

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

# **ĐỒ ÁN TỐT NGHIỆP**

**Xây dựng hệ thống mạng xã hội tìm kiếm việc làm**

**NGUYỄN ĐỨC THÀNH**  
thanh.nd183991@sis.hust.edu.vn

**Ngành Khoa Học Máy Tính**

**Giảng viên hướng dẫn:** ThS. Nguyễn Tiến Thành

\_\_\_\_\_  
Chữ kí GVHD

**Khoa:** Công nghệ thông tin

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 08/2022**

# LỜI CAM KẾT

Họ và tên sinh viên: Nguyễn Đức Thành .....  
Điện thoại liên lạc: 0386851140 .....  
Email: thanh.nd183991@sis.hust.edu.vn .....  
Lớp: Công nghệ thông tin 01 –K63 .....  
Hệ đào tạo: Chính quy .....

Tôi – *Nguyễn Đức Thành* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Thạc sỹ Nguyễn Tiến Thành*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

*Hà Nội, ngày 08 tháng 08 năm 2022*

Tác giả ĐATN

*Nguyễn Đức Thành*

# LỜI CẢM ƠN

Trước hết, tôi xin chân thành cảm ơn thầy Thạc Sỹ Nguyễn Tiến Thành- Bộ môn Khoa học máy tính – Trường Công nghệ Thông tin và Truyền thông – Trường Đại học Bách khoa Hà Nội, người đã hướng dẫn tôi rất nhiều trong suốt quá trình tìm hiểu làm đồ án tốt nghiệp. Đồng thời, tôi xin cảm ơn các thầy cô trong trường đã cung cấp cho tôi những kiến thức cơ bản, cần thiết để tôi hoàn thành đồ án tốt nghiệp này.

Tôi xin gửi lời cảm ơn đến các thành viên lớp Công nghệ thông tin IT3-01, những người bạn đã luôn ở bên cạnh động viên, tạo điều kiện thuận lợi giúp tôi hoàn thành tốt đồ án. Sau cùng, tôi xin gửi lời cảm ơn đến gia đình, bạn bè đã tạo mọi điều kiện để tôi xây dựng thành công đồ án này.

Mặc dù bản thân tôi đã cố gắng nhưng do thời gian, kiến thức và kinh nghiệm có hạn nên sản phẩm và báo cáo của tôi còn có thể có nhiều thiếu sót. . . Vì vậy, tôi rất mong nhận được sự thông cảm và nhận xét từ thầy để sản phẩm cũng như báo cáo của tôi được hoàn thiện hơn.

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Ngày nay, các ứng dụng công nghệ thông tin đã được áp dụng trong mọi lĩnh vực nghiên cứu khoa học kỹ thuật cũng như trong đời sống, máy tính trở thành công cụ trợ giúp đắc lực cho con người trong lưu trữ, phân tích và xử lý thông tin. Đối với lĩnh vực tuyển dụng việc làm, áp dụng công nghệ thông tin mang lại những lợi ích thiết thực hơn bao giờ hết. Trước đây, khi hạ tầng công nghệ chưa phát triển, mọi người thường tìm kiếm công việc qua các mẫu tin trên báo. Người tìm việc cũng có thể dựa vào các công ty môi giới việc làm nhưng cũng chỉ đáp ứng một phần nhỏ thị trường lao động. Những cách này thường tốn thời gian, công sức với cả người tuyển dụng lẫn người tìm việc.

Áp dụng công nghệ thông tin trong việc tìm kiếm việc làm đã gần như xóa bỏ những khuyết điểm của những hình thức tìm kiếm cũ. Tuy nhiên, các nội dung tuyển dụng được đăng khắp mọi nơi trên Internet: mạng xã hội, trang website của công ty, mẫu tin tuyển việc trên các website blog... Điều này gây khó khăn cho người lao động (tìm kiếm việc làm) khi tìm kiếm nguồn thông tin xác thực và cản trở nhà tuyển dụng tiếp cận các ứng viên tiềm năng.

Để tận dụng tối đa ưu điểm và hạn chế tối thiểu khuyết điểm của việc áp dụng công nghệ thông tin đối với lĩnh vực tìm kiếm việc làm, tôi chọn đề tài đồ án tốt nghiệp của mình là “Xây dựng hệ thống mạng xã hội tìm kiếm việc làm”. Hệ thống được phát triển nhằm tạo môi trường giao tiếp nhanh chóng, hiệu quả, tin cậy giữa người tìm việc và nhà tuyển dụng.

Hệ thống được phát triển theo mô hình thiết kế client-server. Bên phía client, hệ thống sử dụng thư viện ReactJS để xây dựng giao diện cho người dùng, phía server sử dụng NodeJS và cơ sở dữ liệu phi quan hệ NoSQL MongoDB. Hệ thống đã đi vào thử nghiệm và bước đầu hoạt động ổn định trên mạng cục bộ (LAN – Local Area Network). Hệ thống sẽ tiếp tục bổ sung, hoàn thiện cơ sở dữ liệu, cải tiến giao diện của người dùng trong các phiên bản tiếp theo.

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	3
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>5</b>
2.1 Khảo sát hiện trạng .....	5
2.2 Tổng quan chức năng .....	6
2.2.1 Biểu đồ use case tổng quát .....	6
2.2.2 Biểu đồ use case phân rã “Quản lý người dùng” .....	10
2.2.3 Biểu đồ use case phân rã “Quản lý bài đăng” .....	11
2.2.4 Biểu đồ use case phân rã “Quản lý công ty” .....	12
2.2.5 Biểu đồ use case phân rã “Quản lý tin tuyển dụng” .....	13
2.2.6 Biểu đồ use case phân rã “Quản lý kỹ năng” .....	14
2.2.7 Biểu đồ use case phân rã “Theo dõi” .....	15
2.2.8 Biểu đồ use case phân rã “Tìm kiếm” .....	16
2.2.9 Biểu đồ use case phân rã “Quản lý công ty của tôi” .....	17
2.2.10 Quy trình nghiệp vụ.....	18
2.3 Đặc tả chức năng .....	20
2.3.1 Đặc tả use case “Quên mật khẩu” .....	20
2.3.2 Đặc tả use case “Cập nhật thông tin” .....	22
2.3.3 Đặc tả use case “Ứng tuyển công việc” .....	24
2.3.4 Đặc tả use case “Đăng tin tuyển dụng” .....	25

2.4 Yêu cầu phi chức năng .....	27
2.4.1 Yêu cầu bảo mật.....	27
2.4.2 Yêu cầu vận hành .....	27
2.4.3 Yêu cầu giao diện.....	28
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>29</b>
3.1 Ngôn ngữ lập trình Javascript .....	29
3.2 Công nghệ lập trình giao diện ReactJS.....	29
3.3 Công nghệ Redux .....	31
3.4 Công nghệ React Query .....	32
3.5 Công nghệ lập trình máy chủ NodeJS .....	33
3.6 Công nghệ JWT .....	33
3.7 Công nghệ lập trình thời gian thực SocketIO .....	34
3.8 Cơ sở dữ liệu MongoDB.....	35
<b>CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ .....</b>	<b>37</b>
4.1 Thiết kế kiến trúc.....	37
4.1.1 Lựa chọn kiến trúc phần mềm .....	37
4.1.2 Thiết kế tổng quan.....	39
4.1.3 Thiết kế chi tiết gói .....	41
4.2 Thiết kế chi tiết.....	46
4.2.1 Thiết kế giao diện .....	46
4.2.2 Thiết kế lớp .....	48
4.2.3 Thiết kế cơ sở dữ liệu .....	52
4.3 Xây dựng ứng dụng.....	57
4.3.1 Thư viện và công cụ sử dụng.....	57
4.3.2 Kết quả đạt được .....	57
4.3.3 Minh hoạ các chức năng chính .....	59

4.4 Kiểm thử.....	65
4.4.1 Kiểm thử chức năng phê duyệt tin tuyển dụng.....	65
4.4.2 Kiểm thử chức năng ứng tuyển công việc .....	66
4.5 Triển khai .....	69
<b>CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....</b>	<b>71</b>
5.1 Phân trang và cuộn vô hạn danh sách dữ liệu .....	71
5.1.1 Đặt vấn đề .....	71
5.1.2 Giải pháp .....	71
5.2 Tạo CV.....	72
5.2.1 Đặt vấn đề .....	72
5.2.2 Giải pháp .....	72
5.3 Thiết kế Category Collection .....	73
5.3.1 Đặt vấn đề .....	73
5.3.2 Giải pháp .....	73
5.4 Thiết kế Notification Collection .....	75
5.4.1 Đặt vấn đề .....	75
5.4.2 Giải pháp .....	75
5.5 Lưu trữ file.....	76
5.5.1 Đặt vấn đề .....	76
5.5.2 Giải pháp .....	76
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>77</b>
6.1 Kết luận.....	77
6.2 Hướng phát triển.....	78
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>82</b>
<b>PHỤ LỤC.....</b>	<b>84</b>

<b>A. ĐẶC TẢ USE CASE.....</b>	<b>84</b>
A.1 Đặc tả use case “Theo dõi người dùng” .....	84
A.2 Đặc tả use case “Đăng bài” .....	84
A.3 Đặc tả use case “Trao đổi” .....	86



## DANH MỤC HÌNH VẼ

Hình 1.1	Lực lượng lao động theo quý, 2020 – 2022 (Đơn vị tính: Triệu người) . . . . .	1
Hình 2.1	Sơ đồ Usecase tổng quan . . . . .	6
Hình 2.2	Phân rã Usecase "Quản lý người dùng" . . . . .	10
Hình 2.3	Phân rã Usecase "Quản lý bài đăng" . . . . .	11
Hình 2.4	Phân rã Usecase "Quản lý công ty" . . . . .	12
Hình 2.5	Phân rã Usecase "Quản lý tin tuyển dụng" . . . . .	13
Hình 2.6	Phân rã Usecase "Quản lý kỹ năng" . . . . .	14
Hình 2.7	Phân rã Usecase "Theo dõi" . . . . .	15
Hình 2.8	Phân rã Usecase "Tìm kiếm" . . . . .	16
Hình 2.9	Phân rã Usecase "Quản lý công ty của tôi" . . . . .	17
Hình 2.10	Biểu đồ hoạt động đăng tin tuyển dụng . . . . .	19
Hình 2.11	Biểu đồ hoạt động thay đổi chủ tin tuyển dụng . . . . .	20
Hình 3.1	Cấu trúc, thành phần của ReactJS . . . . .	30
Hình 3.2	Redux design pattern . . . . .	31
Hình 3.3	Luồng Redux với middleware . . . . .	32
Hình 3.4	Cách thức hoạt động JWT . . . . .	34
Hình 4.1	MVC Design Pattern . . . . .	37
Hình 4.2	MVP Design Pattern . . . . .	38
Hình 4.3	Biểu đồ phụ thuộc gói Client . . . . .	39
Hình 4.4	Biểu đồ phụ thuộc gói Server . . . . .	40
Hình 4.5	Thiết kế gói thành phần giao diện . . . . .	41
Hình 4.6	Thiết kế chi tiết gói auth client . . . . .	42
Hình 4.7	Thiết kế chi tiết gói auth server . . . . .	43
Hình 4.8	Thiết kế màn hình trang chủ . . . . .	44
Hình 4.9	Thiết kế gói nghiệp vụ việc làm . . . . .	45
Hình 4.10	Bảng màu với giao diện sáng . . . . .	46
Hình 4.11	Bảng màu với giao diện tối . . . . .	47
Hình 4.12	Thiết kế giao diện trang việc làm . . . . .	47
Hình 4.13	Thiết kế trang chi tiết công việc . . . . .	48
Hình 4.14	Biểu đồ tuần tự createJob . . . . .	50
Hình 4.15	Biểu đồ tuần tự signPresignedURL . . . . .	51
Hình 4.16	Biểu đồ tuần tự createPost . . . . .	52

Hình 4.17	Biểu đồ thực thể liên kết . . . . .	52
Hình 4.18	Biểu đồ cơ sở dữ liệu phi quan hệ . . . . .	56
Hình 4.19	Giao diện trang chủ người dùng . . . . .	59
Hình 4.20	Giao diện trang việc làm . . . . .	60
Hình 4.21	Giao diện trang chi tiết việc làm . . . . .	60
Hình 4.22	Giao diện trang thông báo . . . . .	61
Hình 4.23	Giao diện trang tìm kiếm . . . . .	61
Hình 4.24	Giao diện trang trao đổi . . . . .	62
Hình 4.25	Giao diện trang công ty . . . . .	62
Hình 4.26	Giao diện trang hồ sơ người dùng . . . . .	63
Hình 4.27	Giao diện trang chủ quản trị viên . . . . .	63
Hình 4.28	Giao diện trang người dùng quản trị viên . . . . .	64
Hình 4.29	Mô hình triển khai của hệ thống . . . . .	69
Hình 5.1	Mối quan hệ kỹ năng khi tạo tin tuyển dụng . . . . .	73
Hình 5.2	Vấn đề mối quan hệ giữa lĩnh vực tương đồng . . . . .	74
Hình 5.3	Thiết kế mối quan hệ giữa lĩnh vực tương đồng . . . . .	74
Hình 6.1	Thiết kế Evaluate Collection . . . . .	78
Hình 6.2	Chuẩn hóa User Collection . . . . .	79

## DANH MỤC BẢNG BIỂU

Bảng 2.1	Danh sách tác nhân . . . . .	7
Bảng 2.2	Danh sách Usecase . . . . .	10
Bảng 2.3	Usecase "Quản lý người dùng" . . . . .	11
Bảng 2.4	Usecase "Quản lý bài đăng" . . . . .	11
Bảng 2.5	Usecase "Quản lý công ty" . . . . .	12
Bảng 2.6	Usecase "Quản lý tin tuyển dụng" . . . . .	13
Bảng 2.7	Usecase "Quản lý kỹ năng" . . . . .	14
Bảng 2.8	Usecase "Theo dõi" . . . . .	15
Bảng 2.9	Usecase "Tìm kiếm" . . . . .	16
Bảng 2.10	Usecase "Quản lý công ty của tôi" . . . . .	17
Bảng 2.11	Quy trình nghiệp vụ đăng tin tuyển dụng . . . . .	18
Bảng 2.12	Quy trình nghiệp vụ thay đổi chủ của tin tuyển dụng . . . . .	20
Bảng 2.13	Đặc tả usecase “Quên mật khẩu” . . . . .	21
Bảng 2.16	Đặc tả usecase “Cập nhật thông tin” . . . . .	22
Bảng 2.18	Đặc tả usecase “Ứng tuyển công việc” . . . . .	25
Bảng 2.20	Đặc tả usecase “Đăng tin tuyển dụng” . . . . .	26
Bảng 3.1	So sánh các thư viện tìm nạp dữ liệu (chi tiết [17]) . . . . .	33
Bảng 4.1	Đặc tả tổng quan gói Client . . . . .	39
Bảng 4.2	Đặc tả tổng quan gói Server . . . . .	40
Bảng 4.3	Thành phần chính của gói giao diện . . . . .	42
Bảng 4.4	Đặc tả gói auth client . . . . .	43
Bảng 4.5	Đặc tả gói auth server . . . . .	43
Bảng 4.6	Đặc tả gói nghiệp vụ việc làm . . . . .	46
Bảng 4.7	Mô tả thuộc tính của lớp JobService . . . . .	49
Bảng 4.8	Mô tả phương thức của lớp JobService . . . . .	50
Bảng 4.9	Mô tả thuộc tính của lớp PostServer . . . . .	50
Bảng 4.10	Mô tả phương thức của lớp PostService . . . . .	51
Bảng 4.11	Thiết kế thực thể User . . . . .	53
Bảng 4.12	Thiết kế thực thể Post . . . . .	53
Bảng 4.13	Thiết kế thực thể Company . . . . .	54
Bảng 4.14	Thiết kế thực thể Job . . . . .	55
Bảng 4.15	Thiết kế thực thể Notification . . . . .	55
Bảng 4.16	Bảng thiết kế liên kết giữa các thực thể . . . . .	56
Bảng 4.17	Danh sách thư viện và công cụ sử dụng . . . . .	57

Bảng 4.18	Thống kê gói src server . . . . .	58
Bảng 4.19	Thống kê gói src client . . . . .	58
Bảng 4.20	Kiểm thử chức năng duyệt tin tuyển dụng . . . . .	66
Bảng 4.21	Bảng kiểm thử chức năng ứng tuyển công việc . . . . .	69
Bảng 4.22	Bảng thông số cầu hình server . . . . .	70
Bảng 5.1	Thiết kế Category Collection . . . . .	75
Bảng A.1	Đặc tả usecase “Theo dõi người dùng” . . . . .	84
Bảng A.2	Đặc tả usecase “Đăng bài” . . . . .	85
Bảng A.4	Đặc tả usecase “Trao đổi” . . . . .	86

## DANH MỤC CÁC TỪ VIẾT TẮT

Viết tắt	Tên tiếng Anh	Tên tiếng Việt
<b>API</b>	Application Programming Inter- face	Giao diện lập trình ứng dụng
<b>HTML</b>	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
<b>CNTT</b>		Công nghệ thông tin
<b>ĐATN</b>		Đồ án tốt nghiệp
<b>SV</b>		Sinh viên
<b>CSS</b>	Cascading Style Sheets	Ngôn ngữ sử dụng để định dạng dữ liệu
<b>JSON</b>	JavaScript Object Notation	
<b>JWT</b>	JSON Web Token	Mã thông báo web JSON
<b>UI</b>	User Interface	Giao diện người dùng
<b>UX</b>	User Experience	Trải nghiệm người dùng
<b>GUI</b>	Graphical User Interface	Giao diện đồ họa người dùng
<b>CV</b>	Curriculum Vitae	Sơ yếu lý lịch
<b>RDBMS</b>	Relational Database Manage- ment System	Hệ quản trị cơ sở dữ liệu quan hệ
<b>RAM</b>	Random Access Memory	Bộ nhớ truy xuất ngẫu nhiên
<b>LAN</b>	Local Area Network	Mạng cục bộ
<b>DOM</b>	Document Object Model	

## DANH MỤC THUẬT NGỮ

Viết tắt	Tên tiếng Anh
<b>Internet</b>	Mạng máy tính được kết nối với nhau để chia sẻ thông tin
<b>Browser</b>	Trình duyệt
<b>Cache memory</b>	Bộ nhớ đệm
<b>Interpreter</b>	Trình thông dịch
<b>Compiler</b>	Trình biên dịch
<b>NodeJS</b>	Là nền tảng được xây dựng dựa trên “V8 Javascript engine”(giúp phân tích và thông dịch javascript thành mã máy) được viết bằng C++
<b>Server</b>	Một máy chủ hay máy phục vụ là một phần của máy tính phần cứng hoặc phần mềm mà cung cấp chức năng cho các chương trình khác.
<b>Client</b>	Thiết bị điện tử dùng để truy cập dịch vụ do máy chủ cung cấp
<b>Cloud Storage</b>	Lưu trữ đám mây
<b>RESTful API</b>	Một tiêu chuẩn trong việc thiết kế API
<b>Web stack</b>	Bộ công cụ phát triển ứng dụng web
<b>Design Pattern</b>	Mẫu thiết kế
<b>Framework</b>	Bộ mã dựng sẵn dựa trên các mẫu thiết kế

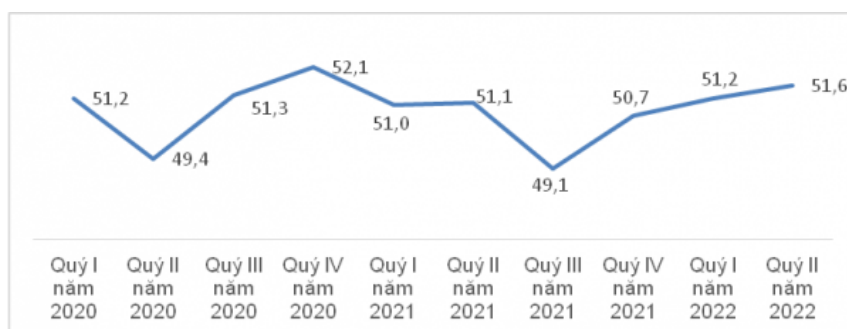
# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong thời đại công nghiệp hóa, hiện đại hóa ngày nay, cuộc sống của con người ngày càng thay đổi trên nhiều lĩnh vực. Đối với lĩnh vực tìm kiếm việc làm, trước đây người lao động tiếp cận việc làm chủ yếu thông qua các tờ rơi, các mẫu tin nhỏ trên các tờ báo, hay hiện đại hơn là thông qua các công ty môi giới tuyển dụng. Nhưng đối với một xã hội công nghệ phát triển mạnh mẽ như hiện nay, các hình thức đó đã không phù hợp và ít xuất hiện.

