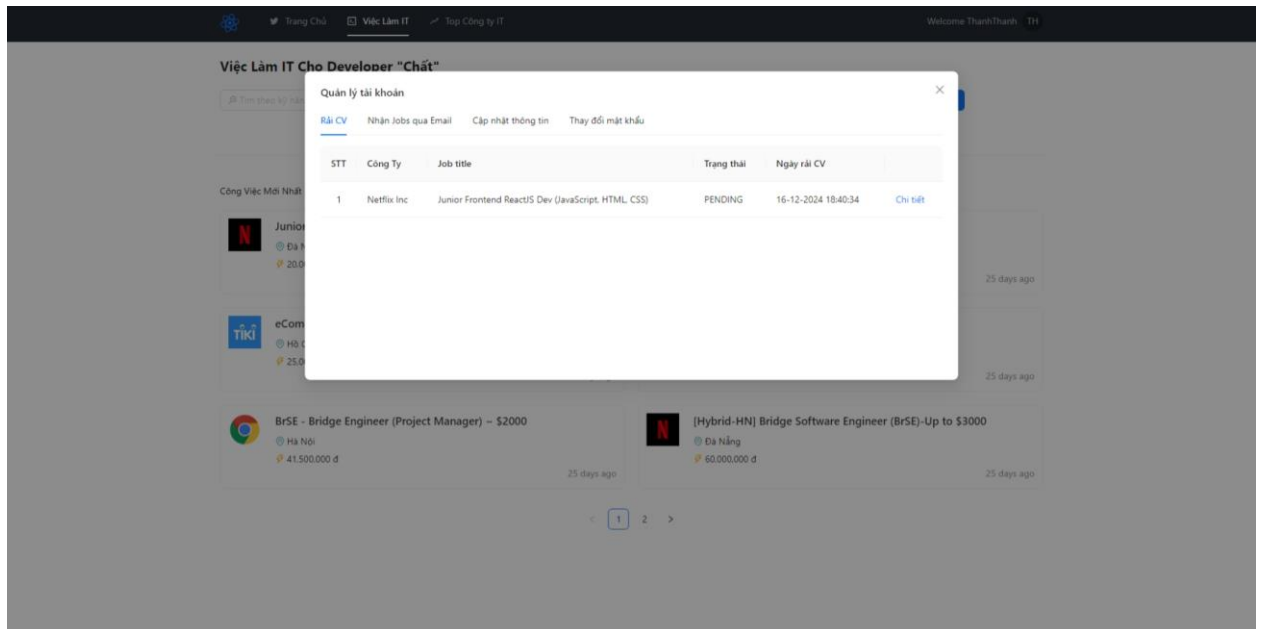
- Tham gia vào việc phát triển sản phẩm của công ty: lập trình web bằng HTML, CSS, iOS, Android.
- Bảo trì, nâng cấp cho front-end sản phẩm của công ty.
- Nghiên cứu và áp dụng các công nghệ mới để tối đa hóa hiệu quả phát triển sản phẩm.
- Phối hợp và hỗ trợ với team back-end một cách chủ động và chặt chẽ để nâng cao trải nghiệm người dùng trên từng điểm tiếp xúc.

