**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**Đề tài: Jobtify**

***Sinh viên thực hiện***

1. Nguyễn Văn Giàu – 20053331

# MỤC LỤC

[MỤC LỤC 1](#_Toc185181313)

[DANH MỤC CÁC HÌNH VẼ 3](#_Toc185181314)

[DANH MỤC CÁC BẢNG BIỂU 4](#_Toc185181315)

[CHƯƠNG 1 : GIỚI THIỆU 5](#_Toc185181316)

[1.1 Tổng quan 5](#_Toc185181317)

[1.2 Mục tiêu đề tài 5](#_Toc185181318)

[1.3 Phạm vi đề tài 5](#_Toc185181319)

[1.4 Mô tả yêu cầu chức năng 6](#_Toc185181320)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 7](#_Toc185181321)

[2.1 Java JDK 21 7](#_Toc185181322)

[2.2 Kafka 8](#_Toc185181323)

[2.3 Redis 10](#_Toc185181324)

[2.4 Kiến trúc microservices 11](#_Toc185181325)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 12](#_Toc185181326)

[3.1 Kiến trúc 12](#_Toc185181327)

[3.2 Phân tích yêu cầu bằng UML 12](#_Toc185181328)

[3.2.1 Usecase tổng quát 12](#_Toc185181329)

[3.2.2 Danh sách tác nhân và mô tả 12](#_Toc185181330)

[3.2.3 Danh sách các tình huống hoạt động (Use cases) 13](#_Toc185181331)

[3.2.4 Tình huống hoạt động 14](#_Toc185181332)

[3.2.4.1 UC01\_Đăng ký 14](#_Toc185181333)

[3.2.4.2 UC02\_Đăng nhập 15](#_Toc185181334)

[3.3 Class diagram 17](#_Toc185181335)

[3.3.1 Entity diagram 18](#_Toc185181336)

[3.3.2 Database diagram 19](#_Toc185181337)

[3.4 Deployment diagram 22](#_Toc185181338)

[CHƯƠNG 4 : HIỆN THỰC 23](#_Toc185181339)

[4.1 Cấu hình phần cứng, phần mềm 23](#_Toc185181340)

[4.2 Giao diện của hệ thống 23](#_Toc185181341)

[4.3 Kế hoạch và hiện thực kiểm thử hệ thống 25](#_Toc185181342)

[4.3.1 Kế hoạch kiểm thử 25](#_Toc185181343)

[4.3.2 Kiểm thử hệ thống 25](#_Toc185181344)

[CHƯƠNG 5 : KẾT LUẬN 28](#_Toc185181345)

[5.1 Kết quả đạt được 28](#_Toc185181346)

[5.2 Hướng phát triển 28](#_Toc185181347)

[TÀI LIỆU THAM KHẢO 29](#_Toc185181348)

# DANH MỤC CÁC HÌNH VẼ

[*11. Usecase tổng quát* 13](#_Toc185172814)

# DANH MỤC CÁC BẢNG BIỂU

[*Bảng 3.1 Danh sách tác nhân và mô tả* 14](#_Toc185172863)

[*Bảng 3.2 Danh sách các tình huống trong hệ thống* 14](#_Toc185172864)

[*4.1 Phần cứng thực hiện kiểm thử* 26](#_Toc185172865)

[*4.2 Phần mềm thực hiện kiểm thử* 26](#_Toc185172866)

[*5.1 Kịch bản kiểm thử.* 26](#_Toc185172867)

[*6. Bảng kết quả kiểm thử.* 27](#_Toc185172868)

# : GIỚI THIỆU

## Tổng quan

## Mục tiêu đề tài

**Hỗ trợ công ty đăng tin tuyển dụng**: Các công ty có thể dễ dàng đăng tải các tin tuyển dụng, bao gồm yêu cầu về kỹ năng, giúp ứng viên biết rõ các yêu cầu của công việc.

**Gợi ý công việc cho ứng viên:** Dựa trên kỹ năng và hồ sơ của ứng viên, hệ thống tự động gợi ý các công việc phù hợp, giúp ứng viên tiết kiệm thời gian tìm kiếm và tăng khả năng tìm được việc làm phù hợp.

**Giúp công ty tìm kiếm ứng viên phù hợp:** Công ty có thể tìm kiếm các ứng viên có kỹ năng đáp ứng yêu cầu công việc, gửi lời mời ứng tuyển, giúp việc tuyển dụng nhanh chóng và hiệu quả hơn.

**Đề xuất kỹ năng bổ sung cho ứng viên:** Hệ thống sẽ phân tích kỹ năng của ứng viên và đề xuất các kỹ năng mà ứng viên có thể học thêm, nhằm cải thiện cơ hội việc làm và phát triển nghề nghiệp.

## Phạm vi đề tài

Ứng dụng được sử dụng bởi quản trị viên hệ thống, công ty tuyển dụng và ứng viên tìm việc.

Ứng dụng hỗ trợ cho nền tảng web mobile và web app.

Mục dích ứng dụng:

+ Quản trị viên: có thể quản lý các kỹ năng cho một hệ thống.

+ Ứng viên: tiện lợi trong việc tìm kiếm công việc phù hợp với bản thân.

+ Công ty: tìm được ứng viên phù hợp với vị trí mô tả từng công việc.

Đối tượng sử dụng tài liệu:

+ Người phát triển phần mềm.

+ Người bảo trì hệ thống phần mềm.

+ Người kiểm thử phần mềm.

## Mô tả yêu cầu chức năng

**Yêu cầu chức năng:**

Công ty đăng nhập vào hệ thống.

Công ty điền thông tin mô tả công việc và các kỹ năng yêu cầu.

Hệ thống lưu thông tin và hiển thị tin tuyển dụng cho ứng viên.

Công ty có thể chỉnh sửa hoặc xóa tin tuyển dụng nếu cần.

Ứng viên đăng nhập vào hệ thống.

Hệ thống xác định kỹ năng của ứng viên dựa trên hồ sơ và các thông tin đã cung cấp.

Hệ thống tự động gợi ý các công việc phù hợp với kỹ năng hiện có của ứng viên.

Ứng viên có thể duyệt qua danh sách các công việc gợi ý và chọn ứng tuyển.

**Yêu cầu phi chức năng:**

Đảm bảo hoạt động được trên mọi thiết bị.

Thời gian cập nhật dữ liệu dưới 2 giây.

Thời gian hiển thị các danh sách phải dưới 5 giây.

Các thông tin quan trọng phải được hiển thị to và rõ để nhận biết.

# : CƠ SỞ LÝ THUYẾT

## Java JDK 21

**JDK 21** (Java Development Kit 21) là phiên bản mới nhất của bộ công cụ phát triển Java, được Oracle phát hành vào tháng 9 năm 2023. Đây là một phiên bản có nhiều cải tiến quan trọng, mang lại hiệu suất tốt hơn, bảo mật cao hơn và hỗ trợ các tính năng mới giúp lập trình viên phát triển ứng dụng hiệu quả hơn.