Thay vào đó, việc áp dụng công nghệ thông tin vào trong lĩnh vực tuyển dụng được phổ biến ngày càng rộng rãi. Đăng tin tuyển dụng việc làm có thể dưới nhiều hình thức khác nhau trên Internet: mạng xã hội, trang website của công ty tuyển dụng. Đối với các bạn sinh viên, cầu nối giữa nhà trường và doanh nghiệp giúp cho sinh viên tiếp cận việc làm thông qua các buổi trò chuyện online trong thời đại dịch bệnh hiện nay. Áp dụng công nghệ thông tin vào trong lĩnh vực tuyển dụng việc làm đã giải quyết được khó khăn mà các hình thức trước kia gặp phải. Người lao động (tìm việc) sẽ không phải tốn nhiều thời gian, công sức để tìm kiếm việc làm.

Sau hai năm dịch bệnh Covid, thị trường việc hoạt động sôi nổi khi các hoạt động kinh tế – xã hội đã cơ bản trở lại hoạt động trong trạng thái bình thường mới, nền kinh tế có dấu hiệu hồi phục. Theo trang website chính thức của Tổng cục thống kê Việt Nam quý II và đầu tháng 6 năm 2022, số người lao động có việc làm đã tăng so với quý trước. Tỷ lệ thất nghiệp và thiếu việc làm trong độ tuổi lao động đều giảm so với quý trước và cùng kỳ năm trước.



**Hình 1.1:** Lực lượng lao động theo quý, 2020 – 2022 (Đơn vị tính: Triệu người)

Hình 1.1 cho ta thấy, số lượng người lao động có việc trong các quý gần đây có dấu hiệu tăng trưởng. Tuy nhiên, người lao động khi tìm kiếm việc làm sau dịch bệnh cần vô cùng cảnh giác, cần sàng lọc các thông tin việc làm tránh bị các đối tượng xấu lợi dụng, dụ dỗ. Gần đây, các công việc như: tuyển cộng tác viên bán

hàng online, cộng tác viên ảo, chốt đơn ảo... liên tục xuất hiện trên không gian mạng đánh vào tâm lý mong muốn có việc làm sau dịch bệnh hòng lừa đảo chiếm đoạt tài sản của người lao động (theo trang vtv.vn).

Hiểu được điều đó, người viết đề án chọn đề tài “Xây dựng mạng xã hội tìm kiếm việc làm”. Hệ thống sẽ cung cấp các thông tin đúng đắn, chính xác cho người lao động dưới sự giám sát, kiểm duyệt của đội ngũ quản trị viên. Hệ thống tạo ra môi trường giao tiếp hiệu quả giữa người lao động và nhà tuyển dụng, giúp họ tìm đến nhau một cách dễ dàng.

## **1.2 Mục tiêu và phạm vi đề tài**

Mục đích của đề tài là tạo ra một trang mạng xã hội hướng việc làm giúp đỡ cho cả người tìm kiếm việc làm và nhà tuyển dụng. Khác với mạng xã hội thông thường, thay vì các hoạt động đăng bài về: con người, nấu ăn, làm đẹp... Các bài đăng trong hệ thống sẽ tập trung vào việc giới thiệu bản thân, quảng bá công ty giúp người sử dụng hiểu rõ về các đối tượng (người dùng và công ty) mà mình quan tâm.

Hệ thống cần có tương tác hai chiều giữa người tìm việc và nhà tuyển dụng. Người tìm việc hoàn toàn có thể chủ động trong việc tìm kiếm việc do nhà tuyển dụng sở hữu. Nhà tuyển dụng cũng có khả năng tìm kiếm những ứng viên phù hợp với việc làm thuộc sở hữu của mình.

Quản trị viên hệ thống có nhiệm vụ kiểm tra xác thực các thông tin tồn tại trong hệ thống bao gồm: người dùng, tin tuyển dụng, hồ sơ công ty... đảm bảo cung cấp chính xác, đúng đắn thông tin “tối tay” người lao động. Đối với nhà quản lý công ty, hệ thống cung cấp các chức năng cho phép họ giám sát công ty của mình bao gồm: tin tuyển dụng, nhà tuyển dụng của công ty, bài đăng của công ty... nhằm điều phối hoạt động công ty của mình trong hệ thống.

Đồng thời, hệ thống có cung cấp dịch vụ trao đổi, thông báo và báo cáo để giúp các thành viên trong mạng lưới giao tiếp với nhau một cách trực tiếp, dễ dàng và hiệu quả.

## **1.3 Định hướng giải pháp**

Hệ thống được phát triển dựa trên giao diện đồ họa người dùng (GUI) cho phép người dùng hoàn thành tác vụ với tốc độ nhanh hơn. Hệ thống sẽ tránh hoàn toàn việc sử dụng giao diện dòng lệnh, bởi vì, nó gây cản trở người dùng khi bắt họ phải ghi nhớ các câu lệnh để thao tác với hệ thống. Ưu điểm của việc sử dụng GUI bởi có các phần tử đồ họa tương tác với các chức năng giống nhau giúp người dùng dễ dàng ghi nhớ và nắm bắt thông tin trực quan, sinh động.



Hệ thống phát triển dựa trên nền tảng website, một ứng dụng có sẵn và được tích hợp trên mọi loại thiết bị điện tử hiện nay. Hệ thống được xây dựng dựa trên mô hình client-server, một mô hình được sử dụng rộng rãi bởi vì các ưu điểm của nó: tốc độ truy cập nhanh, khả năng mở rộng cao, đáng tin cậy, an toàn, và hoạt động với bất kỳ loại ứng dụng nào... Hệ thống sẽ bao gồm hai phần riêng biệt: front-end, back-end được giao tiếp thông qua giao thức JWT 3.6 để xác định danh tích người dùng.

FrontEnd thực hiện nhiệm vụ hiển thị giao diện cho người dùng, dữ liệu từ server (back-end) được lấy theo giao thức RESTful API [1]. Xây dựng giao diện người dùng sử dụng công nghệ ReactJS 3.2, một thư viện được sử dụng rộng rãi bởi các lợi ích của nó đem lại: tương thích với nhiều thư viện khác, dễ tổ chức, linh hoạt tối ưu hiệu năng cho tốc độ xử lý dữ liệu.

Backend chịu trách nhiệm cho việc cung cấp dịch vụ (như cơ sở dữ liệu, in ấn, truyền file, hệ thống...) cho các clients. Hệ thống sử dụng ngôn ngữ lập trình là Javascript 3.1. cùng với framework ExpressJS là một web application framework, nó cung cấp cho chúng những rất nhiều tính năng mạnh mẽ trên nền tảng web, dựa trên nền tảng NodeJS. NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

Hệ thống sử dụng hệ cơ sở quản trị dữ liệu phi quan hệ (NoSql) cụ thể là MongoDB. Đó là một hệ quản trị cơ sở giúp xây dựng một ứng dụng có khả năng thay đổi linh hoạt, hiệu năng cao, lưu được nhiều kiểu dữ liệu...

## **1.4 Bố cục đồ án**

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về khảo sát và phân tích yêu cầu hệ thống. Từ đó, xác định chức năng và đặc tả chức năng của các tác nhân trong hệ thống (phần 2.2 và 2.3): người tìm việc, nhà tuyển dụng, nhà quản lý công ty, quản trị viên. Đồng thời trong chương 2, người viết đồ án nói về các yêu cầu phi chức năng của hệ thống liên quan đến bảo mật (phần 2.4.1), vận hành (phần 2.4.2) và giao diện người dùng (phần 2.4.3).

Thông qua việc phân tích bài toán ở chương 2, chương 3 trình bày các công nghệ được sử dụng trong hệ thống bao gồm: bộ cung cụ phát triển web MERN (Mongo 3.8, Express, NodeJS 3.5, ReactJS 3.2), cùng một số thư viện đi kèm: Redux 3.3, React Query 3.4, SocketIO 3.7....

Ở Chương 4, người viết đồ án dựa trên mẫu kiến trúc Model–View–Presenter

(MVP 4.1.1), lựa chọn bộ công cụ phát triển MERN [2]. Người viết đề án thiết kế chi tiết các gói, lớp, giao diện, kiểm thử chức năng và đi vào triển khai hệ thống trên mạng LAN (Local Area Network – mạng cục bộ).

Trong Chương 5, chương này nói về các khó khăn và cách giải quyết vấn đề mà người viết đề án gặp phải trong quá trình triển khai hệ thống bao gồm: khó khăn và cách giải quyết khi lập trình giao diện (phần 5.1, 5.2), thiết kế cơ sở dữ liệu (phần 5.3, 5.4) và lưu trữ file (phần 5.5).

Chương 6 nói về kết luận những gì đã đạt được trong học phần này và hạn chế cần phải giải quyết trong hệ thống. Từ đó, nêu lên hướng phát triển tiếp theo của hệ thống.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát hiện trạng

Trong bối cảnh toàn cầu hóa, và xu thế hội nhập, dưới sự tác động mạnh mẽ của cuộc cách mạng khoa học công nghệ, đặc biệt là cuộc cách mạng công nghiệp 4.0 đang phát triển mạnh mẽ, thì vấn đề việc làm trở thành vấn đề quan trọng, với nhiều vấn đề đặt ra. Trước kia, khi công nghệ thông tin chưa được áp dụng rộng rãi trong lĩnh vực tìm kiếm việc làm, người lao động chỉ có thể tìm việc thông qua các mẫu tin được đăng trên các mặt báo, mẫu tin được dán trên các lề đường, hay cao cấp hơn là thông qua các kênh trung gian là các công ty mua giới việc làm. Những hình thức tìm kiếm việc làm đó hiện nay đã không còn xuất hiện nhiều bởi nó khá tốn kém chi phí về mặt thời gian và công sức. Hiện nay, mọi người tìm kiếm việc làm trên các trang website được sử dụng rộng rãi và phổ biến.

Người tìm việc có thể lên trang web của chính công ty để tìm kiếm công việc phù hợp với khả năng của bản thân mình. Tuy nhiên, điều này gặp trở ngại khi người dùng mất công sức trong việc tìm đúng công ty và công việc phù hợp với khả năng của mình. Cũng có một cách khác mà mọi người hay sử dụng hiện nay đó chính là thông qua các trang mạng xã hội điển hình là Facebook. Với đông đảo số lượng người dùng Facebook hiện nay thông qua các hội nhóm Facebook, người tìm việc và nhà tuyển dụng đều có thể tìm kiếm việc làm trên đó. Tuy nhiên, việc sử dụng các trang mạng xã hội vẫn còn nhiều bất cập, hạn chế, các trang mạng xã hội này mục đích chính là để giúp người dùng giao lưu, kết nối, hệ thống đó không chú trọng vào việc xây dựng hệ thống giúp tối ưu hóa việc tìm kiếm việc làm. Chính vì vậy, nó thiếu cơ chế phân loại việc làm, tìm kiếm ứng viên, lưu công việc... dẫn đến việc người dùng phải tìm kiếm một cách khá vất vả.

Cũng có một cách khác đó chính là vào các trang tìm kiếm việc làm như: TopCV, ITViec, timviecnhanh.com. Các hệ thống website này cung cấp đầy đủ các chức năng giúp nhà tuyển dụng đăng tải các công việc của mình, cùng với đó là việc phân chia các công việc theo kỹ năng, công ty một cách hợp lý giúp người lao động dễ dàng tìm kiếm các công việc mà mình mong muốn. Tuy nhiên, các hệ thống vẫn chưa cung cấp đầy đủ các thông tin hữu ích về công ty: lượt theo dõi, tổng quan về công ty, hoạt động gần đây của công ty... điều này gây khó khăn cho người lao động khi phải chuyển tab để tìm hiểu các thông tin về công ty nơi mình ứng tuyển công việc. Hơn nữa, đối với những hệ thống như trang website timviecnhanh.com còn hạn chế trong việc giúp cho người dùng thể hiện CV của mình. Từ đó, gây khó khăn cho nhà tuyển dụng khi tìm kiếm các ứng viên tiềm

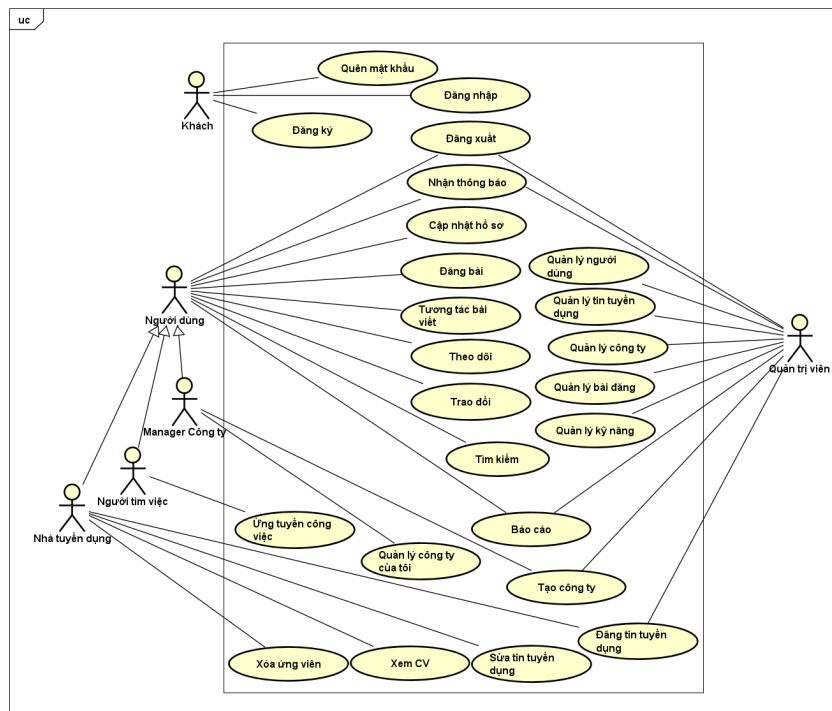
năng cho công việc của mình.

Chính vì những mặt hạn chế đó, hệ thống tìm kiếm việc làm như LinkedIn ra đời giúp cải thiện đáng kể hiệu quả tìm kiếm việc làm cho người lao động. Trong hệ thống LinkedIn, người dùng được cung cấp đa dạng các tính năng: có thể theo dõi hoạt động của các công ty mình muốn, gợi ý việc làm dựa theo kỹ năng của người dùng, đưa ra ứng viên tiềm năng cho nhà tuyển dụng,... Tuy nhiên, hệ thống vẫn chưa phân biệt rạch ròi trong việc trao đổi giữa người dùng và công việc hay xử lý các vấn đề phát sinh trong khi nhà tuyển dụng gặp sự cố.

Dựa trên những điều phân tích ở trên, ta có thể thấy, trên thị trường hiện nay có rất nhiều các kênh thông tin tìm kiếm việc làm mà ở đó người dùng có thể tìm kiếm công việc phù hợp với năng lực của bản thân mình. Tuy nhiên như đã phân tích, các hệ thống đó ít nhiều còn tồn tại các mặt hạn chế đối với người sử dụng. Từ đó, người viết đề án nhận thấy việc xây dựng một hệ thống tạo môi trường tốt cho việc tương tác giữa người lao động (tìm việc) và nhà tuyển dụng là cần thiết. Do đó, người viết đề án chọn đề tài là “Xây dựng hệ thống mạng xã hội tìm kiếm việc làm” nhằm tìm cách khắc phục một số hạn chế đã nêu ra và đảm bảo các ưu điểm của các hệ thống hiện có.

## 2.2 Tổng quan chức năng

### 2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Sơ đồ Usecase tổng quan

STT	Tên tác nhân	Mô tả tác nhân
1	Khách	Là người chưa đăng nhập vào hệ thống các chức năng đăng nhập, đăng ký để trở thành người dùng của hệ thống, khách cũng có chức năng quên mật khẩu giúp thay đổi mật khẩu khi họ quên được gửi qua email mà họ đã đăng ký với hệ thống từ trước
2	Người dùng	Là người sử dụng hệ thống, có thể thực hiện các hoạt động như: cập nhật thông tin hồ sơ cá nhân, đăng bài, tương tác bài viết, trao đổi, theo dõi, tìm kiếm, nhận thông báo...
3	Manager công ty	Người quản lý là người dùng của hệ thống được cung cấp thêm các chức năng như: tạo công ty, quản lý công ty của họ.
4	Người tìm việc	Người tìm việc là người dùng của hệ thống được cung cấp thêm các chức năng như: ứng tuyển công việc, trao đổi công việc,...
5	Nhà tuyển dụng	Nhà tuyển dụng là người dùng của hệ thống được cung cấp thêm các chức năng như: đăng tin tuyển dụng, xem CV của ứng viên, xóa ứng viên...
6	Quản trị viên	Là người điều hành hệ thống được cung cấp các chức năng: nhận thông báo, quản lý người dùng, quản lý công việc, quản lý công ty, quản lý kỹ năng, quản lý bài đăng.