**Yêu Cầu Công Việc**

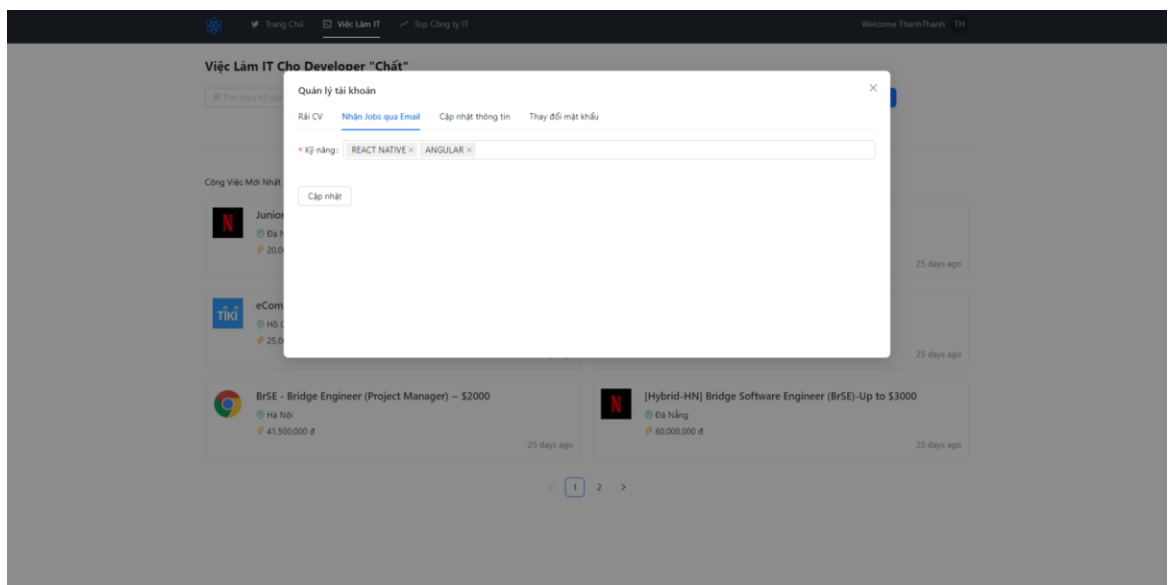
- Kinh nghiệm làm việc: tối thiểu 1 năm kinh nghiệm
- Tốt nghiệp các trường Đại học, chuyên ngành CNTT, Điện tử, Viễn thông...
- Có hiểu biết về lập trình hướng đối tượng.
- Hiểu biết về HTML / CSS / Javascript.
- Thành thạo ReactJS
- Có khả năng đọc hiểu tài liệu tiếng Anh.
- Có khả năng làm việc độc lập và theo nhóm.
- Cần thiết: trình độ ngoại ngữ tốt.**
- Sử dụng laptop cá nhân



*Hình 4.8: Giao diện khi người dùng ứng tuyển một công việc  
(Người dùng có thể rải CV vào một công việc đang được đăng tuyển sau khi  
đã đăng nhập)*



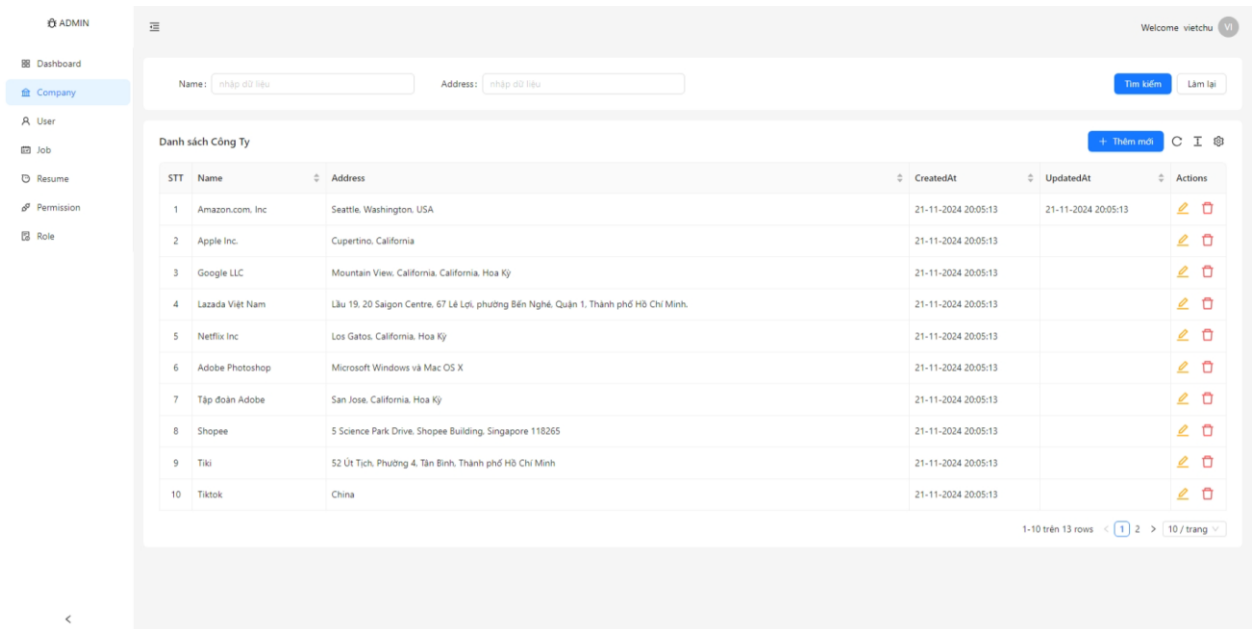
*Hình 4.9: Giao diện quản lí tài khoản của người dùng  
(tại đây người dùng có thể xem lại lịch sử của các công việc đã ứng tuyển)*