Một số tính năng đáng chú ý của **JDK 21** bao gồm:

1. **Pattern Matching cho switch expressions**: JDK 21 mở rộng khả năng của pattern matching trong Java, giúp đơn giản hóa mã nguồn và cải thiện tính đọc hiểu. Pattern matching giờ đây có thể sử dụng trong các câu lệnh switch, giúp giảm sự phức tạp và tối ưu hóa quy trình kiểm tra kiểu dữ liệu.
2. **Record Patterns**: JDK 21 tiếp tục cải tiến record types (kiểu dữ liệu kiểu bản ghi), cho phép lập trình viên dễ dàng thao tác với dữ liệu dạng bản ghi mà không cần phải viết mã boilerplate.
3. **Virtual Threads (Project Loom)**: Đây là tính năng cải tiến quan trọng giúp Java hỗ trợ các luồng ảo (virtual threads), giúp giảm thiểu overhead khi tạo và quản lý các luồng. Điều này sẽ giúp các ứng dụng Java xử lý các tác vụ đồng thời hiệu quả hơn, đặc biệt trong các ứng dụng có tính chất I/O nặng.
4. **JEP 443: Structured Concurrency (Thực thi đồng thời có cấu trúc)**: JDK 21 mang đến khả năng lập trình đồng thời dễ dàng hơn bằng cách cung cấp API mới cho phép người lập trình làm việc với các luồng đồng thời một cách an toàn và hiệu quả hơn.
5. **Nâng cấp hiệu suất và bảo mật:** JDK 21 tiếp tục tối ưu hiệu suất của Java, giúp ứng dụng chạy nhanh hơn và bảo mật tốt hơn nhờ các cải tiến trong JVM (Java Virtual Machine) và bộ công cụ đi kèm.



*Hình 1 JDK 21*

## Kafka



**Apache Kafka** là một hệ thống phân tán, mã nguồn mở, được thiết kế để xử lý và truyền tải một lượng lớn dữ liệu theo thời gian thực. Kafka hoạt động như một **message broker**, cho phép các dịch vụ trong kiến trúc microservices giao tiếp với nhau thông qua các thông điệp (messages). Dữ liệu trong Kafka được tổ chức thành các **topics**, nơi các producer (người sản xuất dữ liệu) gửi dữ liệu và các consumer (người tiêu thụ dữ liệu) đọc từ đó. Kafka có thể xử lý hàng triệu sự kiện mỗi giây, giúp đảm bảo tính khả dụng và độ bền của thông tin.

Kafka là lựa chọn lý tưởng cho các ứng dụng cần xử lý và truyền tải thông tin theo thời gian thực, chẳng hạn như các hệ thống theo dõi, phân tích dữ liệu hoặc các ứng dụng có yêu cầu khối lượng lớn thông điệp bất đồng bộ. Ngoài ra, Kafka cũng hỗ trợ tính năng **fault tolerance** cao, đảm bảo rằng dữ liệu không bị mất trong quá trình truyền tải, ngay cả khi một số thành phần hệ thống gặp sự cố.

## Redis là gì? Ưu điểm của nó và ứng dụng | TopDevRedis

**Redis** là một cơ sở dữ liệu **key-value** nhanh và phổ biến, được sử dụng chủ yếu để lưu trữ dữ liệu trong bộ nhớ (in-memory) với tốc độ truy xuất rất nhanh. Redis hỗ trợ nhiều kiểu dữ liệu, bao gồm strings, lists, sets, và hashes, giúp tối ưu hóa các tác vụ như caching, session management, và message brokering. Redis đặc biệt hiệu quả trong các ứng dụng cần truy xuất dữ liệu nhanh chóng và giảm tải cho các cơ sở dữ liệu chính.

Trong kiến trúc microservices, Redis thường được sử dụng để **cache** dữ liệu tạm thời, giúp giảm độ trễ khi truy vấn các dữ liệu phổ biến, từ đó nâng cao hiệu suất hệ thống. Ngoài ra, Redis cũng hỗ trợ tính năng **queueing**, cho phép các microservices xử lý các tác vụ bất đồng bộ qua các hàng đợi (queues). Redis có thể giúp tối ưu hóa thời gian phản hồi của hệ thống và cải thiện khả năng mở rộng, đặc biệt trong các môi trường có yêu cầu về tốc độ và khả năng chịu tải cao.

## Microservices là gì? Speed up Microservices 1: Tác dụng phụ và một số chiến lược cơ bản | TopDevKiến trúc microservices

Kiến trúc **Microservices** là một phương pháp thiết kế phần mềm trong đó một ứng dụng được chia thành nhiều dịch vụ nhỏ, độc lập, mỗi dịch vụ thực hiện một chức năng cụ thể của hệ thống. Các dịch vụ này giao tiếp với nhau qua các API hoặc message brokers, giúp hệ thống trở nên linh hoạt và dễ mở rộng. Mỗi microservice có thể được triển khai và bảo trì độc lập, sử dụng các công nghệ và cơ sở dữ liệu riêng biệt, phù hợp với yêu cầu của từng chức năng.

Kiến trúc microservices mang lại nhiều lợi ích, bao gồm khả năng mở rộng linh hoạt, tính khả dụng cao và dễ dàng bảo trì. Tuy nhiên, nó cũng đối mặt với các thách thức về quản lý, giao tiếp giữa các dịch vụ, và kiểm thử hệ thống. Việc triển khai các dịch vụ nhỏ này yêu cầu các công cụ như **API Gateway, Service Discovery,** và **Event/Message Brokers** để đảm bảo tính đồng bộ và hiệu quả trong toàn bộ hệ thống.