**Bảng 2.1:** Danh sách tác nhân

STT	Mã Use-case	Tên Use-case	Mô tả Usecase	Tác nhân tương tác
1	UC01	Đăng nhập	Cho phép khách có tài khoản đăng nhập trở thành người dùng của hệ thống	Khách
2	UC02	Đăng ký	Cho phép khách đăng ký vào hệ thống	Khách
3	UC03	Quên mật khẩu	Cho phép khách lấy lại tài khoản thông qua email mà khách đã đăng ký với hệ thống từ trước đó	Khách

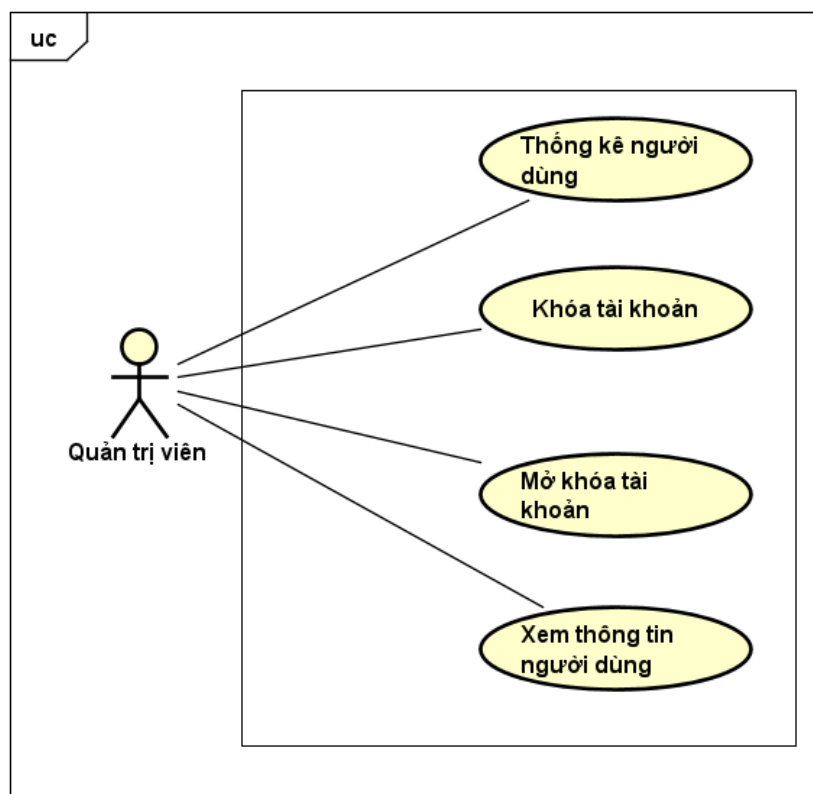
4	UC04	Đăng xuất	Đưa tác nhân thoát khỏi hệ thống	Quản trị viên, người dùng
5	UC05	Nhận thông báo	Cho phép hệ thống thông báo những sự kiện mà tác nhân quan tâm theo thời gian thực được phân loại theo loại: theo dõi người dùng, tạo công ty,...	Quản trị viên, người dùng
6	UC06	Quản lý người dùng	Cho phép quản trị viên có thể xem thông tin về tình trạng của người dùng, thống kê số lượng người dùng đã đăng ký vào hệ thống	Quản trị viên
7	UC07	Quản lý bài đăng	Cho phép quản trị viên theo dõi tình trạng bài đăng lượt tương tác và loại bài đăng của người dùng.	Quản trị viên
8	UC08	Quản lý công ty	Cho phép quản trị viên thống kê, vô hiệu các công ty có trong hệ thống	Quản trị viên
9	UC09	Quản lý tin tuyển dụng	Cho phép quản trị viên thay đổi trạng thái của tin tuyển dụng, thống kê số lượng tin tuyển dụng, các thông tin như số lượng người ứng tuyển, số lượng người theo dõi tin tuyển dụng	Quản trị viên
10	UC10	Quản lý kỹ năng	Cho phép quản trị viên thống kê kỹ năng được xây dựng từ người dùng trong hệ thống	Quản trị viên
11	UC11	Cập nhật thông tin	Cho phép người dùng cập nhật thông tin hồ sơ cá nhân: kỹ năng, học vấn, thông tin tổng quan...	Người dùng

12	UC12	Đăng bài	Cho phép người dùng đăng tải các hoạt động gần đây của bản thân, công ty.	Người dùng
13	UC13	Tương tác bài viết	Cho phép người dùng tương tác với bài đăng.	Người dùng
14	UC14	Theo dõi	Cho phép người dùng theo dõi: người dùng khác, công ty, công việc,	Người dùng
15	UC15	Trao đổi	Cho phép trao đổi giữa người dùng với người dùng hoặc người dùng với công việc	Người dùng
16	UC16	Tìm kiếm	Cho phép người dùng tìm kiếm: người dùng khác, công ty, công việc.	Người dùng
17	UC17	Tạo công ty	Cho phép quản lý tạo công ty của riêng mình và phải được người quản trị viên kiểm duyệt.	Manager công ty, quản trị viên
18	UC18	Quản lý công ty của tôi	Cho phép quản lý của công ty quản lý công ty của họ	Manager công ty
19	UC19	Ứng tuyển công việc	Cho phép người tìm việc ứng tuyển công việc.	Người tìm việc
20	UC20	Đăng tin tuyển dụng	Cho phép nhà tuyển dụng tạo tin tuyển dụng và phải được chấp nhận bởi quản trị viên.	Nhà tuyển dụng, quản trị viên.
21	UC21	Xem CV	Cho phép nhà tuyển dụng xem CV của ứng viên.	Nhà tuyển dụng
22	UC22	Xóa ứng viên	Cho phép nhà tuyển dụng xóa ứng viên không phù hợp	Nhà tuyển dụng
23	UC23	Sửa tin tuyển dụng	Cho phép nhà tuyển dụng sửa tin tuyển dụng của mình trong khi công việc bị vô hiệu.	Nhà tuyển dụng

24	UC24	Báo cáo	Cho phép người dùng báo cáo bài đăng, tin tuyển dụng không phù hợp với quản trị viên	Người dùng, quản trị viên
----	------	---------	--	---------------------------

**Bảng 2.2:** Danh sách Usecase

### 2.2.2 Biểu đồ use case phân rã “Quản lý người dùng”

**Hình 2.2:** Phân rã Usecase "Quản lý người dùng"

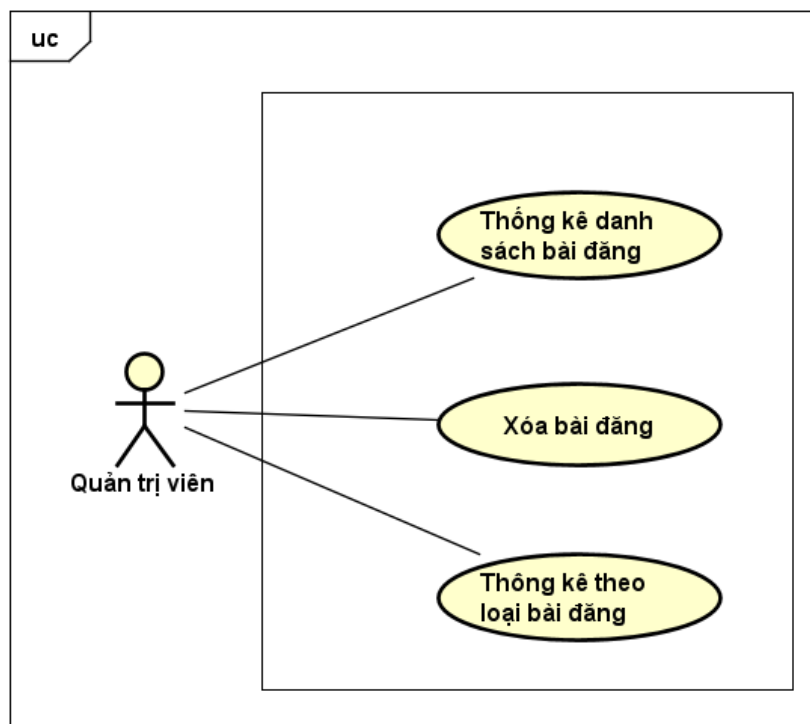
STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC06-1	Thống kê người dùng	Thống kê những người dùng hiện có trong hệ thống	Quản trị viên
2	UC06-2	Khóa tài khoản	Khóa tài khoản của người dùng	Quản trị viên
3	UC06-3	Mở khóa tài khoản	Mở khóa tài khoản của người dùng	Quản trị viên



4	UC06-4	Xem thông tin người dùng	Xem thông tin cá nhân của người dùng: hồ sơ, hoạt động gần đây của người dùng.	Quản trị viên
---	--------	--------------------------	--	---------------

**Bảng 2.3:** Usecase "Quản lý người dùng"

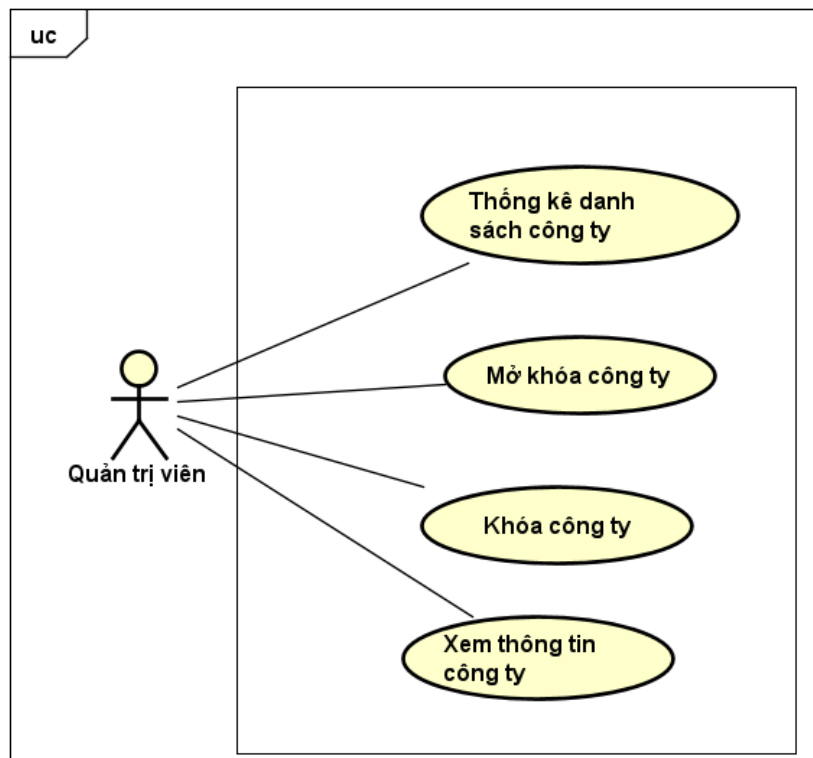
### 2.2.3 Biểu đồ use case phân rã “Quản lý bài đăng”

**Hình 2.3:** Phân rã Usecase "Quản lý bài đăng"

STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC07-1	Thống kê danh sách bài đăng	Thống kê bài đăng hiện có: tên, số lượng ảnh,... và lọc bài đăng	Quản trị viên
2	UC07-2	Xóa bài đăng	Xóa bài đăng, lượt thích, bình luận của bài đăng	Quản trị viên
3	UC07-3	Thống kê theo loại bài đăng	Thống kê theo loại: công ty, cá nhân.	Quản trị viên

**Bảng 2.4:** Usecase "Quản lý bài đăng"

### 2.2.4 Biểu đồ use case phân rã “Quản lý công ty”

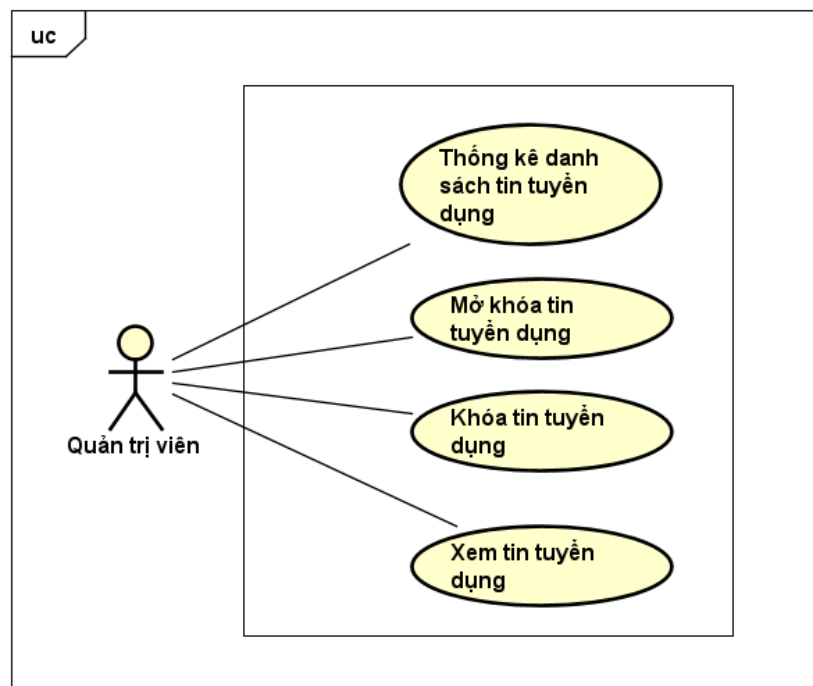


**Hình 2.4:** Phân rã Usecase "Quản lý công ty"

STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC08-1	Thống kê danh sách công ty	Thống kê danh sách công ty có trong hệ thống	Quản trị viên
2	UC08-2	Mở khóa công ty	Mở khóa công ty, công việc của công ty, phòng trao đổi của nhà tuyển dụng công ty.	Quản trị viên
3	UC08-3	Khóa công ty	Khóa công ty, công việc của công ty, phòng trao đổi của nhà tuyển dụng công ty	Quản trị viên
4	UC08-4	Xem thông tin công ty	Xem mô tả chi tiết về công ty	Quản trị viên

**Bảng 2.5:** Usecase "Quản lý công ty"

### 2.2.5 Biểu đồ use case phân rã “Quản lý tin tuyển dụng”

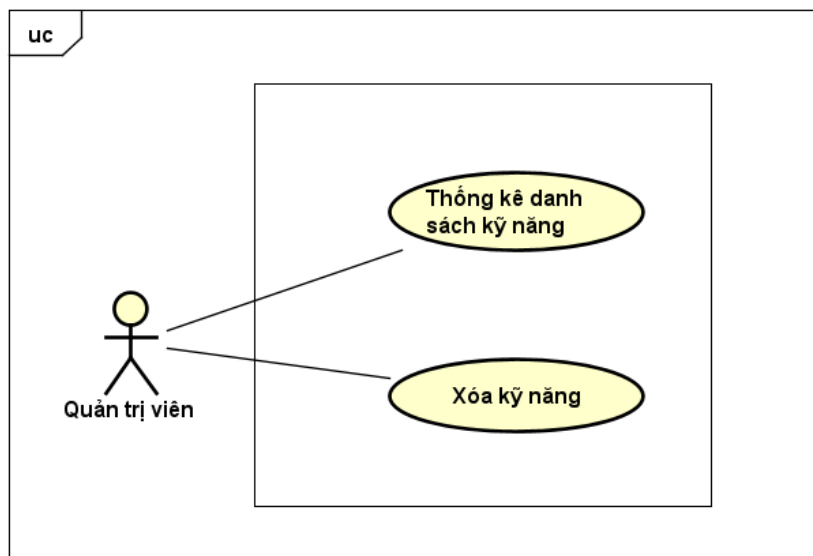


**Hình 2.5:** Phân rã Usecase "Quản lý tin tuyển dụng"

STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC09-1	Thống kê danh sách tin tuyển dụng	Thống kê danh sách tin tuyển dụng có trong hệ thống: theo tên, số lượng người theo dõi,...	Quản trị viên
2	UC09-2	Mở khóa tin tuyển dụng	Mở khóa tin tuyển dụng cho người tìm việc ứng tuyển	Quản trị viên
3	UC09-3	Khóa tin tuyển dụng	Khóa tin tuyển dụng không cho người tìm việc ứng tuyển	Quản trị viên

**Bảng 2.6:** Usecase "Quản lý tin tuyển dụng"

### 2.2.6 Biểu đồ use case phân rã “Quản lý kỹ năng”

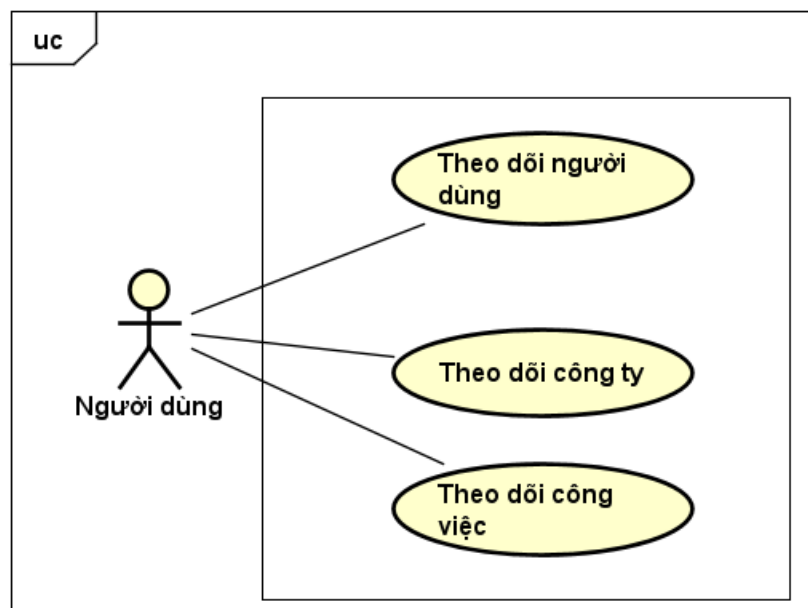


**Hình 2.6:** Phân rã Usecase "Quản lý kỹ năng"

STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC10-1	Thống kê danh sách kỹ năng	Thống kê danh sách kỹ năng có trong hệ thống: tên, số lượng công việc, số lượng người dùng,...	Quản trị viên
2	UC10-2	Xóa kỹ năng	Xóa kỹ năng của người dùng, công việc liên quan	Quản trị viên

**Bảng 2.7:** Usecase "Quản lý kỹ năng"

### 2.2.7 Biểu đồ use case phân rã “Theo dõi”

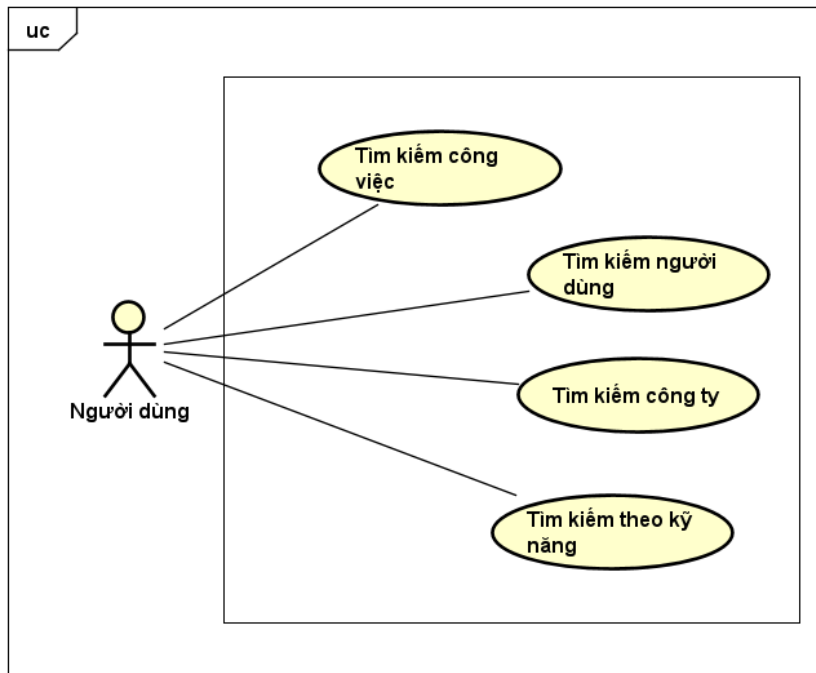


**Hình 2.7:** Phân rã Usecase "Theo dõi"

STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC14-1	Theo dõi người dùng	Theo dõi người dùng để nhận thông tin mới nhất của họ	Người dùng
2	UC14-2	Theo dõi công ty	Theo dõi công ty để nhận thông tin mới nhất của họ	Người dùng
3	UC14-3	Theo dõi công việc	Theo dõi công việc để nhận thông tin, lưu công việc.	Người dùng