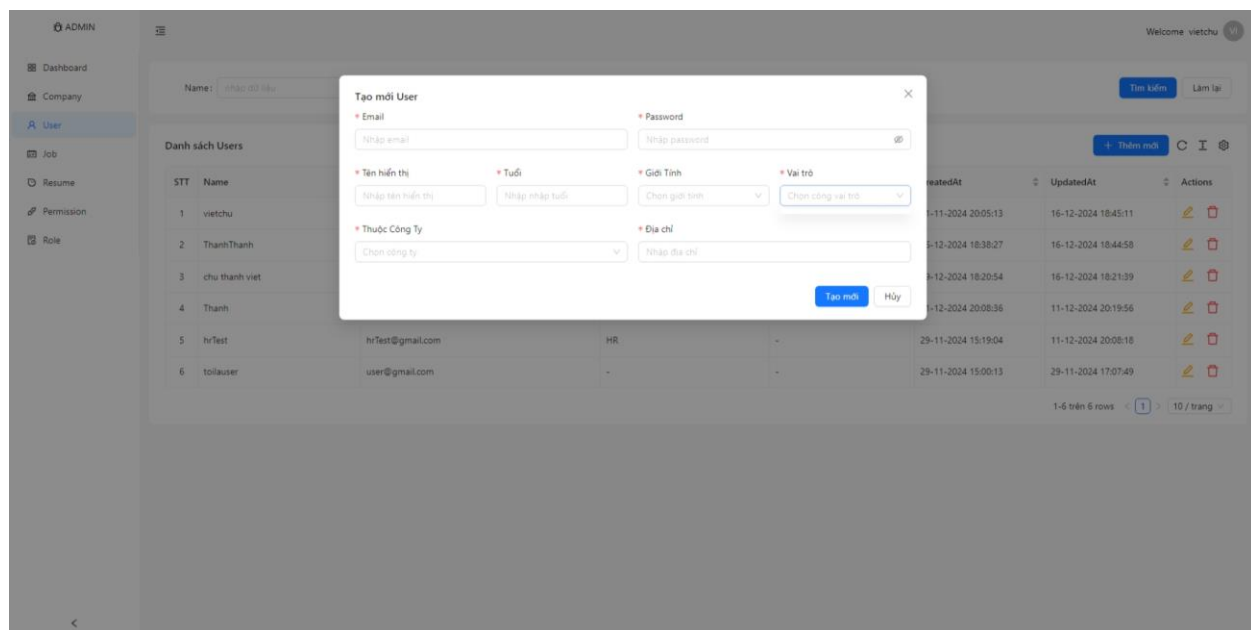
*Hình 4.10: Giao diện quản lí tài khoản của người dùng*

(người dùng có thể nhận email các công việc đang được đăng tuyển theo kỹ năng)