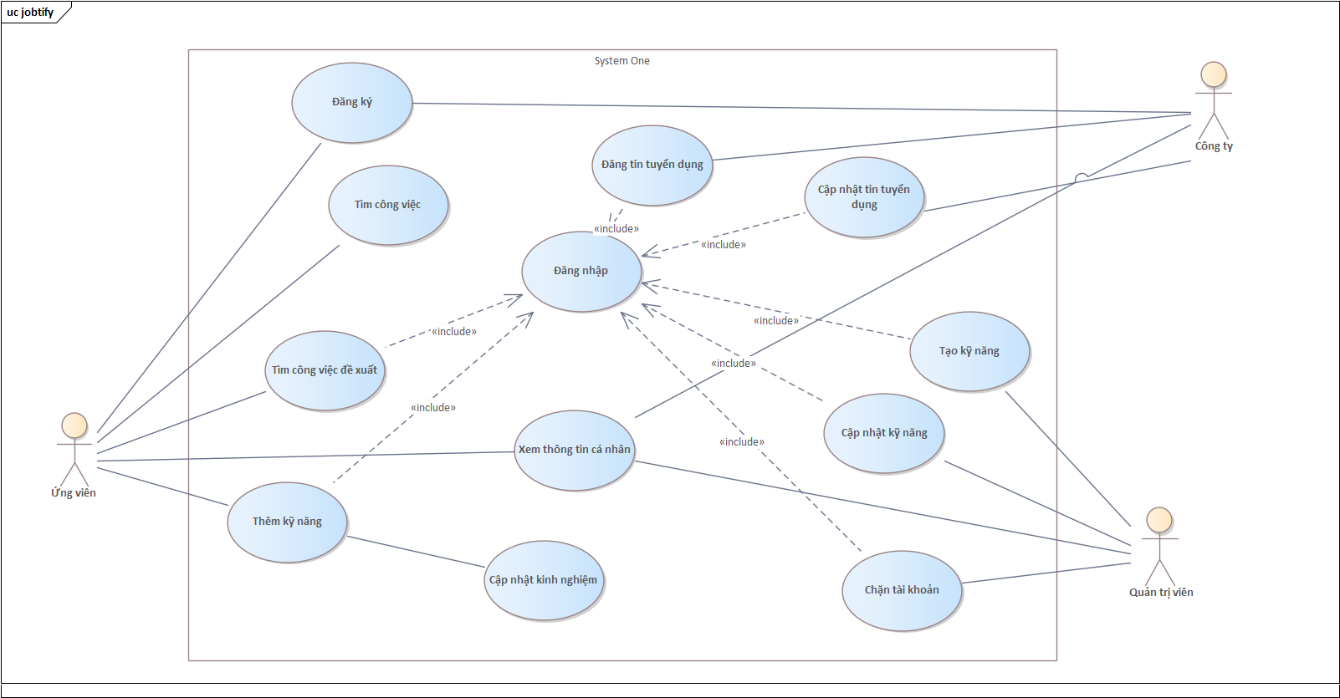
Với sự phát triển của công nghệ container (Docker) và orchestration tools (Kubernetes), microservices đã trở thành một lựa chọn phổ biến cho các hệ thống phân tán và các ứng dụng yêu cầu tính linh hoạt và mở rộng. Tuy nhiên, việc quản lý và duy trì các dịch vụ này vẫn cần sự đầu tư và giải pháp hợp lý.

# : PHÂN TÍCH VÀ THIẾT KẾ

## Kiến trúc

## Phân tích yêu cầu bằng UML

### Usecase tổng quát



*11. Usecase tổng quát*

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Quản trị viên | Thực hiện quản lý các kỹ năng cơ bản và người dùng hệ thống. |
| Ứng viên | Xem thông tin công việc, công việc gợi ý được các công ty đăng bài tuyển dụng, quản lý thông tin kỹ năng cá nhân, địa chỉ. |
| Công ty | Đăng bài tuyển dụng, các yêu cầu kỹ năng cho một công việc. |

*Bảng 3.1 Danh sách tác nhân và mô tả*

### Danh sách các tình huống hoạt động (Use cases)

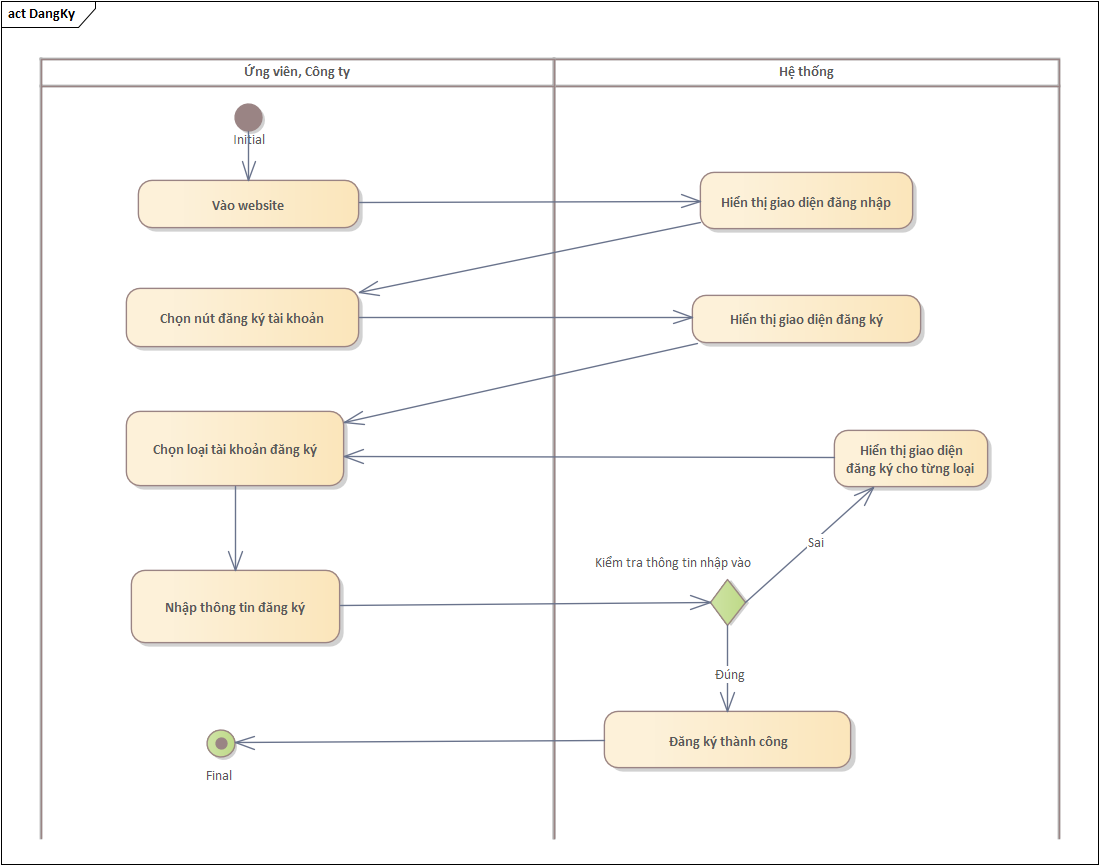
|  |  |
| --- | --- |
| ID | Tên Use case |
| UC01 | Đăng ký |
| UC02 | Đăng nhập |
| UC03 | Quên mật khẩu |
| UC04 | Đăng xuất |
| UC05 | Tìm công việc đề xuất |
| UC06 | Đăng tin tuyển dụng |
| UC07 | Cập nhật tin tuyển dụng |
| UC08 | Cập nhật kỹ năng |
| UC09 | Tạo kỹ năng cơ bản |
| UC010 | Chặn tài khoản |
| UC011 | Cập nhật kinh nghiệm |
| UC012 | Thêm kỹ năng |
| UC013 | Xem thông tin cá nhân |