**Bảng 2.8:** Usecase "Theo dõi"

### 2.2.8 Biểu đồ use case phân rã “Tìm kiếm”

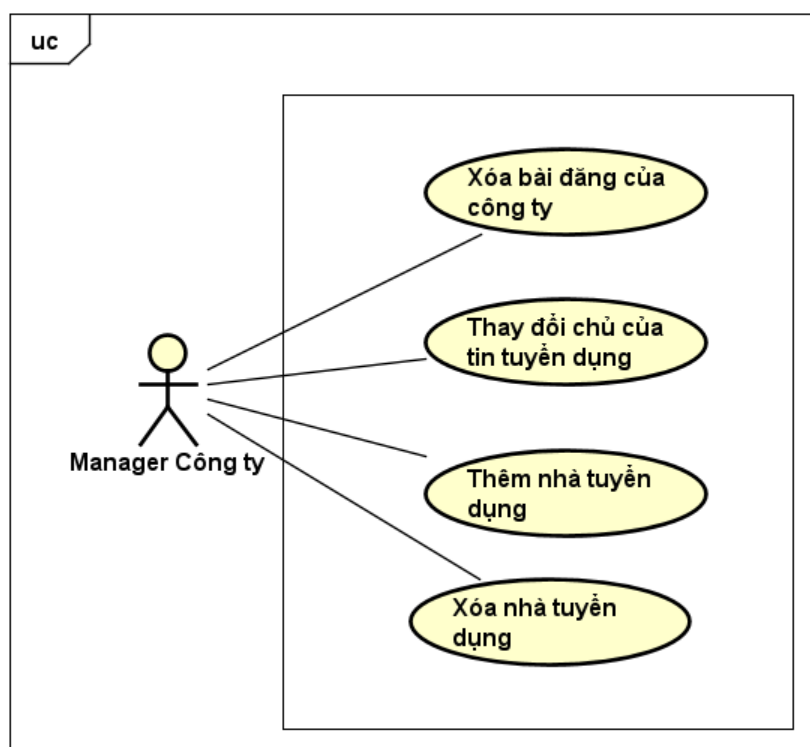


**Hình 2.8:** Phân rã Usecase "Tìm kiếm"

STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC16-1	Tìm kiếm công việc	Tìm kiếm công việc theo từ khóa hoặc theo đề xuất dựa trên hồ sơ người dùng	Người dùng
2	UC16-2	Tìm kiếm người dùng	Tìm kiếm người dùng dựa theo từ khóa hoặc dựa trên hồ sơ của người dùng	Người dùng
3	UC16-3	Tìm kiếm công ty	Tìm kiếm công ty dựa theo từ khóa hoặc dựa trên hồ sơ của người dùng	Người dùng
4	UC16-4	Tìm kiếm theo kỹ năng	Tìm kiếm theo kỹ năng các người dùng, công ty, công việc phù hợp với kỹ năng	Người dùng

**Bảng 2.9:** Usecase "Tìm kiếm"

### 2.2.9 Biểu đồ use case phân rã “Quản lý công ty của tôi”



**Hình 2.9:** Phân rã Usecase "Quản lý công ty của tôi"

STT	Mã Use-case	Tên Use-Case	Mô tả Usecase	Tác nhân tương tác
1	UC18-1	Xóa bài đăng của công ty	Xóa bài đăng không phù hợp hay sai nội dung định hướng của công ty	Manager công ty
2	UC18-2	Thay đổi chủ của tin tuyển dụng	Thay đổi quyền sở hữu của tin tuyển dụng thông báo đến các bên liên quan.	Manager công ty
3	UC18-3	Thêm nhà tuyển dụng	Thêm nhà tuyển dụng cho công ty	Manager công ty
4	UC18-4	Xóa nhà tuyển dụng	Xóa nhà tuyển dụng ra khỏi công ty lúc này họ sẽ trở thành người tìm việc	Manager công ty

**Bảng 2.10:** Usecase "Quản lý công ty của tôi"

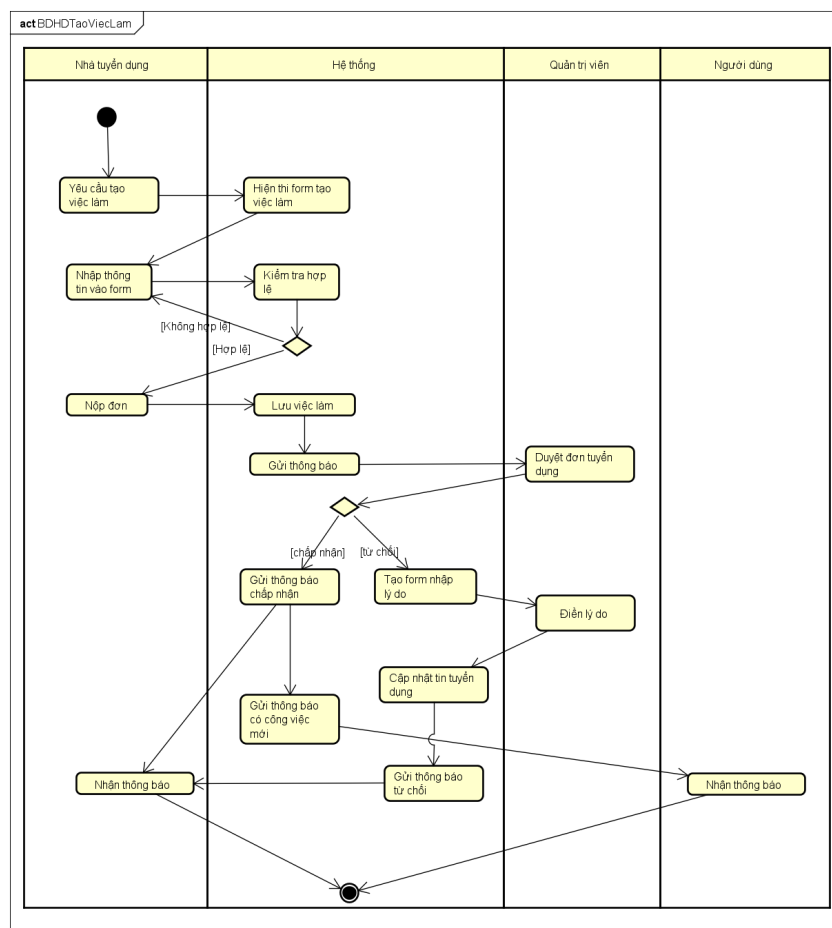
**2.2.10 Quy trình nghiệp vụ****a, Quy trình nghiệp vụ đăng tin tuyển dụng**

Quy trình đăng tin tuyển dụng bao gồm các nhóm người dùng tham gia là nhà tuyển dụng, quản trị viên, và người dùng. Điều này để đảm bảo dữ liệu việc làm được kiểm soát một cách chặt chẽ, chính xác và để cho mọi người có thể nắm bắt được thông tin tuyển dụng một cách nhanh nhất đầy đủ và chính xác nhất.

	<b>Đăng tin tuyển dụng</b>
<b>Mục tiêu quy trình</b>	Đăng tin tuyển dụng việc làm lên hệ thống
<b>Phạm vi quy trình</b>	Nhà tuyển dụng, quản trị viên, người dùng
<b>Đầu vào quy trình</b>	<ul style="list-style-type: none"> <li>• Thông tin tuyển dụng</li> <li>• Thông tin người đăng tin tuyển dụng</li> </ul>
<b>Luồng quy trình</b>	<ul style="list-style-type: none"> <li>• Nhà tuyển dụng chọn chức năng đăng tin tuyển dụng</li> <li>• Hệ thống hiển thị giao diện đăng tin tuyển dụng</li> <li>• Nhà tuyển dụng nhập các thông tin cần thiết cho việc làm</li> <li>• Hệ thống xác thực đầy đủ thông tin, nếu thiếu, hiển thị lỗi và yêu cầu người dùng sửa lại</li> <li>• Việc làm được đăng lên hệ thống với trạng thái bị vô hiệu hóa, hệ thống sẽ gửi thông báo đến quản trị viên để yêu cầu xác nhận tin tuyển dụng.</li> <li>• Quản trị viên xác nhận hoặc từ chối tin tuyển dụng sau khi kiểm tra tính đúng đắn của tin</li> <li>• Trong trường hợp chấp nhận tin tuyển dụng, hệ thống sẽ gửi thông báo đến nhà tuyển dụng thông báo tin tuyển dụng đã được xác nhận đồng thời gửi đến các người dùng đăng theo dõi công ty có tin tuyển dụng đó. Trong trường hợp thất bại, quản trị viên viết lý do, sau đó, hệ thống gửi thông báo từ chối đến nhà tuyển dụng</li> </ul>
<b>Đầu ra quy trình</b>	Gửi thông báo xác nhận hoặc từ chối tin tuyển dụng
<b>Quy trình đặc biệt</b>	Không có

**Bảng 2.11:** Quy trình nghiệp vụ đăng tin tuyển dụng





Hình 2.10: Biểu đồ hoạt động đăng tin tuyển dụng

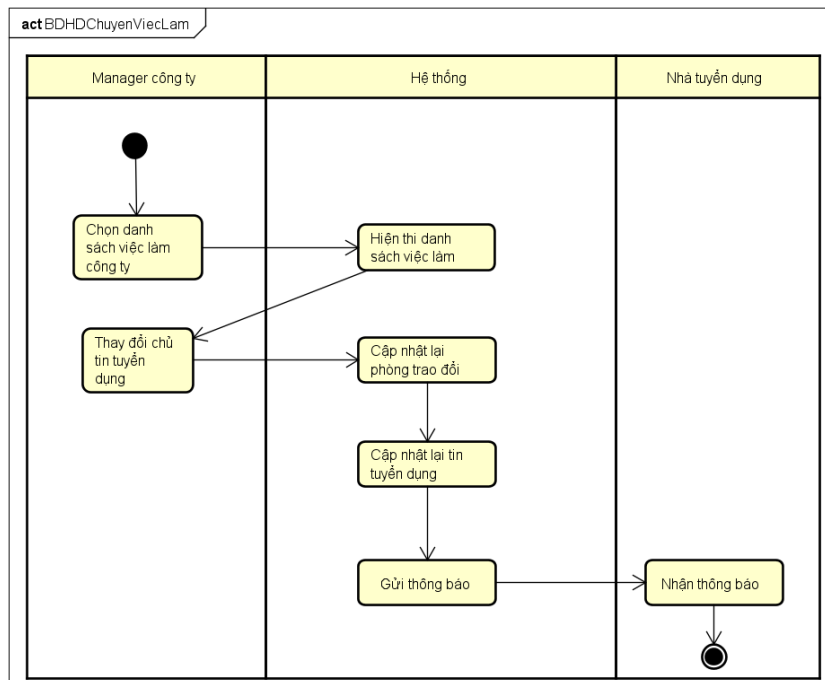
### b, Quy trình nghiệp vụ thay đổi chủ của tin tuyển dụng

Quy trình thay đổi chủ của tin tuyển dụng bao gồm các nhóm người dùng tham gia là nhà tuyển dụng, manager công ty, và người dùng giúp cho công ty ứng phó được với các tình huống không may khi nhà tuyển dụng của công ty gặp sự cố.

	Thay đổi chủ của tin tuyển dụng
<b>Mục tiêu quy trình</b>	Chuyển công việc từ nhà tuyển dụng này đến nhà tuyển dụng khác thuộc cùng một công ty do manager công ty thực hiện
<b>Phạm vi quy trình</b>	Nhà tuyển dụng, manager công ty
<b>Đầu vào quy trình</b>	<ul style="list-style-type: none"> <li>Tin tuyển dụng</li> <li>Nhà tuyển dụng cũ</li> <li>Nhà tuyển dụng mới</li> </ul>

<b>Luồng quy trình</b>	<ul style="list-style-type: none"> <li>• Manager công ty chọn xem danh sách công việc của công ty</li> <li>• Hiện thị danh sách việc làm của công ty</li> <li>• Manager công ty thực hiện chuyển giao công việc</li> <li>• Hệ thống cập nhật lại phòng trao đổi của công việc đó</li> <li>• Hệ thống cập nhật lại thông tin chủ của tin tuyển dụng</li> <li>• Hệ thống gửi thông báo đến các nhà tuyển dụng liên quan</li> </ul>
<b>Đầu ra quy trình</b>	Gửi thông báo thay đổi nhà tuyển dụng thành công
<b>Quy trình đặc biệt</b>	Không có

**Bảng 2.12:** Quy trình nghiệp vụ thay đổi chủ của tin tuyển dụng



**Hình 2.11:** Biểu đồ hoạt động thay đổi chủ tin tuyển dụng

## 2.3 Đặc tả chức năng

### 2.3.1 Đặc tả use case “Quên mật khẩu”

<b>Mã Use case</b>	UC03	<b>Tên Use case</b>	Quên mật khẩu
<b>Tác nhân</b>	Khách		
<b>Tiền điều kiện</b>	Đã vào trang quên mật khẩu		

<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1	Khách	Nhập email cần lấy mật khẩu (*)
	2	Khách	Gửi yêu cầu lấy mật khẩu
	3	Hệ thống	Kiểm tra thông tin có đã nhập đúng trường dữ liệu hay chưa
	4	Hệ thống	Kiểm tra tính hợp lệ
	5	Hệ thống	Gửi email reset mật khẩu
	6	Khách	Click đường dẫn được gửi qua email
	7	Hệ thống	Hiển thị form đặt lại mật khẩu
	8	Khách	Điền biểu mẫu đặt lại mật khẩu (**)
	9	Khách	Nộp biểu mẫu
	10	Hệ thống	Kiểm tra thông tin có đã nhập đúng trường dữ liệu hay chưa
	11	Hệ thống	Kiểm tra tính hợp lệ
	12	Hệ thống	Thực hiện đặt lại mật khẩu cho khách, xóa token
	13	Hệ thống	Thông báo đặt lại mật khẩu thành công
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	3a	Hệ thống	Thông báo lỗi: cần nhập đầy đủ trường thông tin bắt buộc
	4a	Hệ thống	Thông báo lỗi email không tồn tại
	5a	Hệ thống	Thông báo lỗi khi gửi email
	10a	Hệ thống	Thông báo nhập thiếu hoặc sai trường dữ liệu
	11a	Hệ thống	Báo token hết hạn
<b>Hậu điều kiện</b>	Không có		

**Bảng 2.13:** Đặc tả usecase “Quên mật khẩu”

Dữ liệu đầu vào (\*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
-----	----------------	-------	----------	------------------	-------

1	Email	Email của tài khoản	Có	Định dạng email	alice@gmail.com
---	-------	---------------------	----	-----------------	-----------------

Dữ liệu đầu vào (\*\*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	password	Mật khẩu đặt lại	Có	Từ 6 đến 20 ký tự	aclie12312
2	confirm password	Kiểm tra lại mật khẩu	Có	Đúng với trường password	aclie12312

### 2.3.2 Đặc tả use case “Cập nhật thông tin”

Mã Use case	UC11	Tên Use case	Cập nhật thông tin
Tác nhân	Người dùng		
Tiền điều kiện	Đang ở trong profile của bản thân		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Chọn mục cần cập nhật
	2.	Hệ thống	Hiển thị giao diện cập nhật thông tin
	4.	Người dùng	Nhập thông tin cần cập nhật (*)
	5.	Người dùng	Nhấn nút “Nộp”
	6.	Hệ thống	Kiểm tra tính hợp lệ
	7.	Hệ thống	Cập nhật lại trang profile của người dùng
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	6a.	Hệ thống	thông báo lỗi: Hiển thị thông báo các trường thông tin còn thiếu hoặc không hợp lệ
Hậu điều kiện	Không có		

**Bảng 2.16:** Đặc tả usecase “Cập nhật thông tin”

Dữ liệu đầu vào (\*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	name	Tên người dùng được hiển thị trên hệ thống	Có	Phải là duy nhất trong hệ thống	phongvu
2	email	Email của người dùng	Có	Phải là duy nhất trong hệ thống	alice@gmail.com
3	avatar	ảnh đại diện của người dùng	Không	Chuỗi string	http://localhost/abc.png
4	about	Giới thiệu thông quan về bản thân mình	Không	Chuỗi tring ít hơn 1000 ký tự	Hello 123
5	educations	Trình độ văn hóa mà người dùng đạt được	Không	Là một mảng các trường đào tạo của người dùng	[{school: "HUST", degree: "Cu Nhan", field_of_style : "Software", start_date: 08/2018, end_date: 08/2022}]
6	skills	Kỹ năng mà người dùng có	Không	Một mảng các kỹ năng	[ "C#", "Java"]
7	experiences	Kinh nhiệm việc làm của người dùng	Không	Một mảng lưu trữ kinh nhiệm việc làm	[{company: "ABC", position: "Junior", desc : "viết gì đó??", start_date: 08/2018, end_date: 08/2022}]

8	languages	Ngoại ngữ của người dùng	Không	Một mảng chứa ngôn ngữ	[English, Việt-nameese]
9	social	Đường liên kết đến các trang website của người dùng	Không	Object (mỗi phần tử lưu trữ đường dẫn đến trang mạng xã hội)	{youtube: http://abc, facebook: http://abc, }

### 2.3.3 Đặc tả use case “Ứng tuyển công việc”

Mã Use case	UC19	Tên Use case	Ứng tuyển công việc
<b>Tác nhân</b>	Người tìm việc		
<b>Tiền điều kiện</b>	Đang ở trang xem chi tiết công việc hoặc tìm kiếm, đề xuất công việc		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người tìm việc	Nhấn nút ứng tuyển công việc
	2.	Hệ thống	Hiển thị giao diện ứng tuyển công việc
	3.	Người tìm việc	Nhập các trường theo yêu cầu (*)
	4.	Người tìm việc	Nhấn nút nộp
	5.	Hệ thống	Kiểm tra tính hợp lệ các trường đã điền
	6.	Hệ thống	Kiểm tra tính hợp lệ của đơn ứng tuyển
	7.	Hệ thống	Thông báo ứng tuyển thành công
	8.	Hệ thống	Gửi thông báo đến nhà tuyển dụng chịu trách nhiệm cho công việc

Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a.	Hệ thống	thông báo lỗi: nhập thiếu hoặc sai định dạng các trường
	6a	Hệ thống	Thông báo lỗi: quá hạn ứng tuyển hoặc đã ứng tuyển rồi
	7a	Hệ thống	Lỗi upload file ứng tuyển
<b>Hậu điều kiện</b>	Cập nhật danh sách đã ứng tuyển		

**Bảng 2.18:** Đặc tả usecase “Ứng tuyển công việc”

Dữ liệu đầu vào (\*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	email	Email hiện tại người tìm việc đang sử dụng	Có	Định dạng email	bob@gmail.com
2	phone	Số điện thoại hiện tại người tìm việc sử dụng	Có	Định dạng số điện thoại	0386851140
3	CV	Đơn ứng tuyển của ứng viên	Có	Chuỗi string có định dạng đường dẫn	http://abc.com

**2.3.4 Đặc tả use case “Đăng tin tuyển dụng”**

<b>Mã Use case</b>	UC20	<b>Tên Use case</b>	Đăng tin tuyển dụng
<b>Tác nhân</b>	Nhà tuyển dụng		
<b>Tiền điều kiện</b>	Đang ở trang chủ của hệ thống		

<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Nhà tuyển dụng	Nhấn nút tạo “Việc làm”
	2.	Hệ thống	Hiển thị giao diện tạo việc làm
	3.	Nhà tuyển dụng	Nhập các trường theo yêu cầu (*)
	4.	Nhà tuyển dụng	Nhấn nút nộp
	5.	Hệ thống	Kiểm tra tính hợp lệ các trường đã điền
	6	Hệ thống	Kiểm tra tính hợp lệ của tin tuyển dụng
	7	Hệ thống	Thông báo chờ quản trị viên duyệt
	8	Hệ thống	Gửi thông báo đến quản trị viên có việc là vừa tạo
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	5a.	Hệ thống	Thông báo lỗi: nhập thiếu hoặc sai định dạng các trường
	6a	Hệ thống	Thông báo lỗi: sai hạn chót, công ty bị khóa...
<b>Hậu điều kiện</b>	Không		

**Bảng 2.20:** Đặc tả usecase “Đăng tin tuyển dụng”

Dữ liệu đầu vào (\*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	title	Nhan đề của tin tuyển dụng	Có	Chuỗi từ 3 đến 200 ký tự	lalafl
2	salary	Lương của công việc	Có	Định dạng số hoặc khoảng lương	2000000
3	job_type	Kiểu công việc	Có	Một trong 3 giá trị: fulltime, parttime, freelance	fulltime



4	numbers	Số lượng tuyển	Có	Chữ số	10
5	responsibilities	Mô tả trách nhiệm công việc	Có	Ít nhất 3 ký tự	Thực hiện phân tích nghiệp vụ các hệ thống, vẽ biểu đồ
6	requirements	Yêu cầu của công việc	Có	Ít nhất 10 ký tự	Biết một ngôn ngữ lập trình
7	location	Địa điểm làm việc	Có	Chuỗi	Hà nội
8	experience	Kinh nghiệm đã trải qua	Có	Chuỗi	1-3 năm kinh nghiệm
9	rank	Chức vụ	Có	Chuỗi	Intern
10	other	Thông tin khác	Không	Chuỗi	Yêu cầu biết chơi đàn
11	deadline	Hạn chót ứng tuyển	Có	Ngày	10/11/2022
12	categories	Kỹ năng của công việc	Có	Mảng kỹ năng	[Java, C#]

## 2.4 Yêu cầu phi chức năng

### 2.4.1 Yêu cầu bảo mật

Hệ thống phải được vận hành một cách chính xác, đáng tin cậy. Các tác nhân trong hệ thống được phân quyền rõ ràng đảm bảo mỗi đối tượng chỉ được thực hiện phần việc mà mình có quyền hạn. Hệ thống có khả năng ngăn chặn phòng ngừa tấn công từ bên ngoài để bảo đảm dữ liệu người dùng. Mật khẩu lưu trong cơ sở dữ liệu phải được mã hóa sử dụng thuật toán của thư viện argon2 [3], các thông tin cá nhân phải được đảm bảo tính riêng tư.