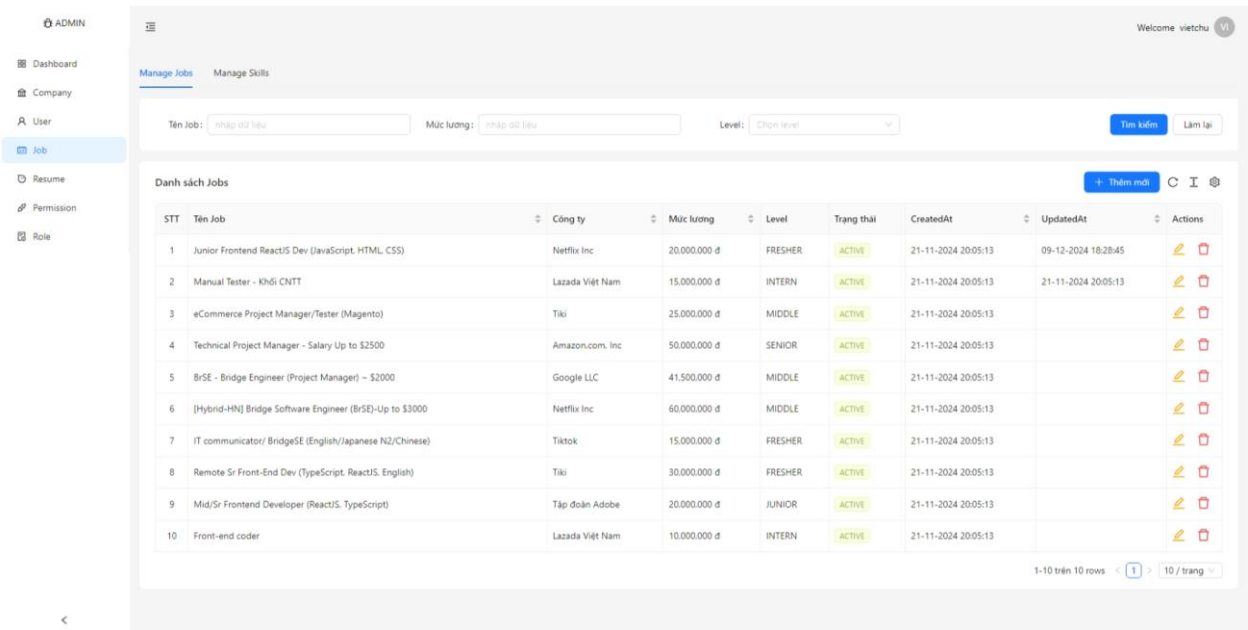
#### 4.4. Giao diện và chức năng cho người quản trị viên của hệ thống



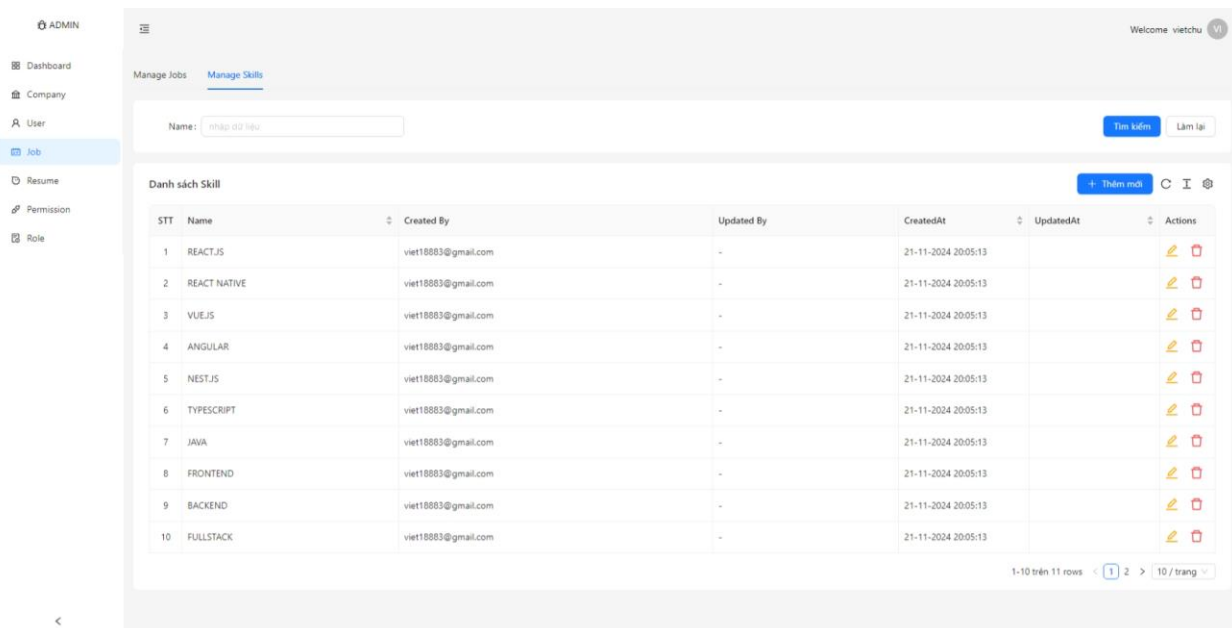
Hình 4.11: Giao diện trang quản trị công ty  
(quản trị viên có thể thêm sửa xóa thông tin công ty)