*Bảng 3.2 Danh sách các tình huống trong hệ thống*

### Tình huống hoạt động

#### UC01\_Đăng ký

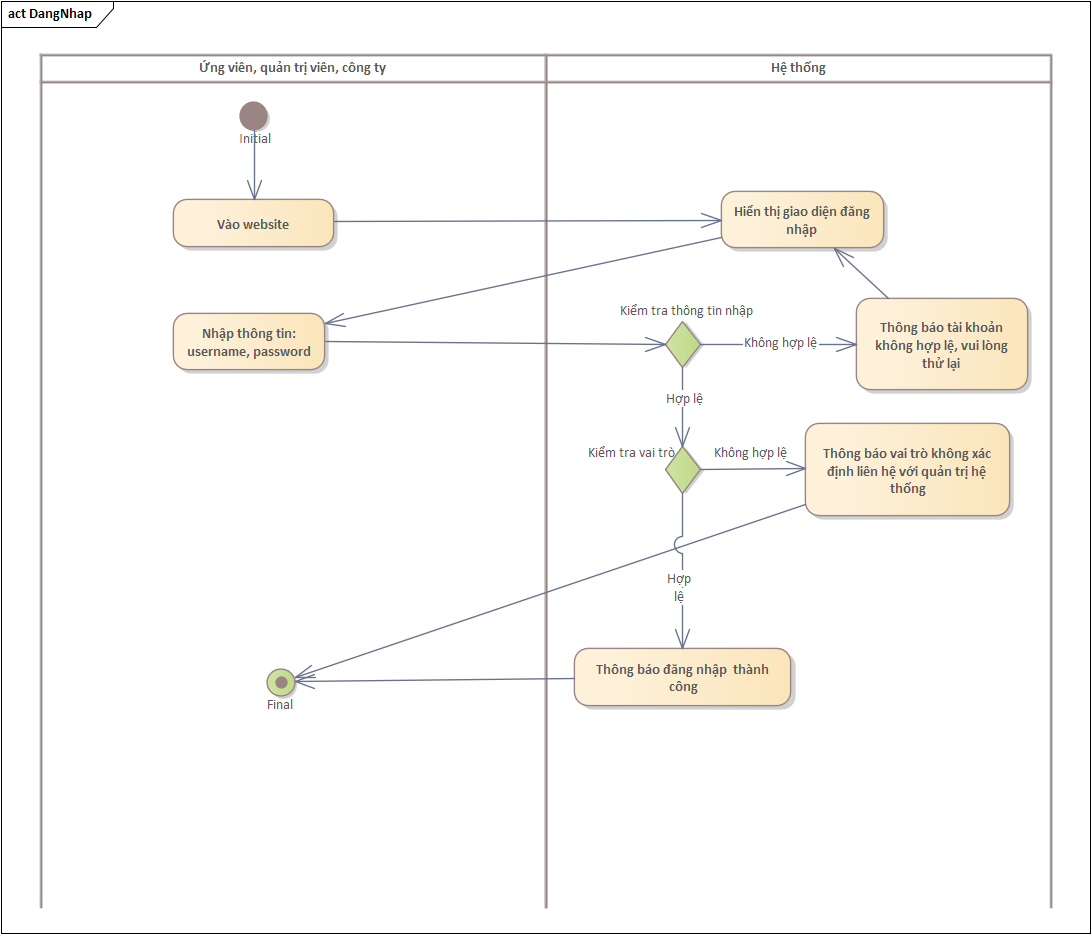
|  |  |
| --- | --- |
| * **Tên use case**: UC01\_Đăng ký | |
| * **Mô tả sơ lược**: Đăng ký tài khoản của khách hàng trước khi sử dụng các dịch vụ của RecyclePro | |
| * **Actor chính**: Khách hàng | |
| * **Actor phụ**: Không có | |
| * **Tiền điều kiện (Pre-condition):** Chưa có tài khoản sử dụng | |
| * **Hậu điều kiện (Post-condition):** Tài khoản đăng ký thành công sẽ được lưu vào cơ sở dữ liệu. | |
| * **Luồng sự kiện chính (main flow):** | |
| **Khách hàng** | **Hệ thống** |
| 1. Vào trang chủ |  |
|  | 1. Hiển thị giao diện đăng nhập. |
| 1. Chọn nút đăng ký tài khoản. |  |
|  | 1. Hiển thị giao diện đăng ký tài khoản. |
| 1. Nhập các thông tin: tên người dùng(username), địa chỉ mail (email), mật khẩu(password) và nhập lại mật khẩu (confirm password), loại tài khoản(type), địa chỉ (address), địa chỉ website (webURL) |  |
| 1. Nhấn vào nút đăng ký (signup) |  |
|  | 1. Kiểm tra các thông tin nhập vào. Nếu hợp lệ thông báo thành công. |
| 1. Xác nhận |  |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 7.1. Nếu không hợp lệ thông báo trường nhập vào nào bị lỗi. |
| 7.2. Người dùng nhập lại thông tin. |  |
| 7.3. Quay lại bước 7. |  |



*1. Sơ đồ hoạt động đăng ký*

#### UC02\_Đăng nhập

|  |  |
| --- | --- |
| * **Tên use case**: UC02\_Đăng nhập | |
| * **Mô tả sơ lược**: Người dùng muốn đăng nhập vào hệ thống để sử dụng dịch vụ | |
| * **Actor chính**: Ứng viên, Công ty, Quản trị viên | |
| * **Actor phụ**: Không có | |
| * **Tiền điều kiện (Pre-condition):** Kết nối internet, người dùng đã có tài khoản | |
| * **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Khách hàng** | **Hệ thống** |
| 1. Truy cập website |  |
|  | 1. Hiển thị giao diện đăng nhập |
| 1. Nhập vào các thông tin: username (username), mật khẩu (password) |  |
|  | 1. Kiểm tra thông tin nhập vào. |
|  | 1. Kiểm tra vai trò |
|  | 1. Thông báo đăng nhập thành công |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 4.1. Thông báo thông tin đăng nhập không đúng. |
|  | 4.2. Quay lại bước 2 |
|  | 5.1. Thông báo vai trò không hợp lệ |
|  | 5.2. Quay lại bước 2 |