### 2.4.2 Yêu cầu vận hành

Hệ thống được xây dựng dựa trên nền tảng ứng dụng website, chính vì vậy, hệ thống cần đảm bảo chạy tốt trên bất cứ ứng dụng website nào (Google Chorm, Cốc Cốc,...). Hệ thống cần phải hoạt động ổn định 24/7, hoạt động 500 giờ liên tục không lỗi. Hệ thống có thể hoạt động trở lại bình thường trong vòng 2 giờ sau khi xảy ra lỗi. Thời gian đáp ứng tối đa của hệ thống (nếu không được nêu rõ) là 2 giây

lúc bình thường hoặc 5 giây lúc cao điểm. Hệ thống hỗ trợ 1000 người truy xuất vào cơ sở dữ liệu cùng lúc.

#### **2.4.3 Yêu cầu giao diện**

Hệ thống có giao diện đồ họa, đảm bảo dễ sử dụng, thân thiện với sử dụng. Bộ cục hệ thống cần được phân chia rõ ràng, giúp người sử dụng hiểu và thao tác dễ dàng. Thiết kế giao diện cần có độ tương phản giữa màu nền và màu chữ, nội dung bảng biểu, thanh điều hướng chú trọng sử dụng các hình vẽ và những từ ngữ cô đọng, dễ hiểu, giúp người sử dụng nắm bắt nhanh nhất ý nghĩa. Ngôn ngữ trong hệ thống được kiểm soát, xét duyệt bởi quản trị viên đảm bảo tính lịch sự, văn phong rõ ràng, không chứa những từ ngữ phản cảm, gây kích động.

## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 3.1 Ngôn ngữ lập trình Javascript

Javascript [4] viết tắt là JS được bắt đầu phát triển từ năm 1995 tại Netscape Communications được phát triển bởi Brendan Eich. JavaScript là một trình thông dịch ngôn ngữ lập trình chính thức được nhúng bên trong trình duyệt web, làm cho website của bạn trở lên sinh động, lôi cuốn người xem. Javascript được hoạt động khi HTML và CSS phân tích hoàn tất. Khi điều này được thực hiện, JavaScript được thực thi theo thứ tự chúng được tìm thấy trên trang web: các biến và hàm được xác định, các lời gọi hàm được thực thi, trình xử lý sự kiện được kích hoạt,... Các hoạt động này dẫn đến việc DOM được cập nhật bởi JavaScript và được hiển thị ngay lập tức bởi trình duyệt.

Tuy nhiên vào năm 2009, với sự xuất hiện của Node.js [5], Javascript giờ đây không những có thể thông dịch và chạy ở browser, mà nó còn có thể chạy được ở phía server-side thông qua trình biên dịch là V8 Engine [6].

Cho đến ngày nay, Javascript đã trở thành ngôn ngữ phổ biến, có một cộng đồng phát triển to lớn nhờ các ưu điểm của nó đem lại. Javascript ở phía client giúp chúng ta xử lý nhiều việc như: validation (xác thực), filter (lọc) và pagination (phân trang)... Từ đó, phía server sẽ tiết kiệm được lượng lớn lưu lượng khi client giao tiếp với server. JS được đánh giá là ngôn ngữ dễ học nhất hiện nay. JS hoạt động được trên nhiều trình duyệt, nền tảng, mã JS nhanh hơn và nhẹ hơn các ngôn ngữ khác.

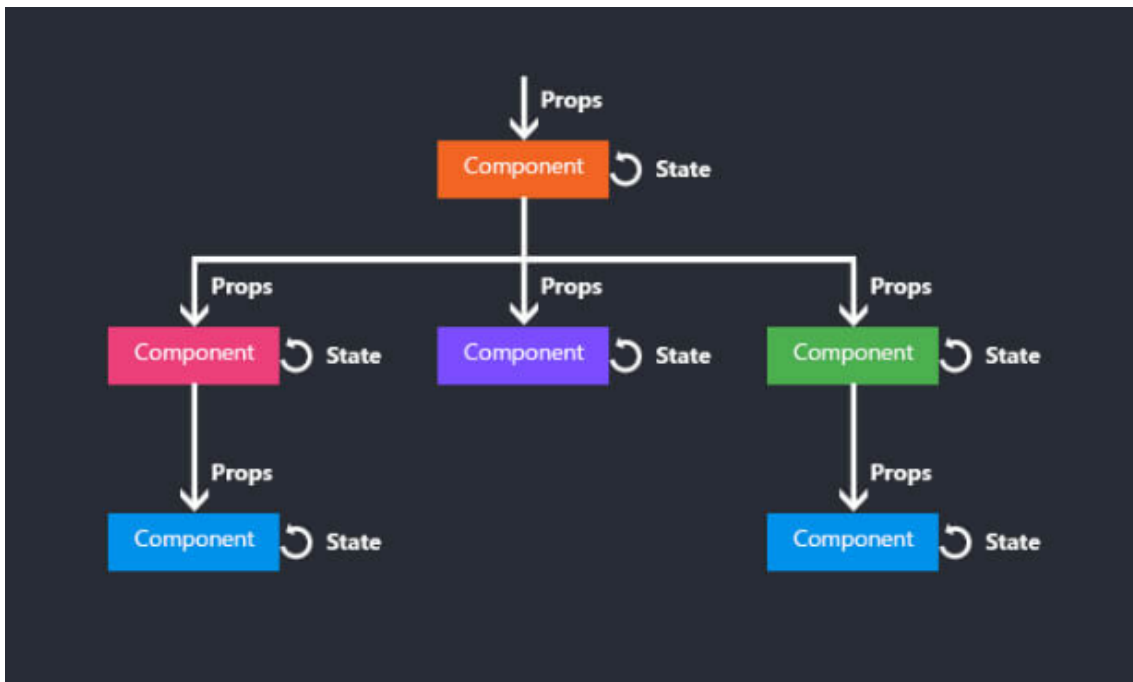
Tuy nhiên, ngôn ngữ nào cũng có những ưu điểm và hạn chế của nó, JS không ngoại lệ. JS dễ bị khai thác, có thể được dùng để thực thi mã độc trên máy tính của người dùng.

### 3.2 Công nghệ lập trình giao diện ReactJS

ReactJS (hay React, React.js) là một thư viện JavaScript front-end mã nguồn mở miễn phí để xây dựng giao diện người dùng [7]. React được phát triển và duy trì bởi công ty Facebook [8]. Nó khuyến khích việc tạo ra các thành phần giao diện người dùng có thể tái sử dụng, hiển thị dữ liệu thay đổi theo thời gian. React được sử dụng làm View (V) trong mô kiến trúc Model-View-Controller [9](MVC) .

React được xây dựng xung quanh các component, chứ không dùng template như

các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo. Có hai khái niệm cơ bản trong React là: state và props.



**Hình 3.1:** Cấu trúc, thành phần của ReactJS

Nhìn vào hình 3.1, props được truyền qua các component giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến. Trong khi đó, state là khả biến, thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật giao diện người dùng (UI). Về cơ bản, việc xây dựng một ứng dụng MVC phía client với ràng buộc dữ liệu 2 chiều (2 way data-binding) là khá đơn giản.

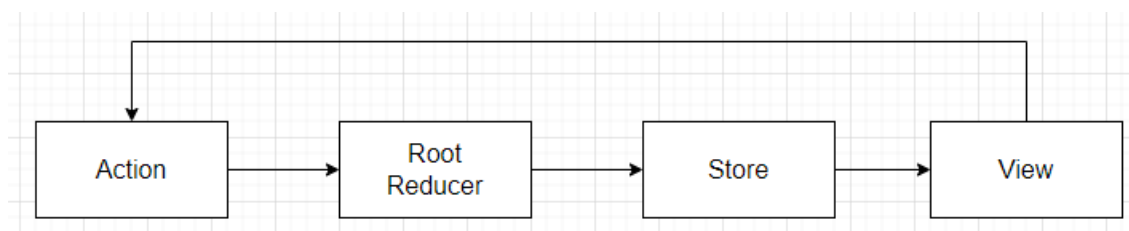
Tuy nhiên nếu dự án ngày càng mở rộng, nhiều tính năng hơn, làm cho việc bảo trì dự án gặp khó khăn, đồng thời hiệu năng cũng bị giảm. Bạn cũng có thể giải quyết vấn đề đó bằng các thư viện khác như Backbone.js [10] hay Angular.js [11]. React ra đời sau AngularJS, nó sinh ra để dành cho các ứng dụng lớn dễ dàng quản lý và mở rộng. Mục tiêu chính của React là nhanh, đơn giản, hiệu năng cao và dễ dàng mở rộng. Trong hệ thống này, React sẽ giúp phân tách các thành phần giao

diện để dễ quản lý tổ chức và tái sử dụng.

### 3.3 Công nghệ Redux

Như đã phân tích ở trên, React xây dựng từ các thành phần riêng rẽ được gọi là component. Các state hoặc props có nhiệm vụ kiểm soát trạng thái của component. Tuy nhiên khi hệ thống ngày càng mở rộng, việc điều phối các state và truyền props xuống các component con trở thành một nhiệm vụ phức tạp. Điều này dẫn đến việc cần có các công cụ để quản lý trạng thái chung của hệ thống. Hiện nay, có nhiều công cụ giúp quản lý trạng thái của hệ thống như: MobX [12], apolloGraphQL [13], ContextAPI, Pullstate [14], and RxJs [15]. Tuy nhiên với ứng dụng React, Redux là một thư viện được sử dụng phổ biến.

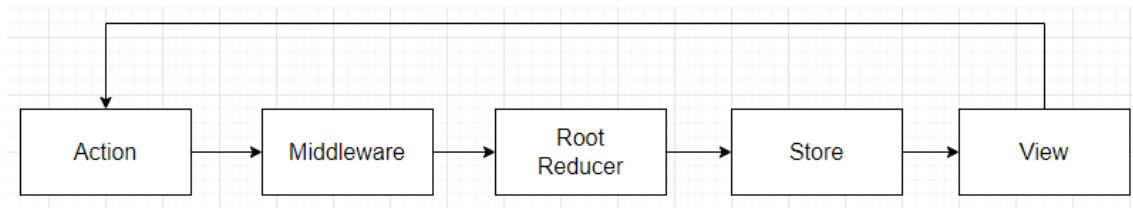
Cách Redux hoạt động rất đơn giản [16]. Có một Store trung tâm chứa toàn bộ trạng thái của ứng dụng. Mỗi thành phần có thể truy cập trạng thái được lưu trữ mà không phải gửi từ thành phần này sang thành phần khác. Có ba phần xây dựng: actions, store, and reducers. Actions nói một cách đơn giản là các sự kiện. Chúng là cách duy nhất bạn có thể gửi dữ liệu từ ứng dụng của mình đến "store" Redux. Dữ liệu có thể là từ các tương tác của người dùng, các lệnh gọi API hoặc là gửi form. Reducers là các hàm thuần túy lấy trạng thái hiện tại của ứng dụng, thực hiện một hành động và trả về trạng thái mới. Các trạng thái này được lưu trữ dưới dạng đối tượng và chúng xác định trạng thái của ứng dụng thay đổi như thế nào để đáp ứng với hành động được gửi đến "store". Trong một ứng dụng lớn, reducers có thể chia tách nhỏ hơn thành các slice, mỗi slice đảm nhận kiểm soát một phần trạng thái toàn cục của hệ thống. Các slice này sau đó sẽ được tổng hợp trong một nơi được gọi là RootReducer. RootReducer truyền tham số cho Store là nơi giữ trạng thái ứng dụng. Chỉ có một Store trong bất kỳ ứng dụng Redux nào.



**Hình 3.2:** Redux design pattern

Redux khi thực hiện asynchronous action (hành động bất đồng bộ) như lời gọi API, sẽ xảy ra hai hành động đối với Store. Hành động thứ 1 xảy ra ngay khi action được gọi, hành động thứ 2 xảy ra khi lời gọi bất đồng bộ hoàn thành. Để thực hiện các hành động không đồng bộ, Redux giới thiệu các middleware giúp: ghi nhật ký

hành động, gửi hành động mới, báo cáo lỗi và kích hoạt các yêu cầu không đồng bộ tiếp theo.



**Hình 3.3:** Luồng Redux với middleware

Hệ thống sử dụng Redux Toolkit là một thư viện tiêu chuẩn của Redux. Redux Toolkit giúp làm giảm thời gian công sức khi viết các đoạn mã theo cấu trúc của Redux. Redux-Thunk là middleware mặc định của Redux Toolkit.

### 3.4 Công nghệ React Query

ReactJS không cung cấp một cách tìm nạp dữ liệu từ các component. Các thư viện quản lý trạng thái như Redux, RxJS, MobX chỉ đáp ứng quản lý trạng thái ở máy khách. Tuy nhiên, chúng không quá tuyệt vời khi làm việc với các trạng thái không đồng bộ hoặc trạng thái máy chủ.

React Query [17] ra đời giúp giải quyết hiệu quả vấn đề đó. React Query giúp dễ dàng tìm nạp, lưu vào bộ nhớ đệm, đồng bộ hóa và cập nhật trạng thái máy chủ. React Query giúp chúng ta tạo ra các hook giúp chúng ta tránh lặp code, dễ bảo trì, tăng trải nghiệm người sử dụng và tiết kiệm băng thông khi giao tiếp với máy chủ.

	<b>React Query</b>	<b>SWR</b>	<b>Apollo Client</b>	<b>RTK-Query</b>	<b>React Router</b>
Github Repo / Stars	28k	23k	18k	8.1k	47k
Platform Requirements	React	React	React, GraphQL	Redux	React
Caching Strategy	Hierarchical Key ->Value	Unique Key ->Value	Normalized Schema	Unique Key ->Value	Nested Route ->value
Queries	Yes	Yes	Yes	Yes	Yes
Parallel Queries	Yes	Yes	Yes	Yes	Yes

Polling /Intervals	Yes	Yes	Yes	Yes	No
Paginated Queries	Yes	Yes	Yes	Yes	Yes
Infinite Queries	Yes	Yes	Yes	No	No

**Bảng 3.1:** So sánh các thư viện tìm nạp dữ liệu (chi tiết [17])

### 3.5 Công nghệ lập trình máy chủ NodeJS

NodeJS [5] là một nền tảng được xây dựng trên V8 JavaScript Engine [6] – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng. NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux. Hiện nay, ngôn ngữ lập trình phía máy chủ có thể sử dụng các ngôn ngữ C++, C#, Python nhưng NodeJS đem lại những ưu điểm. NodeJS được viết bằng JavaScript, vì thế, lập trình viên sẽ tốn ít công sức hơn khi làm quen với NodeJS nếu nền tảng lập trình phía client sẵn có. NodeJS có một cộng đồng lớn, chính vì vậy, nếu bạn cần hỗ trợ gì về NodeJS, sẽ nhanh chóng có người hỗ trợ bạn. Không chỉ vậy, NodeJS còn được đánh giá có tốc độ xử lý nhanh, nhờ cơ chế xử lý bất đồng bộ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào, dễ dàng mở rộng, phát huy hiệu quả với các ứng dụng thời gian thực như các ứng dụng nhắn tin, các mạng xã hội.

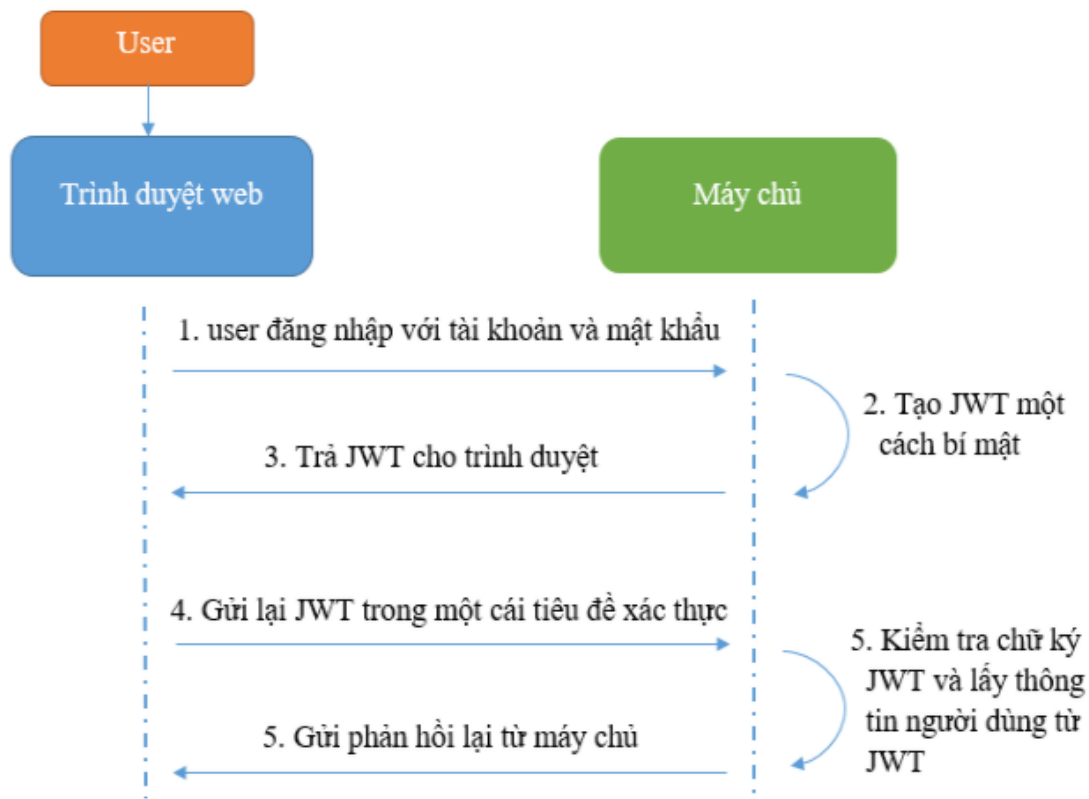
Tuy nhiên, bên cạnh đó, Nodejs gây hao tốn tài nguyên và thời gian. Nodejs được viết bằng C++ và JavaScript nên khi xử lý cần phải trải qua một quá trình biên dịch. Nếu bạn cần xử lý những ứng dụng tốn tài nguyên CPU thì không nên sử dụng Nodejs. NodeJS phù hợp cho việc phát triển các ứng dụng mới, tuy nhiên đối với các dự án lớn thì không phải là sự lựa chọn tối ưu.