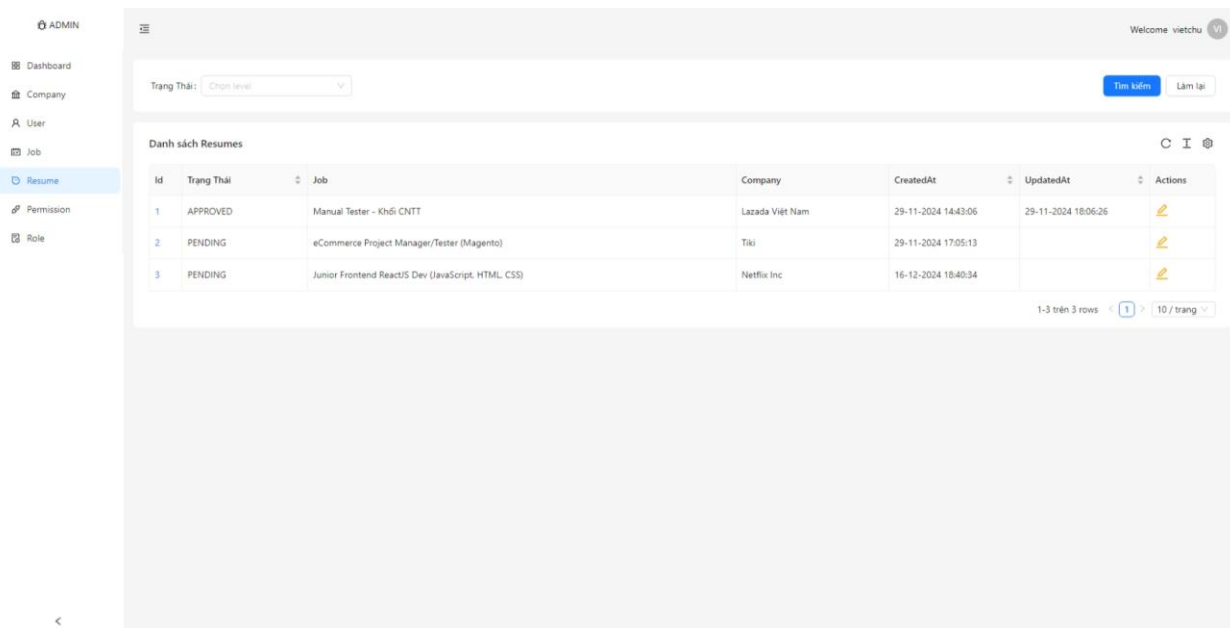
Hình 4.12: Giao diện trang quản trị người dùng  
(quản trị viên có thể thêm, sửa, xóa và sét vai trò và quyền hạn cho một người dùng)



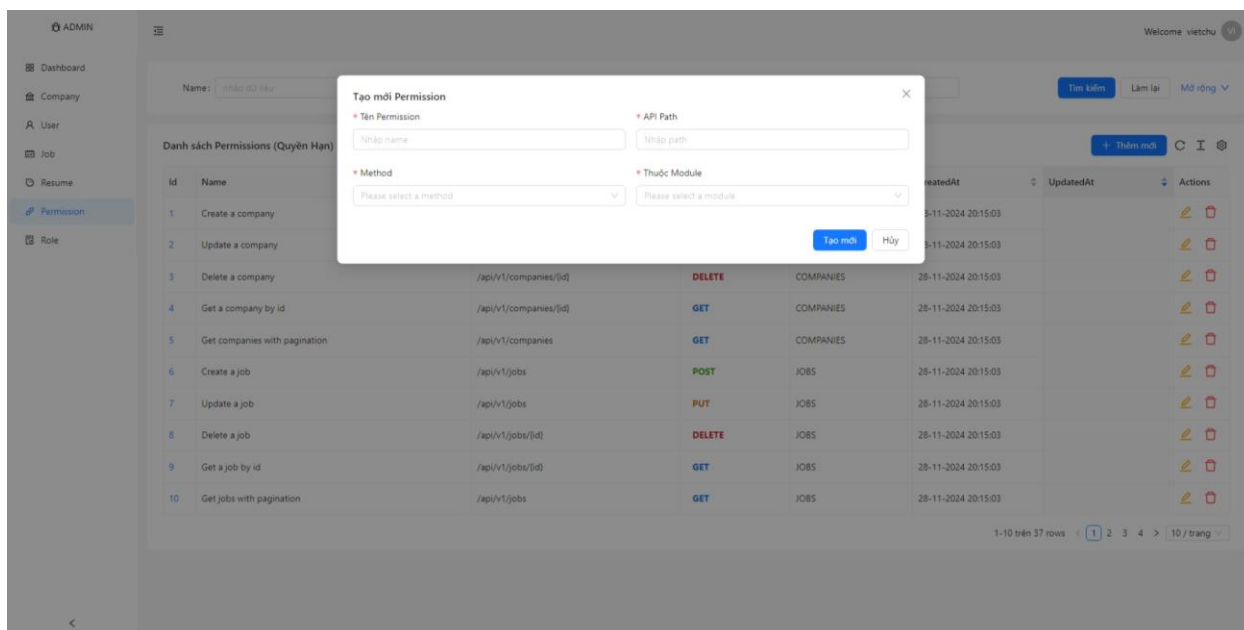
Hình 4.13: Giao diện trang quản trị công việc  
(quản trị viên có thể đăng, sửa, xóa một công việc)