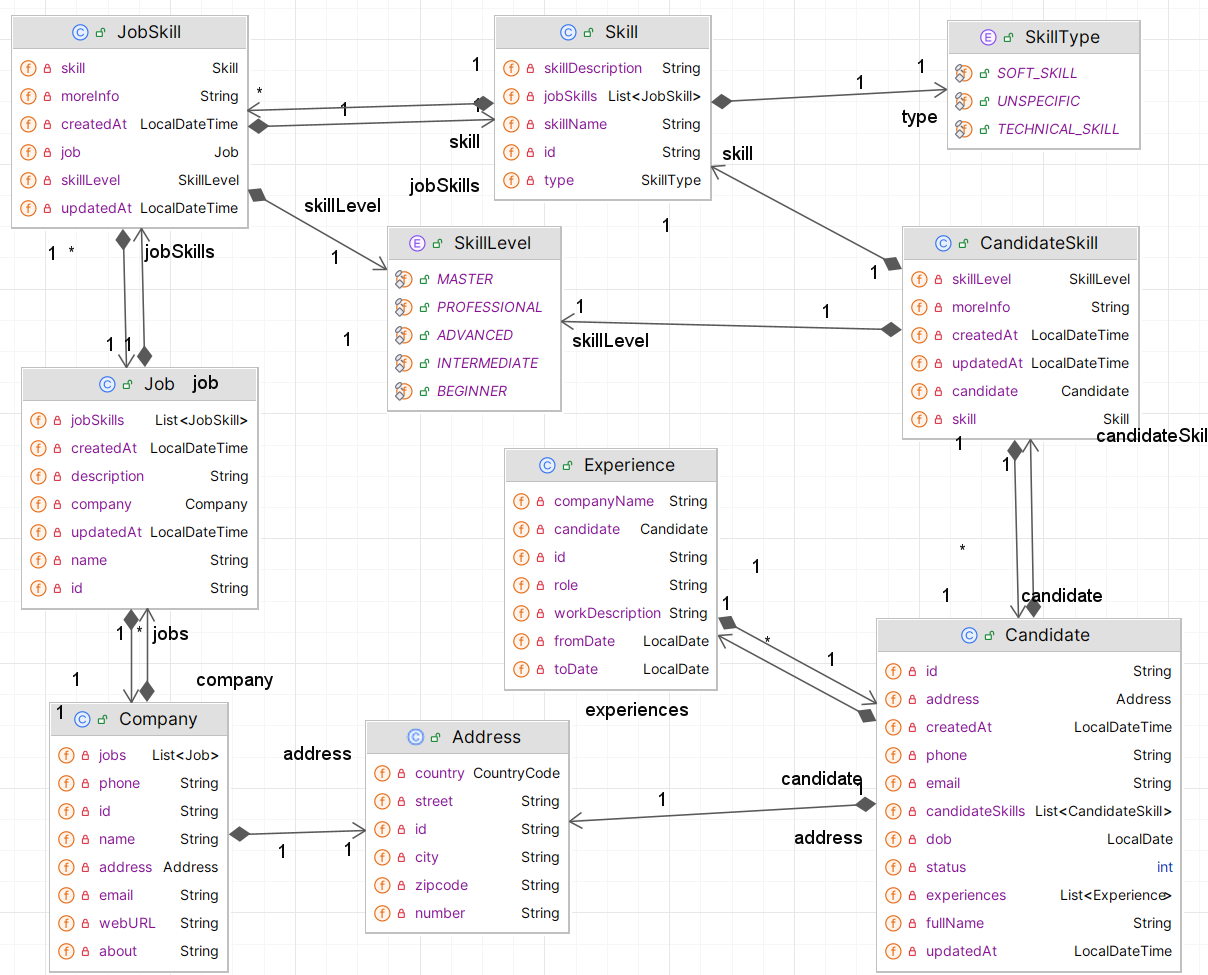


*2. Sơ đồ hoạt động đăng nhập.*

## Class diagram

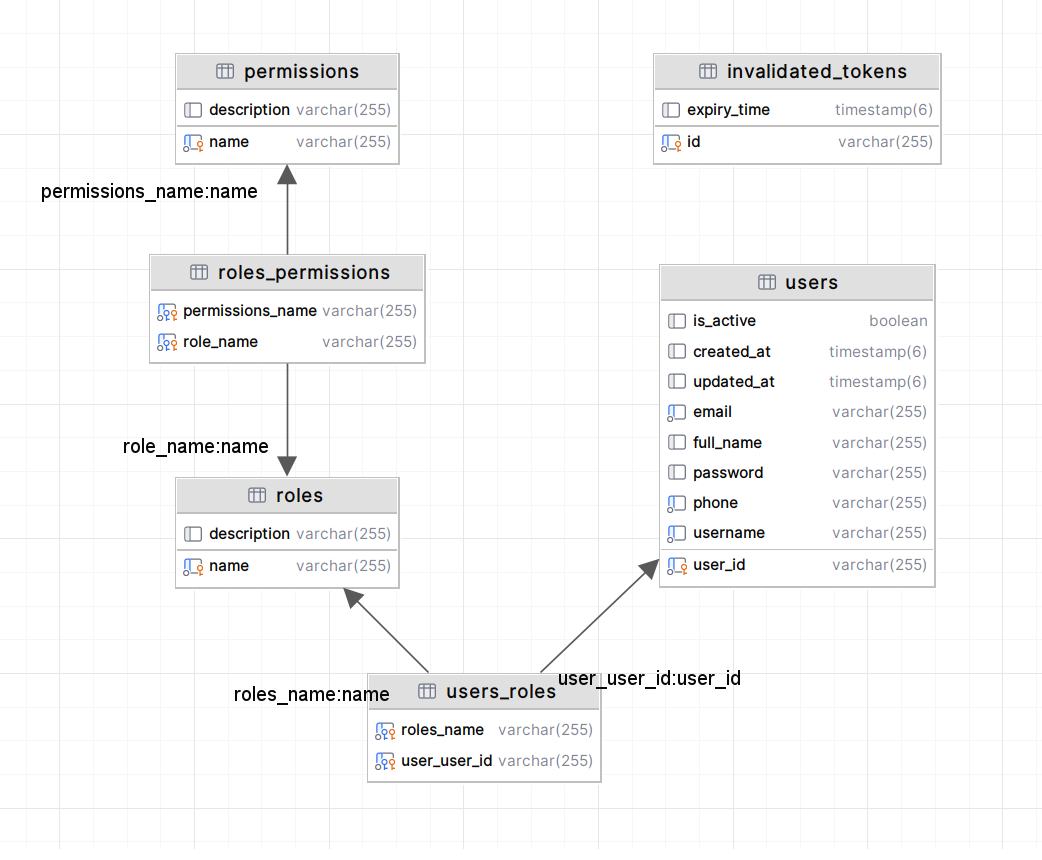
### Entity diagram

*1. indentity service*