Hệ thống sử dụng kết hợp NodeJS với ExpressJS là một framework được xây dựng trên nền tảng của NodeJS. ExpressJS [18] cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. ExpressJS hỗ trợ các method HTTP [19] và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

### 3.6 Công nghệ JWT

Hệ thống sử dụng JWT (JSON Web Token) giúp phân quyền người dùng, nhằm để phân biệt yêu cầu gửi từ client đến server, xác thực quyền và bảo mật các tài

nguyên của hệ thống. JWT là một chuẩn mở (RFC 7519 [20]) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWT có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC [21]) hoặc một public / private key sử dụng mã hoá RSA [22]. JWT là sự kết hợp (bởi dấu .) một Object Header dưới định dạng JSON được encode base64, một payload object dưới định dạng JSON được encode base64 và một Signature cho URI cũng được mã hóa base64.



**Hình 3.4:** Cách thức hoạt động JWT

Theo hình 3.4, JWT được lưu trữ trên máy khách, chính vì vậy, server không gặp phải bất cứ vấn đề gì khi mở rộng hệ thống và không cần lưu trữ bất kỳ thông tin gì trên máy chủ, mọi thứ hoạt động chỉ phụ thuộc vào chuỗi token và khóa bí mật (private key) mà server sở hữu. Để đảm bảo tính bảo mật, hệ thống lên thường xuyên thay đổi khóa bí mật của hệ thống.

### 3.7 Công nghệ lập trình thời gian thực SocketIO

Trong các hệ thống hiện đại ngày nay, việc giao tiếp từ máy chủ đến máy khách là điều bắt buộc, trong trường hợp bạn cần máy khách(client) hay máy chủ(server) có thể nhận sự thay đổi bên kia thì bạn có thể giải quyết bằng cách sử dụng: AJAX,



long-polling, short-polling, HTML 5 server-sent events...[23] Tuy nhiên, sử dụng các cách này có độ trễ lớn và gây tốn tài nguyên. Bởi vậy, SocketIO [24] ra đời cho phép xây dựng để xử lý việc giao tiếp giữa server và client ngay lập tức và chiếm ít tài nguyên nhất.

Trong SocketIO, các kết nối sẽ được thiết lập với WebSocket, cung cấp một kênh giao tiếp chi phí thấp giữa máy chủ và máy khách. Trong trường hợp kết nối WebSocket không thể thực hiện được, nó sẽ trở lại trạng thái bỏ phiếu dài HTTP (HTTP long-polling). Nếu kết nối bị mất, máy khách sẽ tự động cố gắng kết nối lại. SocketIO mở rộng quy mô đến nhiều máy chủ và gửi sự kiện đến tất cả các máy khách được kết nối một cách dễ dàng.

Trong đồ án, hệ thống có sử dụng SocketIO để realtime cho các chức năng: tương tác với bài viết, trao đổi giữa người dùng, nhận thông báo giữa các người dùng... Việc sử dụng SocketIO đem lại nhiều hiệu quả, giúp người sử dụng dễ dàng nắm bắt các thông tin nhanh chóng, dễ dàng và thuận tiện nhất.

### **3.8 Cơ sở dữ liệu MongoDB**

Hệ thống sử dụng MongoDB - một hệ quản trị cơ sở dữ liệu thích hợp cho các ứng dụng mạng xã hội, bán hàng [25]. Vì vậy, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON (Binary JSON). MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

Trong MongoDB có hai khái niệm cơ bản là: Collection và Document [26] [27]. Một document trong MongoDB có thể hiểu như một bản ghi record trong cơ sở dữ liệu quan hệ (RDBMS). Tuy nhiên có sự khác biệt giữa hai loại khái niệm này là một document không nhất thiết phải có cấu trúc cố định, còn record cần cấu trúc dữ liệu phải giống hệt nhau giữa các bản ghi. Collection trong MongoDB tương đương như một bảng trong kiểu cơ sở dữ liệu quan hệ. Một Collection gồm nhiều documents ở trong nó và mỗi Collection phải có kiểu cấu trúc riêng của nó.

Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu. Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi chèn, xóa hay cập nhật nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS tạo ra

hiệu suất truy vấn cao hơn so với RDBMS[27].

Trong MongoDB có hai khái niệm cần nắm rõ: Normalization và Denormalization [28]. Normalization là kỹ thuật chia dữ liệu thành nhiều bảng để giảm bớt sự dư thừa và không nhất quán của dữ liệu và để đạt được tính toàn vẹn của dữ liệu. Mặt khác, Denormalization là kỹ thuật kết hợp dữ liệu vào một bảng duy nhất để làm cho việc truy xuất dữ liệu nhanh hơn. Normalization hỗ trợ tốt cho việc chèn, xóa, sửa dữ liệu, Denormalization lại hỗ trợ tốt cho việc đọc, truy vấn dữ liệu. MongoDB giúp bạn có thể thực hiện cả 2 điều này. Chính vì vậy, phân tích nghiệp vụ hệ thống và sử dụng phù hợp một trong hai hình thức tổ chức dữ liệu này (Normalization hay Denormalization) sẽ giúp cho hệ thống có hiệu suất cao, tốc độ thực thi nhanh.

Tuy nhiên, MongoDB cũng có mặt hạn chế. Đó là cần RAM lớn do dữ liệu được caching. Dữ liệu trong MongoDB không bị ràng buộc như RDBMS nhưng người sử dụng lưu ý cẩn thận mọi thao tác để không xảy ra các kết quả ngoài ý muốn gây ảnh hưởng đến dữ liệu.

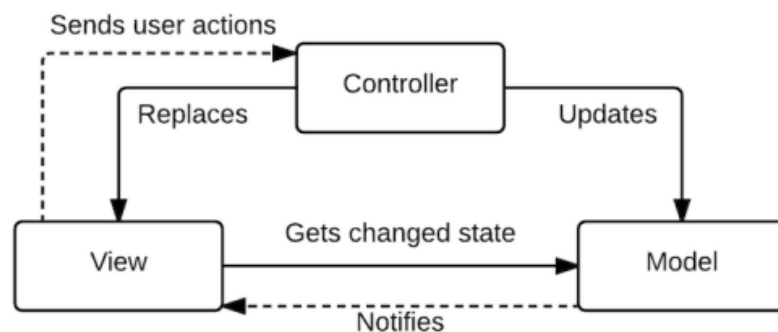
## CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm

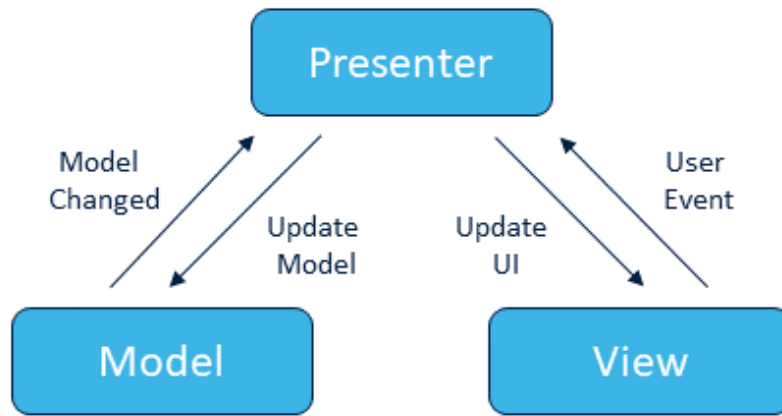
Hệ thống được triển khai trên nền tảng ứng dụng website, dựa trên kiến trúc client - server theo mẫu thiết kế 3 lớp MVP. Để hiểu MVP là gì, trước hết, ta cần tìm hiểu mẫu thiết kế MVC là gì? Vì MVP được phát triển dựa trên MVC.

MVC từ viết tắt bởi 3 từ Model – View – Controller [9]. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình chia mã nguồn thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình. Model (M): là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu... View (V) là giao diện hiển thị dành cho người dùng. Đây là nơi mà người sử dụng sẽ thực hiện các thao tác truy vấn, tìm kiếm... dữ liệu. Bộ phận này đại diện cho các thành phần UI (ví dụ như: XML, HTML...). Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Controller (C): bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Controller sẽ xử lý các dữ liệu của người dùng qua Model và trả về kết quả ở View.



**Hình 4.1:** MVC Design Pattern

Mẫu thiết kế MVP là một mẫu thiết kế mạnh mẽ và dễ hiểu cung cấp cho một loạt các nhiệm vụ phát triển ứng dụng và thành phần. Hơn nữa, MVP cũng có thể thích ứng trên nhiều client/server và nhiều kiến trúc ứng dụng đa tầng [29]. Hình 4.2 minh họa mẫu thiết kế MVP:

**Hình 4.2:** MVP Design Pattern

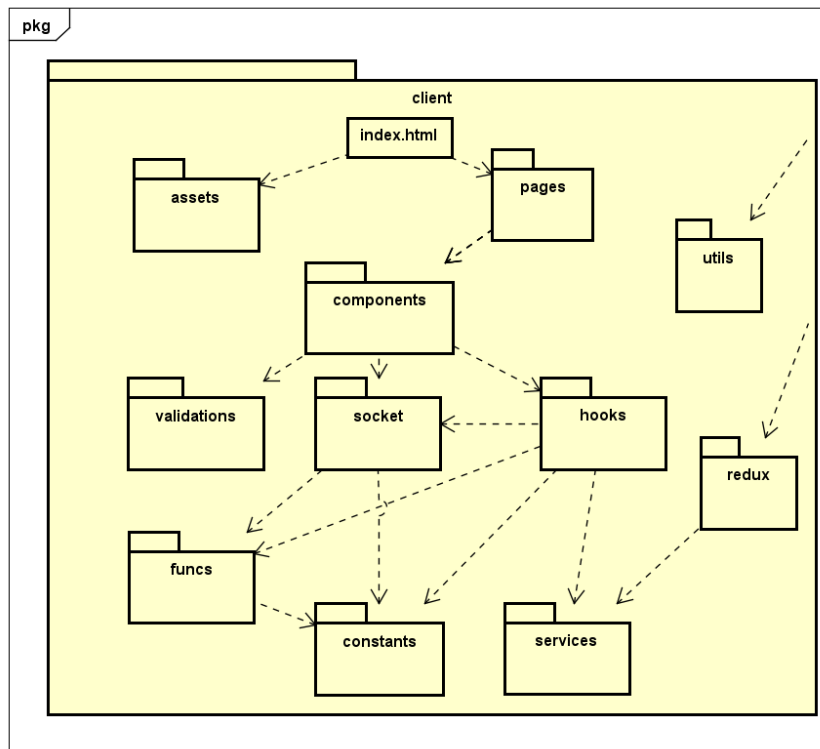
Mẫu thiết kế MVP được xây dựng dựa trên nền tảng mẫu thiết kế MVC. Trong đó, Controller (C) được thay thế bởi Presenter (P) trong mẫu kiến trúc MVP. Presenter chịu trách nhiệm cho việc tiếp nhận tương tác từ View (V) giao tiếp với Model (M) xử lý dữ liệu và trả về kết quả.

So sánh hai mô hình thiết kế MVP và MVC, ta thấy, MVP có nhiều ưu điểm: View không sử dụng trực tiếp Model trong MVP mà giao tiếp giữa chúng được thực hiện bởi Presenter (Controller trong MVC). Trong MVC, View sẽ đọc dữ liệu trực tiếp từ Model thay vì thông qua Controller. Khi View cần được sửa đổi, Model không bị ảnh hưởng bởi mẫu MVP, nhưng trong mẫu MVC, Model sẽ bị ảnh hưởng bởi các thay đổi của View. Trong MVP, Presenter và View là hoàn toàn độc lập với nhau. Từ hai điểm khác nhau của MVC và MVP, có thể kết luận rằng MVP tốt hơn đáng kể so với MVC trong việc tách các thành phần và tái sử dụng [30].

Hệ thống sử dụng bộ công nghệ (web stack) MERN (MongoDB, Express, ReactJS, Node.js) [2] được triển khai theo mẫu thiết kế MVP được lập trình trên ứng dụng web. Trong đó, phần V (View) sử dụng ReactJS cùng với một số thư viện hỗ trợ chia ứng dụng thành pages (các trang) được tách riêng đảm bảo tính loose coupling. Một page (trang) được chia thành các component (thành phần) để dễ tái sử dụng. Phần P (Presenter) bao gồm các gói routes, controllers, sockets, services thực hiện việc tiếp nhận dữ liệu và phản hồi dữ liệu cho người dùng. Phần M (Model) nằm trong gói models cung cấp các phương thức thao tác với dữ liệu được gọi bởi Presenter. Hơn nữa, MVC được áp dụng trong phần View của hệ thống giúp cho việc phát triển và bảo trì mã nguồn dễ dàng và thuận tiện hơn (sử dụng ReactJS 3.2 cùng với các thư viện hỗ trợ Redux và React Query).

### 4.1.2 Thiết kế tổng quan

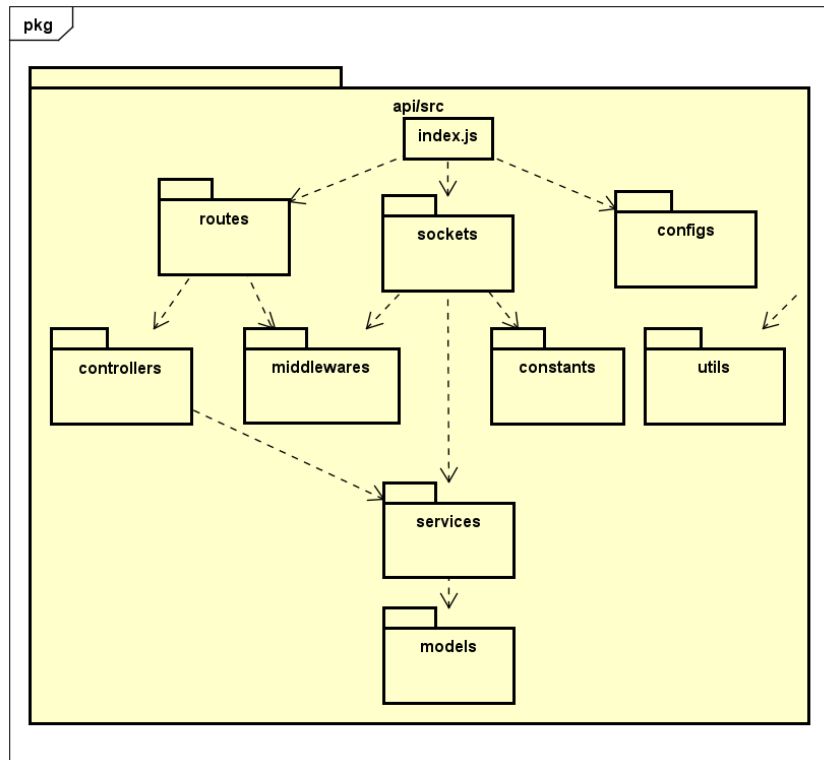
#### a, Biểu đồ phụ thuộc gói client



**Hình 4.3:** Biểu đồ phụ thuộc gói Client

STT	Gói	Mô tả
1	assets	Chứa file hình ảnh của hệ thống
2	pages	Chứa các trang của hệ thống
3	components	Chứa các thành phần của hệ thống có thể tái sử dụng
4	validations	Chứa các file xác thực form của hệ thống
5	socket	Chứa các file đảm nhận tạo, nhận, sự kiện từ SocketIO của hệ thống
6	hooks	Chứa file lưu trữ các state của components giúp tái sử dụng, dễ bảo trì hệ thống
7	funcs	Xử lý các dữ liệu từ SocketIO nhận được
8	constants	Chứa các hằng số của SocketIO và hệ thống
9	services	Có nhiệm vụ gọi các truyền nhận dữ liệu với server theo giao thức RESTful API
10	redux	Chứa các state chung của hệ thống
11	utils	Chứa các tiện ích

**Bảng 4.1:** Đặc tả tổng quan gói Client

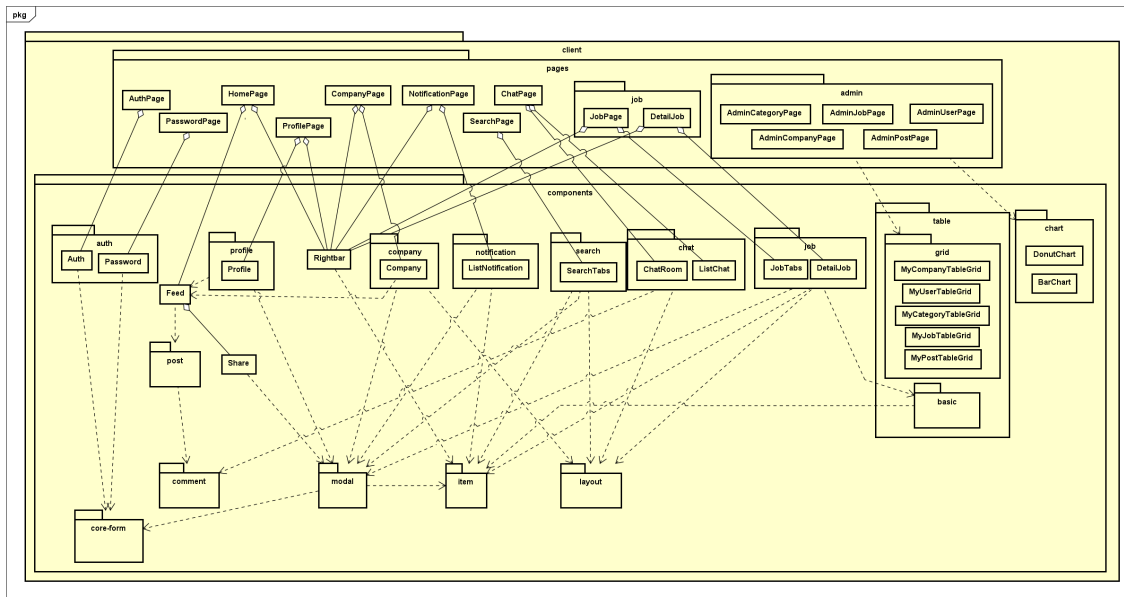
**b, Biểu đồ phụ thuộc gói server****Hình 4.4:** Biểu đồ phụ thuộc gói Server

STT	Gói	Mô tả
1	routes	Xử lý phân định các yêu cầu lời gọi RESTful API đầu vào
2	sockets	Tạo, nhận các sự kiện của hệ thống
3	configs	Cấu hình của hệ thống
4	middlewares	Xác thực tính hợp lệ của lời gọi RESTful API, SocketIO, xác định danh tính người dùng
5	constants	Chứa các hằng số của SocketIO và hệ thống
6	controllers	Định dạng dữ liệu trả về, xử lý lỗi
7	services	Thực hiện các nghiệp vụ cho hệ thống
8	models	Bao gồm các thành phần cơ bản của hệ thống, với thuộc tính, các phương thức đại diện cho các bảng trong cơ sở dữ liệu.
9	utils	Chứa các tiện ích của hệ thống

**Bảng 4.2:** Đặc tả tổng quan gói Server

### 4.1.3 Thiết kế chi tiết gói

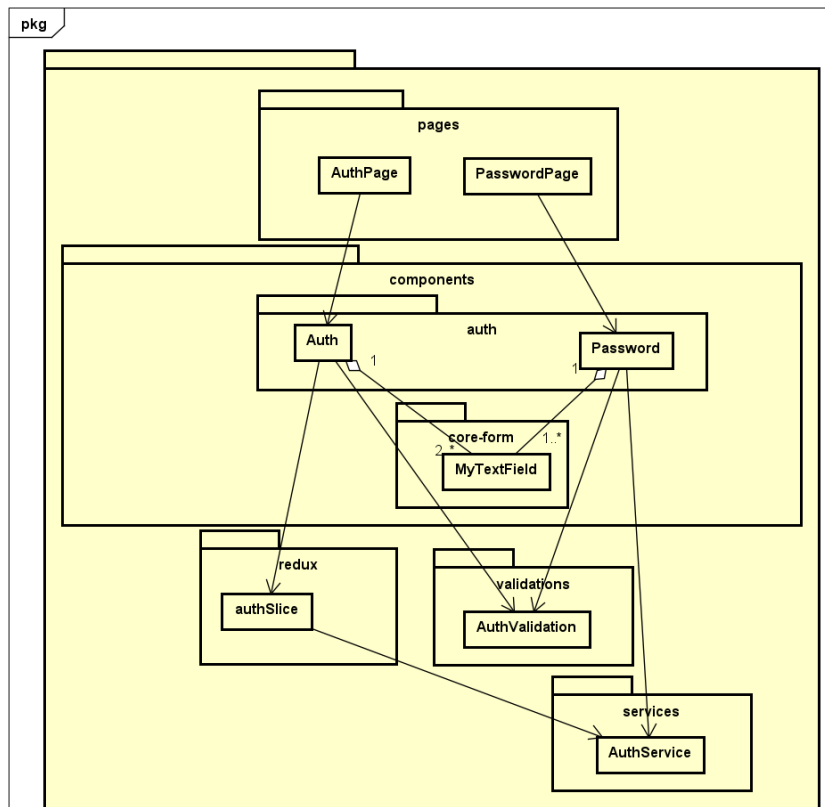
**a, Thiết kế chi tiết gói thành phần giao diện**



### Hình 4.5: Thiết kế gói thành phần giao diện

STT	Gói/Lớp	Mô tả
1	auth	Gói chứa thành phần cho tác nhân khách: đăng ký, đăng nhập, quên mật khẩu
2	Feed	Hiển thị danh sách bài đăng
3	profile	Gói chứa các thành phần chính của hồ sơ: thông tin cá nhân, các bài đăng của họ
4	company	Gói chứa các thành phần chính của công ty: hồ sơ công ty, danh sách việc làm. . .
5	Rightbar	Phần được hiện thị bên phải của hệ thống: đề xuất người dùng, công việc, công ty.
6	search	Gói để hiển thị giao diện tìm kiếm: người dùng, công ty, công việc.
7	table	Gói chứa 2 gói con là: grid (hiển thị bảng cho quản trị viên), basic (hiển thị bảng cho người dùng).
8	chart	Gói hiển thị biểu đồ: cột, donut.
9	modal	Gói chứa danh sách các modal được sử dụng trong hệ thống (tạo bài đăng mới, tạo tin tuyển dụng mới, danh sách người dùng. . .).