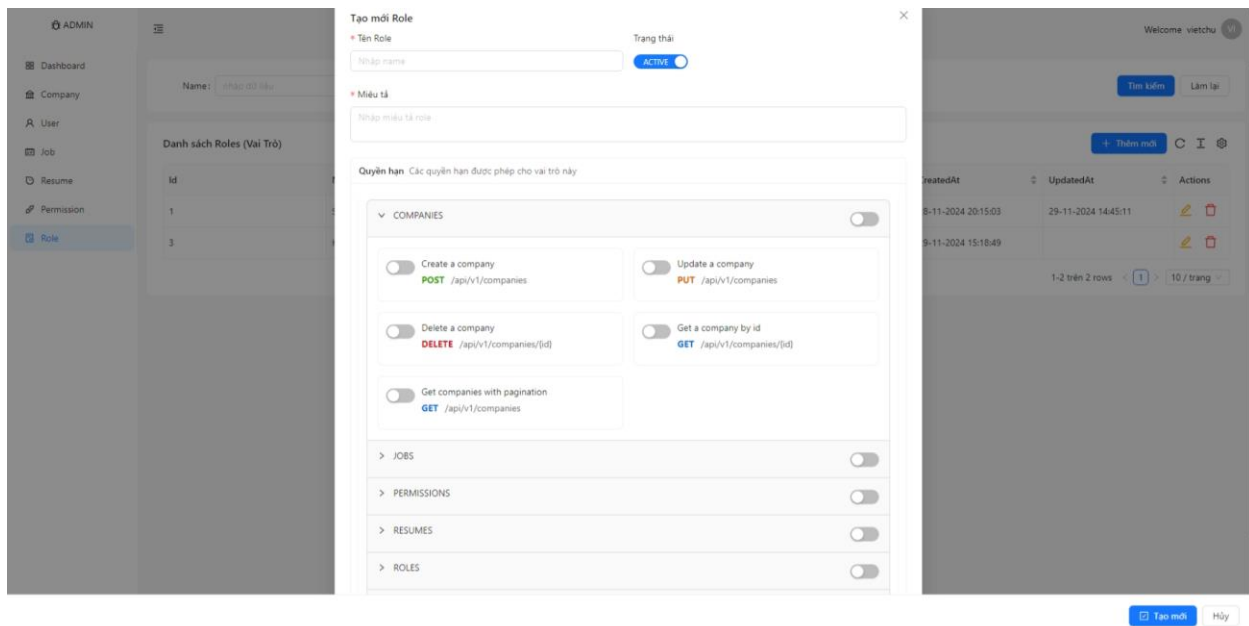
Hình 4.14: Giao diện trang quản trị các kỹ năng cho công việc  
(quản trị viên có thể đăng, sửa, xóa một kỹ năng)



Hình 4.15: Giao diện trang quản trị các hồ sơ đã được ứng tuyển (quản trị viên có thể xem chi tiết hồ sơ, cập nhật trạng thái cho hồ sơ)



Hình 4.16: Giao diện trang quản trị các quyền hạn của hệ thống (quản trị viên có thể tạo một quyền hạn mới, sửa quyền hạn, xóa đi những quyền hạn không phù hợp)



*Hình 4.17: Giao diện trang quản trị vai trò  
(quản trị viên có thể tạo một vai trò mới, chỉnh sửa vai trò, xóa đi những vai  
trò không phù hợp)*

## **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **5.1. Ưu điểm**

Sau một thời gian tìm hiểu và nghiên cứu, em đã xây dựng được một Website quản lý việc làm với các chức năng cơ bản. Website được thiết kế với giao diện thân thiện, trực quan, giúp người dùng dễ dàng truy cập và tìm kiếm thông tin việc làm mà không cần đăng nhập. Đồng thời, giao diện quản trị cung cấp đầy đủ các chức năng quản lý như theo dõi thông tin người dùng, vai trò, quyền hạn, công ty, và danh sách công việc. Điều này giúp quản trị viên thao tác nhanh chóng, thuận tiện.