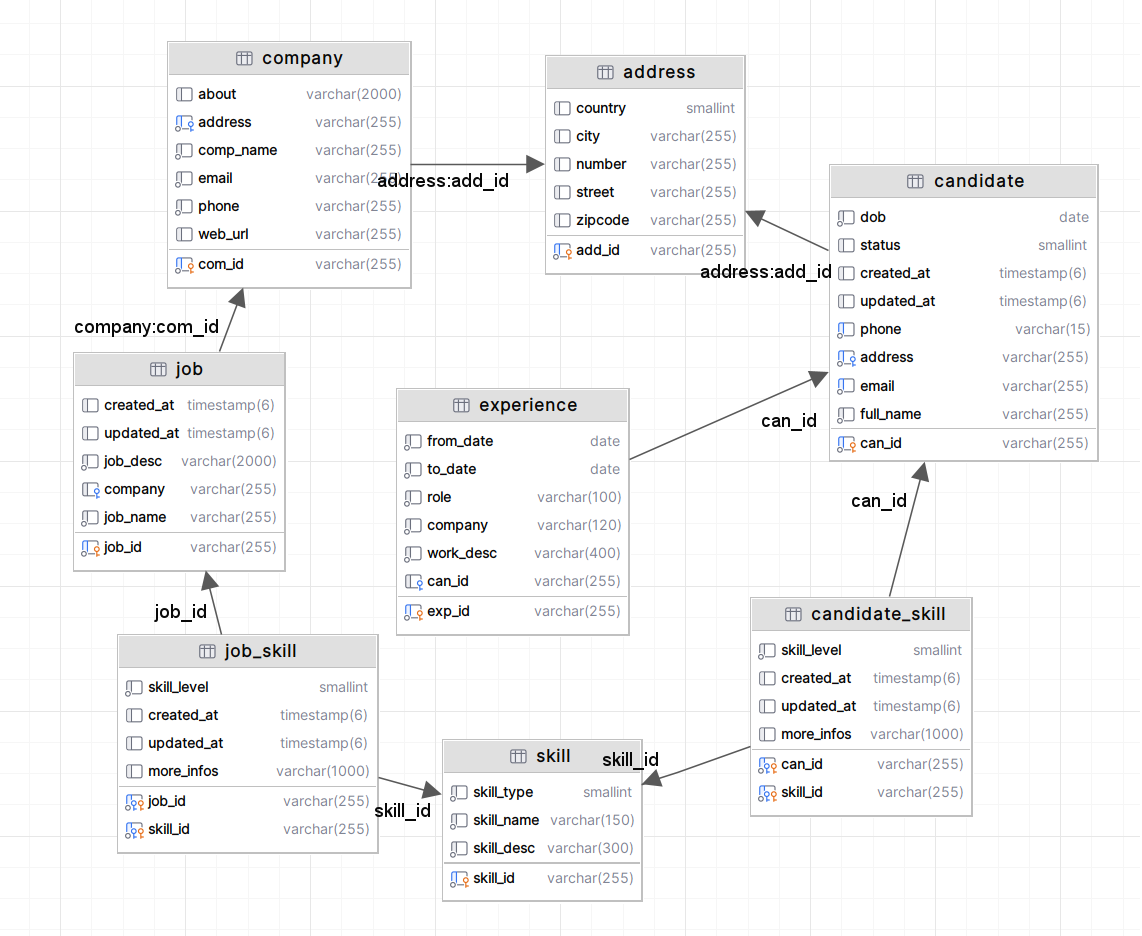


*2. jobtify service*

### Database diagram

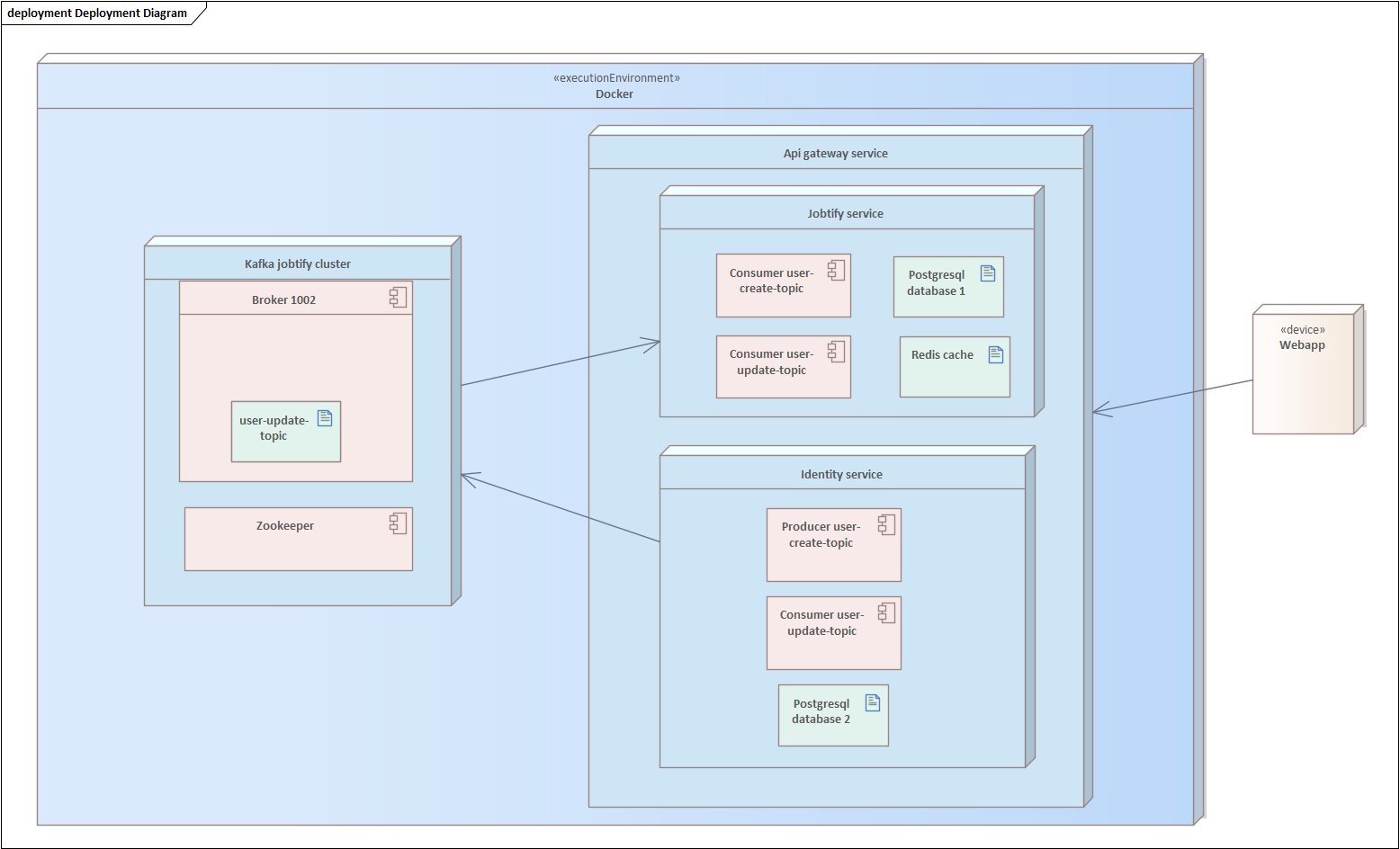


1. *identity service*