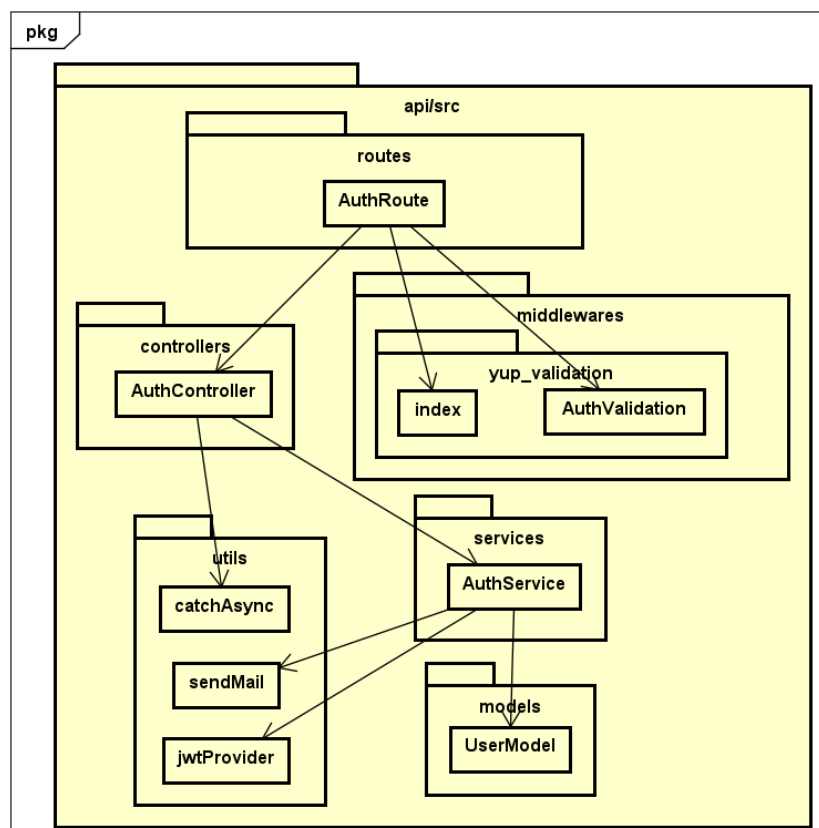
10	item	Gói chứa thành phần con được sử dụng ở nhiều nơi trong hệ thống: phần tử người dùng, phần tử công ty, phần tử tin tuyển dụng.
----	------	---

**Bảng 4.3:** Thành phần chính của gói giao diện**b, Thiết kế chi tiết gói auth****Hình 4.6:** Thiết kế chi tiết gói auth client

STT	Lớp	Mô tả
1	AuthPage	Hiển thị trang đăng nhập hoặc đăng ký
2	PasswordPage	Hiển thị trang quên mật khẩu hoặc trang thay đổi mật khẩu
3	Auth	Tạo ra mẫu đơn đăng nhập hoặc đăng ký sử dụng react-hook-form
4	Password	Tạo ra mẫu đơn quên mật khẩu hoặc thay đổi mật khẩu
5	MyTextField	Là trường input để điền vào form
6	authSlice	Chứa các hành động đăng nhập và đăng ký



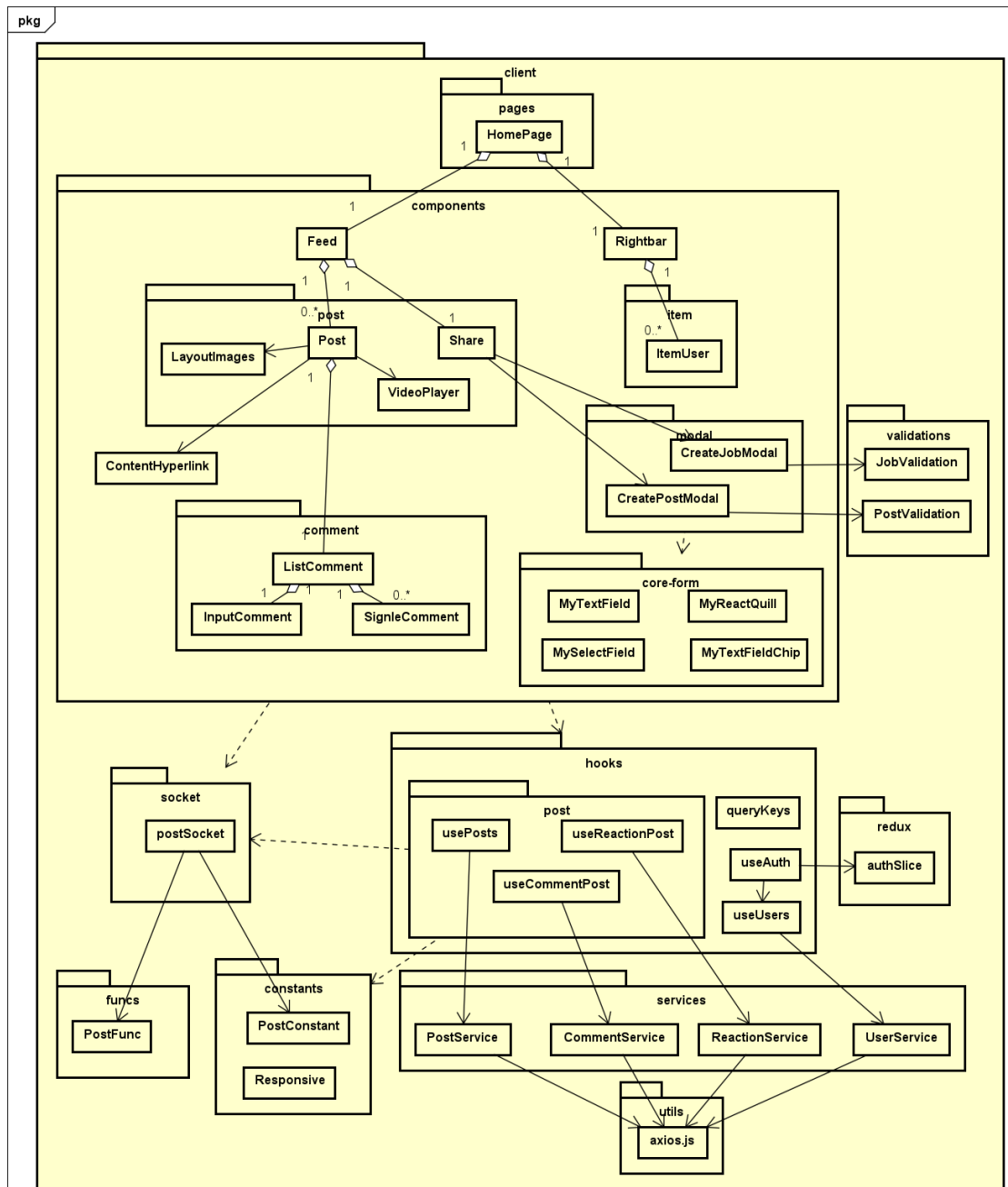
7	AuthValidation	Chứa các schema giúp xác thực biểu mẫu của form
8	AuthService	Giao tiếp nhận dữ liệu từ server

**Bảng 4.4:** Đặc tả gói auth client**Hình 4.7:** Thiết kế chi tiết gói auth server

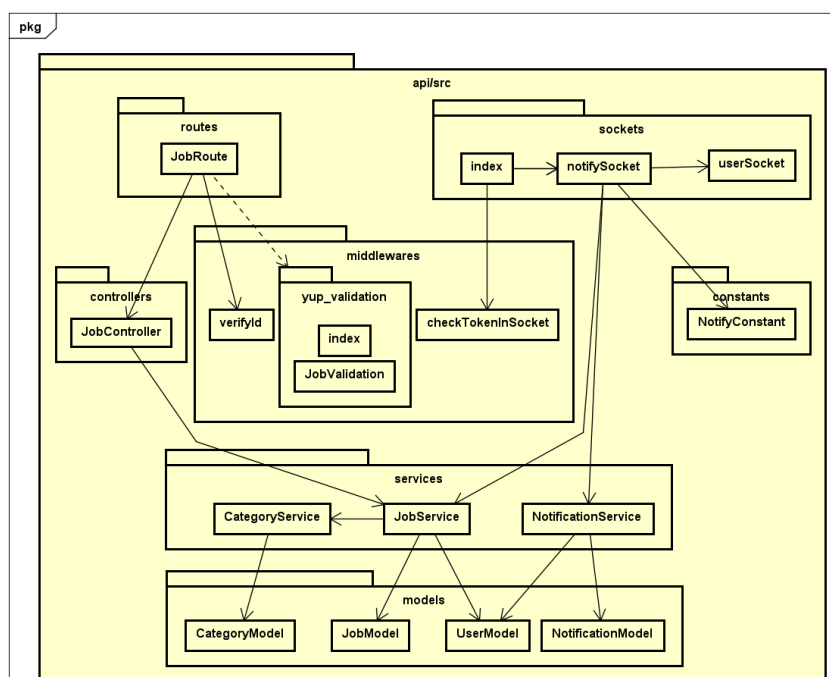
STT	Lớp	Mô tả
1	AuthRoute	Điều hướng các RESTful API từ client
2	index	Xác thực dữ liệu tới từ client: query, params, body
3	AuthValidation	Chứa các schema để xác thực dữ liệu từ client
4	AuthController	Gọi các nghiệp vụ và định dạng đầu ra của dữ liệu
5	catchAsync	Bắt lỗi ngoại lệ
6	AuthService	Thực hiện nghiệp vụ của hệ thống: đăng ký, đăng nhập, đổi mật khẩu...
7	sendMail	Dịch vụ gửi thư điện tử
8	jwtProvider	Ký mã token để xác thực người dùng
9	UserModel	Lưu trữ cấu trúc trên cơ sở dữ liệu

**Bảng 4.5:** Đặc tả gói auth server

## c, Thiết kế chi tiết giao diện trang chủ



Hình 4.8: Thiết kế màn hình trang chủ

**d, Thiết kế gói cho nghiệp vụ việc làm máy chủ****Hình 4.9:** Thiết kế gói nghiệp vụ việc làm

STT	Lớp	Mô tả
1	JobRoute	Điều hướng RESTful API request đến từ client, request được thông qua các một loạt các middleware để kiểm tra tính hợp lệ
2	index (gói socket)	Khởi tạo không gian tên, middleware cho socket
3	notifySocket	Truyền, nhận các sự kiện thông báo: followUser, followCompany, applyJob...
4	userSocket	Chứa danh sách người dùng hiện thời trong hệ thống với không gian tên: namespace("/")
5	verifyId	Xác thực danh tính người dùng dựa vào mã JWT
6	index (gói yup_validation)	Xác thực dữ liệu được truyền từ client theo giao thức RESTful API dựa vào các schema
7	JobValidation	Chứa các schema xác thực
8	checkTokenInSocket	Kiểm tra tính hợp lệ mã JWT trong socket
9	JobController	Định dạng trạng thái, dữ liệu trả về cho client
10	NotifyConstant	Chứa hằng số cho các sự kiện thông báo
11	CategoryService	Chứa các nghiệp vụ liên quan đến kỹ năng

12	JobService	Chứa nghiệp vụ liên quan đến tin tuyển dụng (công việc)
13	Notification Service	Chứa nghiệp vụ liên quan đến thông báo
14	CategoryModel	Định nghĩa schema, chứa các phương thức thao tác với bảng Category
15	UserModel	Định nghĩa schema, chứa các phương thức thao tác với bảng User
16	JobModel	Định nghĩa schema, chứa các phương thức thao tác với bảng Job
17	Notification Model	Định nghĩa schema, chứa các phương thức thao tác với bảng Notification

**Bảng 4.6:** Đặc tả gói nghiệp vụ việc làm

## 4.2 Thiết kế chi tiết

### 4.2.1 Thiết kế giao diện

Thiết kế giao diện của người dùng phải đảm bảo tuân thủ các nguyên tắc thiết kế, đáp ứng đầy đủ yêu cầu người dùng. Hệ thống phải đảm bảo đơn giản hóa các tác vụ cho người sử dụng, trình bày bố cục thuận tiện rõ ràng nhất cho người sử dụng dễ dàng hiểu và thao tác

Giao diện hệ thống hỗ trợ tính responsive với từng kích thước pixel của màn hình thiết bị. Cụ thể, hệ thống sẽ quy định các breakpoint để hiển thị giao diện tương ứng, có 4 loại breakpoint được hệ thống hỗ trợ: xs, extra-small: 0px; sm, small: 600px; md, medium: 900px; lg, large: 1200px; Về màu sắc, hệ thống có hai chế độ: sáng và tối, các màu sắc của hệ thống được dựa trên bảng màu của thư viện MUI – một thư viện hỗ trợ xây dựng giao diện nhanh chóng, dễ dàng tùy chỉnh.

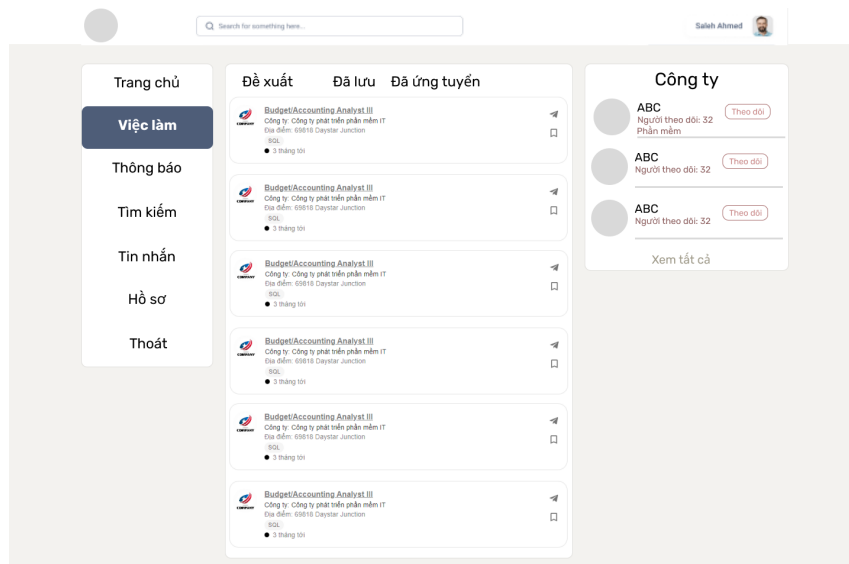
**Hình 4.10:** Bảng màu với giao diện sáng



Hình 4.11: Bảng màu với giao diện tối

### a, Giao diện trang việc làm (tin tuyển dụng)

Giao diện cung cấp việc làm ở ba loại: danh sách việc làm được đề xuất cho người tìm việc – dựa trên hồ sơ của người tìm việc, danh sách việc làm mà người tìm việc quan tâm được đánh dấu là đã lưu và danh sách việc làm mà người dùng đã ứng tuyển. Mỗi việc làm bao gồm các thông tin cơ bản: tên nhan đề việc làm, tên công ty sở hữu việc làm, địa điểm làm việc, danh sách kỹ năng cần có của việc làm, thông tin về hạn chót nộp đơn ứng tuyển. Bên cạnh đó, giao diện của trang cũng cung cấp danh sách các công ty mà người dùng có thể quan tâm dựa trên hồ sơ của người dùng khai báo.



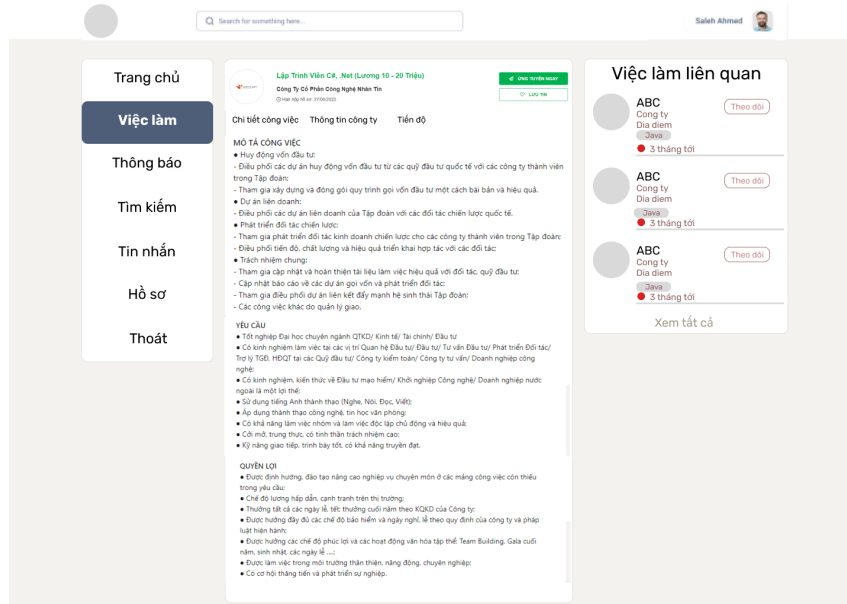
Hình 4.12: Thiết kế giao diện trang việc làm

### b, Giao diện trang chi tiết công việc

Trang chi tiết công việc cung cấp các thông tin chi tiết của một việc làm, bao gồm: mức lương, số lượng tuyển, yêu cầu đối với ứng viên... Ứng viên có thể lưu

tin hoặc có thể ứng tuyển thông qua giao diện nút bấm rõ ràng, trực quan. Đối với nhà tuyển dụng sở hữu việc làm hệ thống cho phép nhà tuyển dụng xem tiến độ của công việc bao gồm: số lượng người theo dõi, số lượng ứng viên đã nộp đơn ứng tuyển.