Website hỗ trợ nhiều tính năng cơ bản, đáp ứng tốt nhu cầu thị trường tìm việc làm trực tuyến. Người dùng giờ đây có thể tìm kiếm việc làm, nộp hồ sơ ứng tuyển và liên hệ với nhà tuyển dụng một cách nhanh chóng, trong khi các nhà tuyển dụng cũng có thể quản lý thông tin của công ty, đăng tuyển công việc và xem xét hồ sơ các ứng viên. Các tính năng như phân loại việc làm theo kỹ năng mang đến sự tiện lợi và tối ưu trải nghiệm cho cả người tìm việc lẫn nhà tuyển dụng.

Qua việc tự tìm hiểu và xây dựng được website quản lý việc làm, cùng với sự hướng dẫn chỉ bảo của thầy giáo hướng dẫn. Em cũng đã biết thêm được nhiều kiến thức mới hơn như công nghệ lập trình web như Spring Boot, MySQL và React, dự án đảm bảo tính ổn định, bảo mật và dễ dàng mở rộng trong tương lai. Hiểu được quy trình nghiệp vụ khi làm một website cung cấp dịch vụ, đưa thông tin trên mạng.

### **5.2. Nhược điểm**

Tuy nhiên, một trong những nhược điểm chính của dự án chính là hệ thống chỉ tập trung vào các chức năng cơ bản mà chưa tích hợp các tính năng nâng cao như tự động gợi ý việc làm, phân tích dữ liệu người dùng hoặc tích hợp trí tuệ nhân tạo để tối ưu hóa trải nghiệm của cả nhà tuyển dụng và người tìm việc. Điều này làm giảm đi tính cạnh tranh đối với các nền tảng lớn hiện nay trên thị trường lao động trực tuyến.

### **5.3. Hướng phát triển**

Sau khi nghiên cứu và thực hiện đề tài, em đã vạch ra được hướng phát triển tiếp theo của đề tài như sau:

- Nâng cao trải nghiệm người dùng bằng các sử dụng những thiết kế hiện đại, cải thiện bố cục, màu sắc và font chữ.
- Tích hợp thêm các tính năng nâng cao như sử dụng AI/ML để phân tích hồ sơ người dùng để gợi ý công việc phù hợp cho họ.
- Tích hợp hệ thống đánh giá và xếp hạng công việc.
- Sử dụng Websocket và Firebase để gửi thông báo.
- Tăng cường bảo mật cho trang web.

## TÀI LIỆU THAM KHẢO

- [1] Cao Tùng Anh; (2019). *Cơ sở dữ liệu và quản trị cơ sở dữ liệu*, Hutech University, Tp.HCM;
- [2] Mai Ngọc Thu, Nguyễn Đình Ánh. (2019). *Lập trình web*. Tp.HCM Hutech;
- [3] Nguyễn Đình Ánh, Nguyễn Huy Cường, Trần Đăng Khoa, Phạm Hữu Kỳ. 2023. *Thực hành Lập trình Web*. Hutech University, Tp.HCM;
- [4] <https://gpcoder.com/6282-tong-quan-ve-jpa-java-persistence-api/>
- [5] <https://www.geeksforgeeks.org/introduction-to-spring-boot/>
- [6] <https://dev.mysql.com/doc/workbench/en/wb-intro.html>
- [7] <https://simplilearn.com/tutorials/mysql-tutorial/mysql-workbench>
- [8] <https://www.postman.com/api-platform/>
- [9] <https://datascientest.com/en/postman-what-is-it-how-does-it-work>
- [10] <https://dev.to/weder96/spring-boot-everything-you-need-to-know-and-what-nobody-told-you-o4j>
- [11] <https://viblo.asia/p/spring-boot-la-gi-Qpmlezzn5rd>
- [12] <https://www.oracle.com/database/spring-boot/>
- [13] <https://techmaster.vn/posts/36698/loi-ich-va-han-che-cua-viec-su-dungspring-boot>