*2. jobtify service*

## Deployment diagram



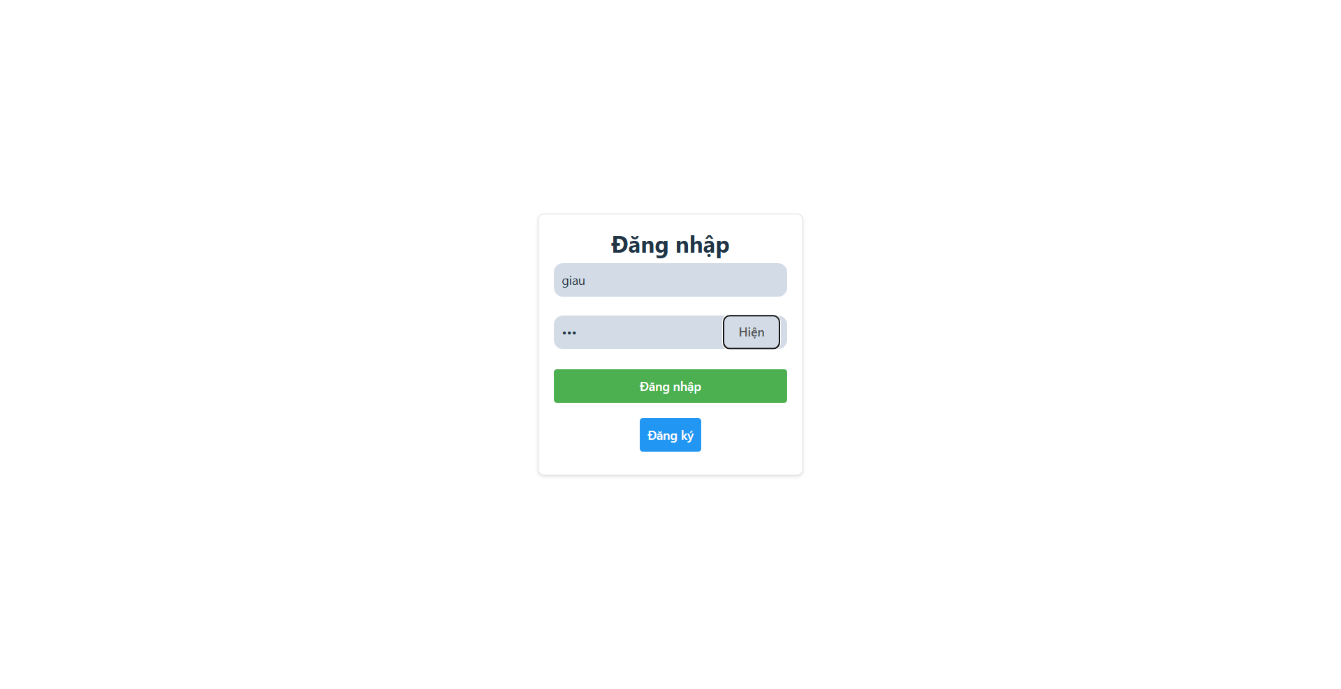
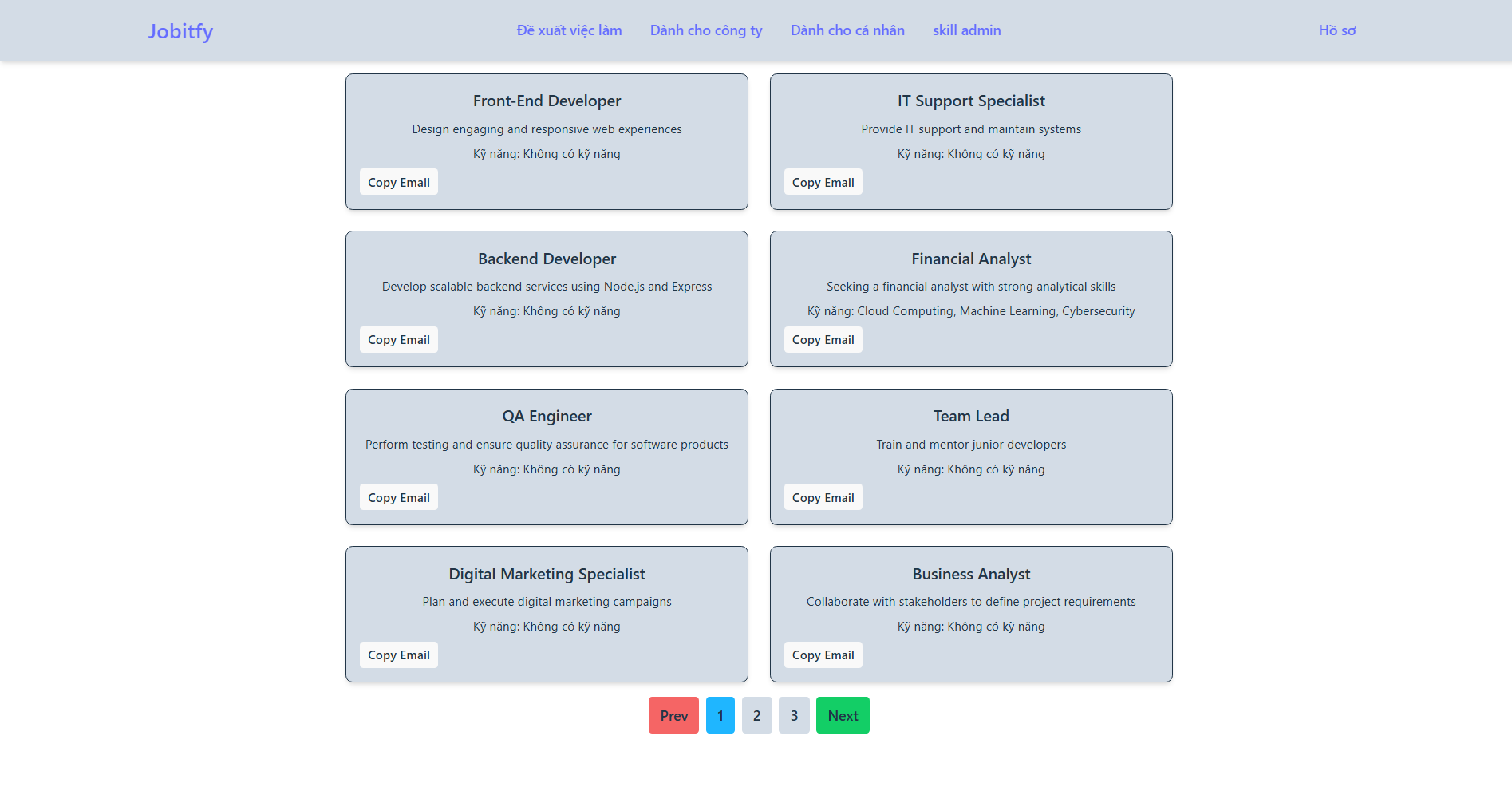
*1. Sơ đồ triển khai.*

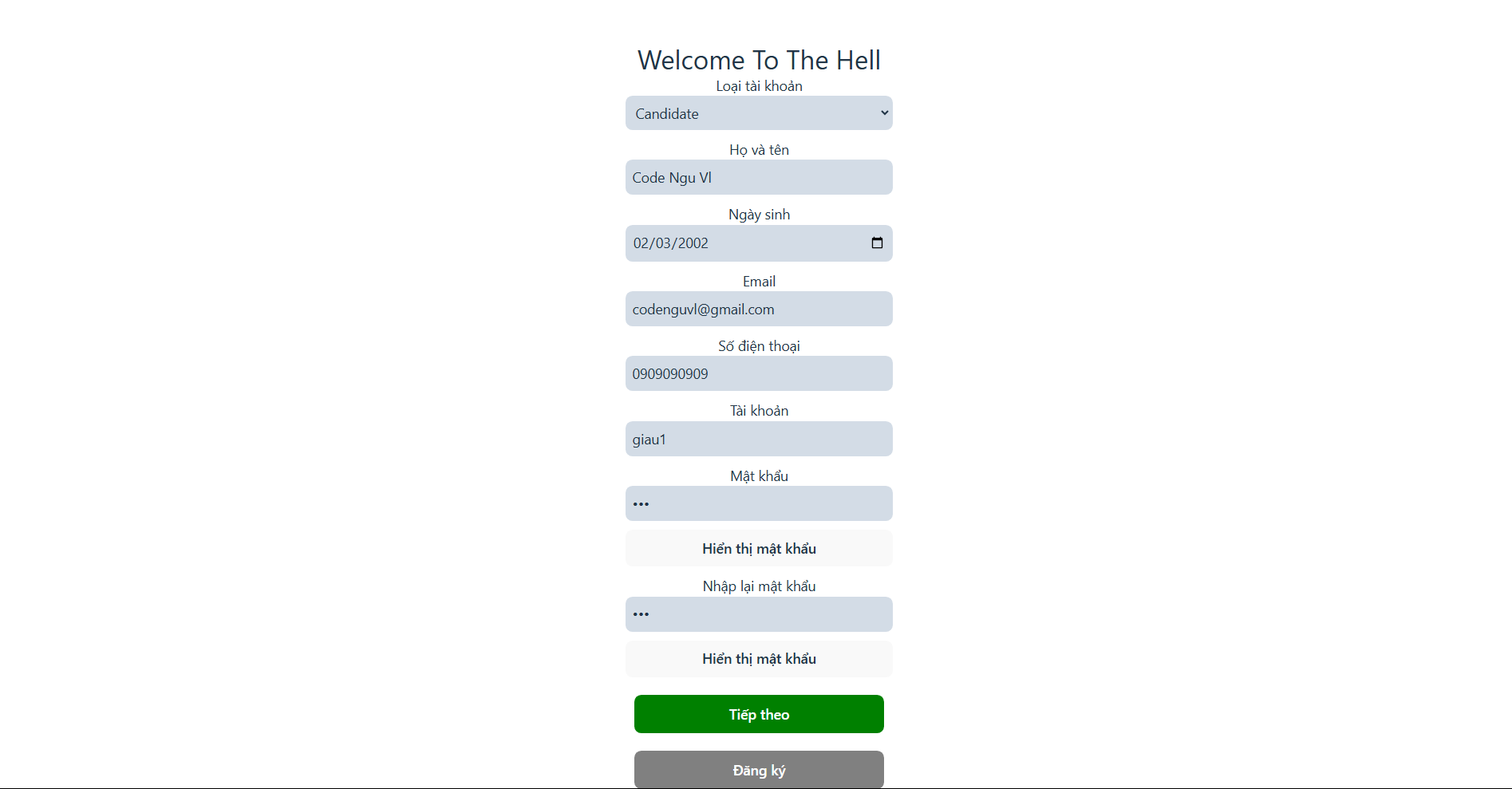
# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

Thiết bị sử dụng windows 10 hoặc cao hơn.

## Giao diện của hệ thống







## Kế hoạch và hiện thực kiểm thử hệ thống

### Kế hoạch kiểm thử

*4.1 Phần cứng thực hiện kiểm thử*

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **SSD** | **Architecture** |
| Intel Core i5 12600k, 3.70 GHz - 4.90 GHz | 32 GB | 256GB | x86\_64 |

*4.2 Phần mềm thực hiện kiểm thử*

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| IntelliJ Ultimate | 2024.2.1 | IDE cho java |
| Microsoft Windows 11 | 11 23H2 | Hệ điều hành |

### Kiểm thử hệ thống

*5.1 Kịch bản kiểm thử.*

| **Test ID** | **Chức năng** | **Mô tả** | **Điều kiện trước** | **Dữ liệu Test** | **Kết quả mong muốn** | **Ghi chú** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Đăng nhập | Kiểm tra chức năng dăng nhập. | Khởi động ứng dụng, kết nối internet. | Nhập đúng tài khoản và mật khẩu. | Màn hình chính hiện lên. |  |
| 2 | Đăng nhập | Kiểm tra chức năng đăng nhập. | Khởi động ứng dụng, kết nối internet. | Nhập sai tài khoản. | Hệ thống thông báo “Đăng nhập thất bại”. |  |
| 3 | Đăng nhập | Kiểm tra chức năng đăng nhập. | Khởi động ứng dụng, kết nối internet. | Nhập sai mật khẩu. | Hệ thống thông báo “Đăng nhập thất bại”. |  |
| 4 | Đăng nhập | Kiểm tra chức năng đăng nhập. | Khởi động ứng dụng, kết nối internet. | Không nhập tài khoản | Hệ thống thông báo “Đăng nhập thất bại”. |  |
| 5 | Đăng nhập | Kiểm tra chức năng đăng nhập. | Khởi động ứng dụng, kết nối internet. | Mật khẩu <6 kí tự. | Hệ thống thông báo “Đăng nhập thất bại”. | … |
| 6 | Đăng ký | Kiểm tra chức năng đăng ký. | Khởi động ứng dụng, kết nối internet, chưa có tài khoản. | Nhập username, email, mật khẩu(6 kí tự), xác nhận mật khẩu đúng định dạng. | Hệ thống thông báo “Đăng ký thành công”. |  |

*6. Bảng kết quả kiểm thử.*

| **Test ID** | **Ngày testing** | **Người tham gia Test** | **Pass/Fail** | **Độ nghiêm trọng** | **Tóm tắt lỗi** | **Ghi chú** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 12/12/2024 | Nguyễn Văn Giàu | 3/1 | Thấp | Xác thực chuyển đường dẫn thất bại lỗi xác thực |  |
|  |  |  |  |  |  |  |

# : KẾT LUẬN

## Kết quả đạt được

Giao diện đăng ký, đăng nhập

Hệ thống gợi ý công việc cho ứng viên

Quản lý kĩ năng cho quản trị viên

Cập nhật thông tin người dùng

Xác thực jwt, phân quyền người dùng.

## Hướng phát triển

Bổ sung giao diện cho ứng viên và công ty.

Tích hợp elastic search và đông bộ dữ liệu với mongodb atlas phục vụ việc phân tích đề xuất công việc phù hợp cho ứng viên.

Bổ sung tính năng gửi email cho ứng viên khi tìm thấy công việc phù hợp.

# TÀI LIỆU THAM KHẢO

Các tài liệu từ Internet

1. [**Sparx Systems** - Sparx Enterprise Architect User Guide.Sparx Systems Documentation – Truy cập lần cuối 12/2024.](https://sparxsystems.com/resources/tutorials/)
2. [**Visual Paradigm** - Visual Paradigm Documentation. Visual Paradigm Documentation – Truy cập lần cuối 12/2024](https://circle.visual-paradigm.com/diagram-examples/)**.**
3. [**Apache Kafka** - Apache Kafka Documentation. Apache Software Foundation – Truy cập lần cuối 12/2024.](https://kafka.apache.org/20/documentation.html)
4. [**Redis** - Redis Documentation. Redis.io – Truy cập lần cuối 12/2024.](https://redis.io/docs/latest/develop/)
5. [**Spring Framework** - Spring Framework Documentation. Spring.io – Truy cập lần cuối 12/2024.](https://docs.spring.io/spring-framework/reference/index.html)
6. [**Spring for Apache Kafka** - Spring for Apache Kafka Documentation. Spring.io – Truy cập lần cuối 12/2024.](https://spring.io/projects/spring-kafka)
7. [**Spring Data Redis** - Spring Data Redis Documentation. Spring.io – Truy cập lần cuối 12/2024.](https://docs.spring.io/spring-data/redis/docs/2.4.5/reference/html/#reference)