Ngoài ra, giao diện trang ở phía bên phải của màn hình cung cấp các việc làm có liên quan dựa trên danh sách kỹ năng của việc làm hiện tại đang được xem.



Hình 4.13: Thiết kế trang chi tiết công việc

#### 4.2.2 Thiết kế lớp

##### a, Lớp JobService chứa các nghiệp vụ liên quan đến tin tuyển dụng của hệ thống

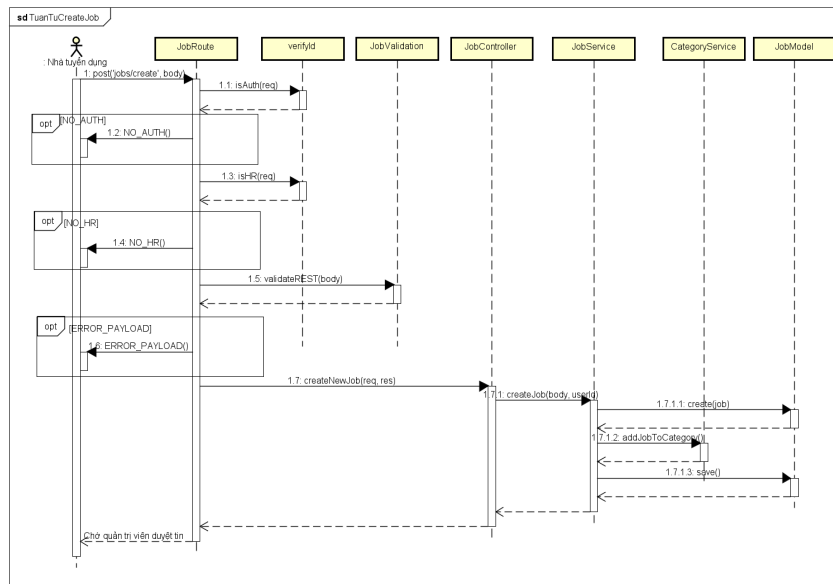
STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1.	Category	Category	Một đối tượng của Category-Model chứa phương thức thao tác với bảng Category
2.	Company	Company	Chứa các phương thức thao tác với bảng Company
3.	Job	Job	Chứa các phương thức thao tác với bảng Job
4.	User	User	Chứa các phương thức thao tác với bảng User
5.	convertToSlug	convertToSlug	Chuyển chuỗi thành chuỗi slug
6.	CategoryService	CategoryService	Nghệp vụ của category

7.	RoomService	RoomService	Nghệp vụ của room
----	-------------	-------------	-------------------

**Bảng 4.7:** Mô tả thuộc tính của lớp JobService

STT	Phương thức	Kiểu trả về	Mô tả
1.	createJob	Object	Tạo mới công việc
2.	updateActiveJobs	Object	Cập nhật trạng thái của công việc theo companyId
3.	updateApplicantJob	Object	Người tìm kiếm việc làm ứng tuyển công việc
4.	deleteApplicantJob	Object	Chủ sở hữu tin tuyển dụng xóa ứng viên
5.	updateJob	Object	Nhà tuyển dụng cập nhật công việc hoặc quản trị viên thêm lý do từ chối công việc
6.	confirmJob	Object	Quản trị viên từ chối hoặc chấp nhận công việc
7.	getJobByCategory	Object	Tìm công việc theo tên kỹ năng
8.	listJobRelateCurrentJob	Object	Tìm các công việc liên quan theo mã công việc
9.	infoJob	Object	Hiển thị thông tin chi tiết công việc với từng loại tác nhân: người tìm việc, nhà tuyển dụng, quản trị viên, nhà quản lý công ty.
10.	recommendJobByUser	Object	Gợi ý các công việc dựa vào hồ sơ kỹ năng của người dùng
11.	viewApplyJob Service	Object	Xem đơn ứng tuyển
12.	listJobByCompany Service	Object	Danh sách công việc của công ty
13.	listMyJobHR	Object	Danh sách công việc của nhà tuyển dụng
14.	getJobStatsService	Object	Thống kê danh sách việc làm trong năm theo tháng
15.	getJobStatusService	Object	Thống kê trạng thái của công việc

16.	changeOwnerJob	Object	Nhà quản lý công ty thay đổi quyền sở hữu của tin tuyển dụng
-----	----------------	--------	--

**Bảng 4.8:** Mô tả phương thức của lớp JobService**Hình 4.14:** Biểu đồ tuần tự createJob

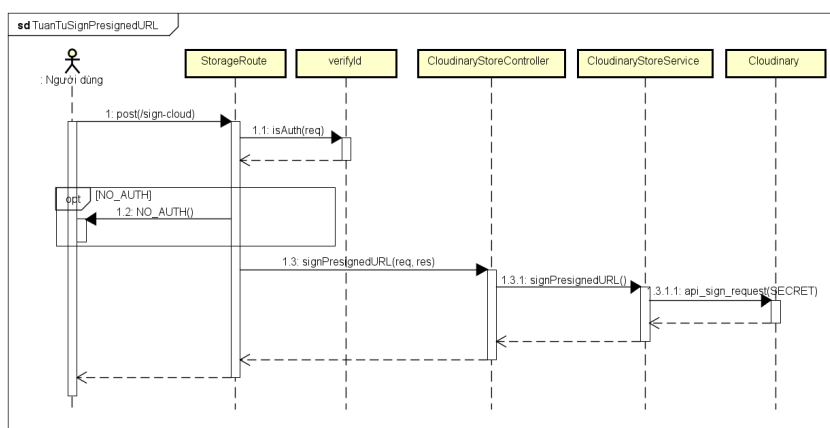
**b, Lớp PostService chứa các nghiệp vụ lên quan đến bài đăng của hệ thống**

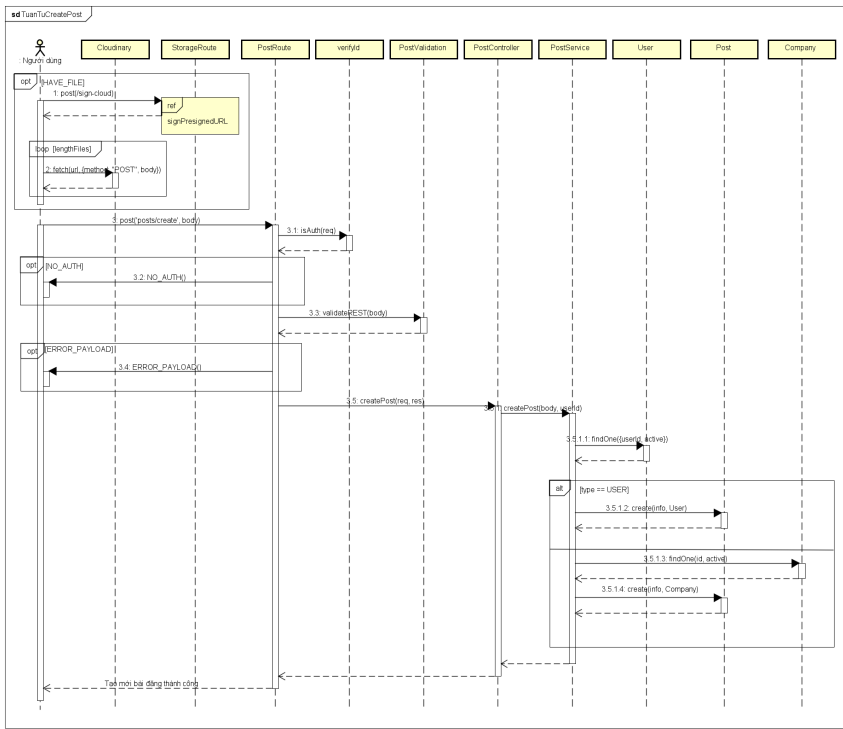
STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1.	Post	Post	Là một đối tượng của PostModel chứa phương thức giao tiếp với bảng Post
2.	Company	Company	Chứa các phương thức thao tác với bảng Company
3.	Reaction	Reaction	Chứa các phương thức thao tác với bảng Reaction
4.	Comment	Comment	Chứa các phương thức thao tác với bảng Comment
5.	User	User	Chứa các phương thức thao tác với bảng User

**Bảng 4.9:** Mô tả thuộc tính của lớp PostServer



STT	Phương thức	Kiểu trả về	Mô tả
1.	createPost	Object	Tạo mới bài đăng kiểu cá nhân và công ty
2.	findPost	Object	Tìm bài đăng theo mã
3.	listPostAdmin	Object	Danh sách bài đăng với quyền quản trị viên
4.	listPostByUser	Object	Danh sách bài đăng của người dùng lấy theo mã người dùng
5.	listPostByCompany	Object	Danh sách bài đăng của công ty lấy theo mã công ty
6.	listPostByFeed	Object	Danh sách bài đăng dựa trên lượt theo dõi người dùng và theo dõi công ty của người dùng
7.	getPostStatsService	Object	Thống kê danh sách bài đăng của năm theo tháng
8.	getPostTypeService	Object	Thống kê danh sách bài đăng dựa theo kiểu bài viết: cá nhân hoặc công ty.
9.	deletePostService	Object	Xóa bài đăng, tương tác, bình luận của bài đăng.

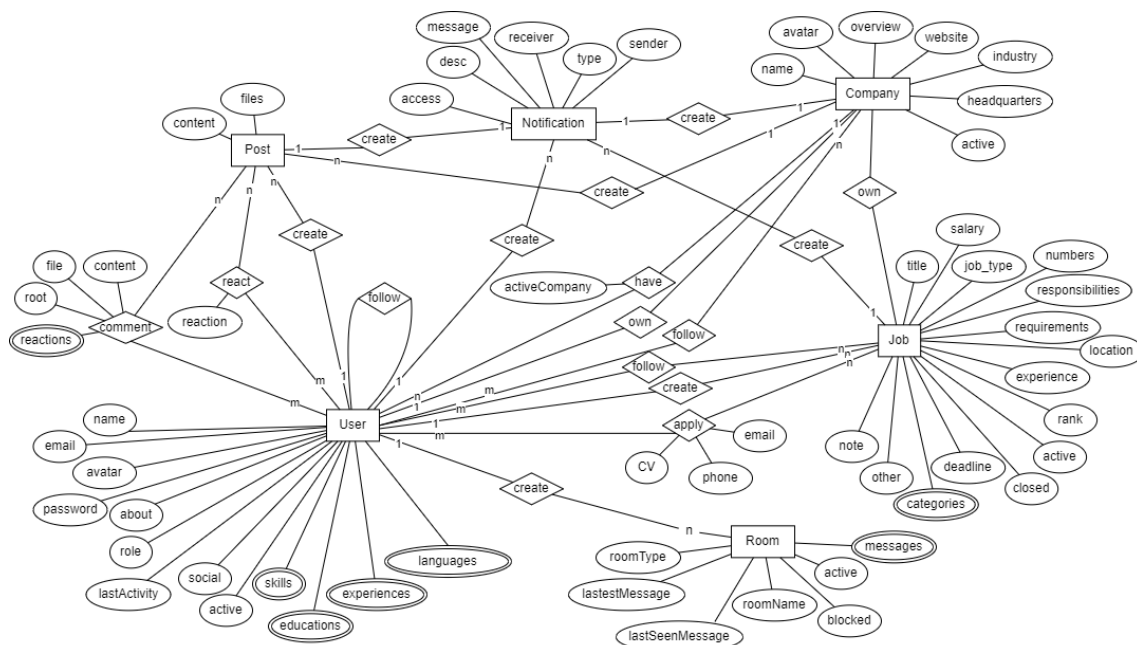
**Bảng 4.10:** Mô tả phương thức của lớp PostService**Hình 4.15:** Biểu đồ tuần tự signPresignedURL



### Hình 4.16: Biểu đồ tuần tự createPost

### 4.2.3 Thiết kế cơ sở dữ liệu

**a, Xây dựng biểu đồ thực thể liên kết**



### Hình 4.17: Biểu đồ thực thể liên kết

Thực thể User là người sử dụng hệ thống bao gồm các tác nhân: người tìm việc, nhà tuyển dụng, nhà quản trị công ty (manager công ty), quản trị viên.

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
name	String	2-30 ký tự	Tên đăng nhập
email	String	NOT_NULL, unique	Email đăng nhập
avatar	String		Ảnh đại diện người dùng
password	String	NOT_NULL	Mật khẩu
about	String	0-1000 ký tự	Tông quan về người dùng
role	String	Một trong 4 giá trị: admin, manager, HR, employee.	Vai trò của tài khoản
languages	Array		Trình độ ngoại ngữ
social	Object		Thông tin liên lạc qua các trang mạng xã hội.
active	Boolean		Trạng thái tài khoản
skills	Array		Danh sách kỹ năng của tài khoản
educations	Array		Danh sách học vấn của tài khoản
experiences	Array		Danh sách kinh nghiệm việc làm

**Bảng 4.11:** Thiết kế thực thể User

Thực thể Post: lưu trữ bài đăng của hệ thống

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
content	String	NOT_NULL	Nội dung bài đăng
files	Array		Danh sách tệp đính kèm theo bài đăng

**Bảng 4.12:** Thiết kế thực thể Post

Thực thể Company: lưu trữ công ty của hệ thống

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
name	String	NOT_NULL	Tên công ty
avatar	String		Hình đại diện của công ty

overview	String		Tổng quan về công ty
website	String		Website của công ty
industry	String		Lĩnh vực của công ty
headquarters	String		Trụ sở của công ty
active	String		Trạng thái của công ty

**Bảng 4.13:** Thiết kế thực thể Company

Thực thể Job: lưu trữ tin tuyển dụng công việc của công ty

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
title	String	NOT_NULL, 3-200 ký tự	Tên của công việc
salary	String	NOT_NULL	Lương của công việc
job_type	String	Nhận một trong ba giá trị: full-time, parttime, freelance	Hình thức làm việc
numbers	Number	NOT_NULL	Số lượng tuyển
responsibilities	String	Nhiều hơn 10 ký tự	Mô tả trách nhiệm của công việc
requirements	String	Nhiều hơn 10 ký tự	Yêu cầu trình độ người ứng tuyển đối với công việc
location	String	Nhiều hơn 3 ký tự	Địa điểm làm việc của công việc
rank	String	NOT_NULL	Vị trí công việc: Intern, Middle, Senior. . .
deadline	Date	NOT_NULL	Thời gian hết hạn ứng tuyển
note	String		Ghi chú khi công việc bị từ chối bởi quản trị viên
categories	Array		Danh sách các kỹ năng cần có với công việc hỗ trợ tìm kiếm.
other	String		Thông tin bổ sung: đãi ngộ, quyền lợi. . .
active	Boolean		Chấp thuận hay từ chối công việc của quản trị viên

closed	Boolean		Công việc bị khóa bởi nhà tuyển dụng
--------	---------	--	--------------------------------------

**Bảng 4.14:** Thiết kế thực thể Job

Thực thể Notification: lưu trữ danh sách thông báo của hệ thống

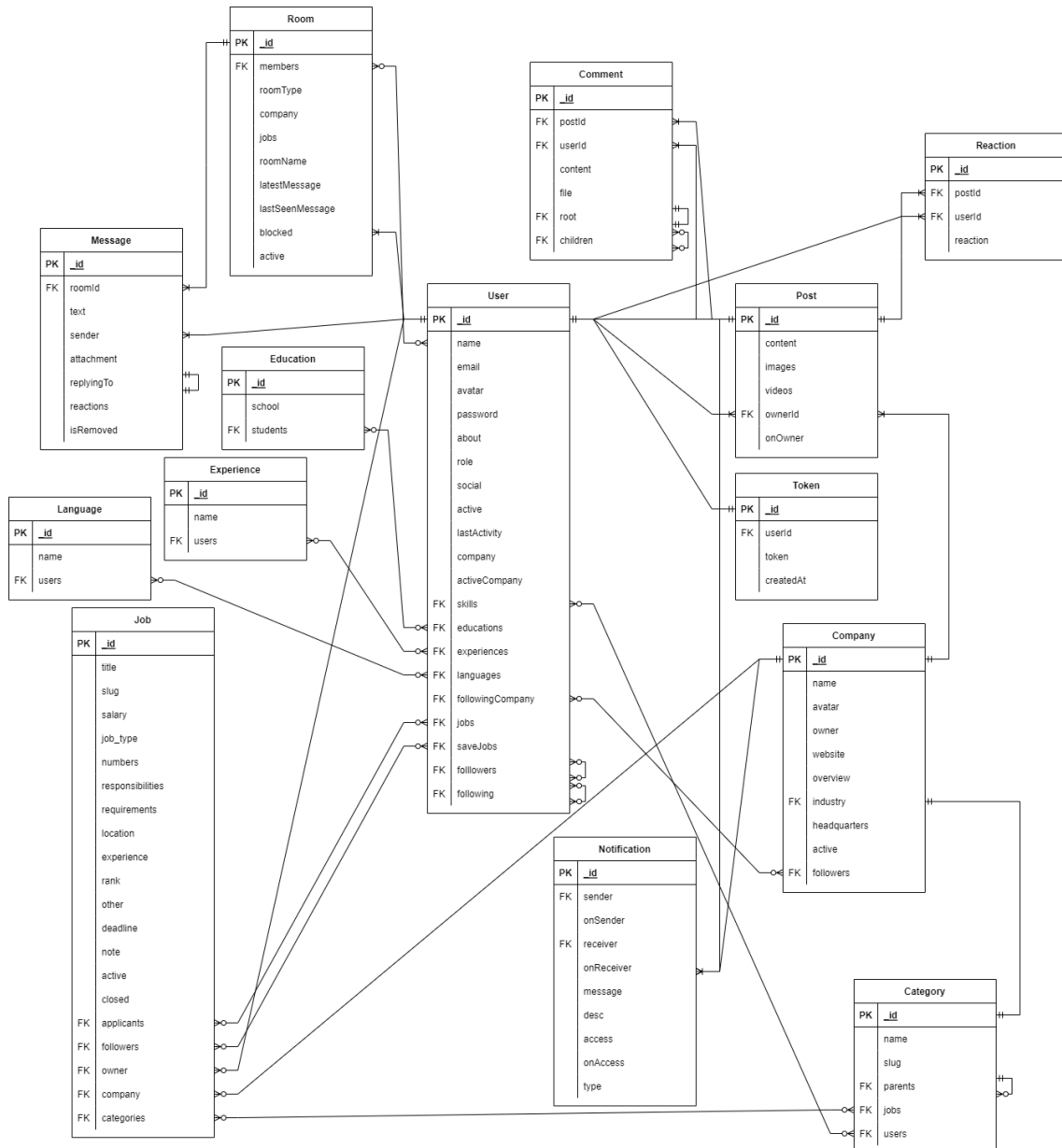
Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
sender	ObjectId		Mã người gửi
receiver	Array		Danh sách người nhận
message	String		Thông điệp
desc	String		Chi tiết thông điệp
access	ObjectId		Đối tượng truy cập của thông báo
type	String		Kiểu của thông báo

**Bảng 4.15:** Thiết kế thực thể Notification

Các thực thể có mối quan hệ chặt chẽ với nhau được minh họa ở bảng sau đây:

Thực thể nguồn	Thực thể đích	Tên liên kết	Ràng buộc	Mô tả
User	User	follow	Đệ quy	Người dùng theo dõi người dùng khác
User	Post	create	1-n	Người dùng tạo bài đăng
User	Post	reaction	n-m	Người dùng tương tác bài đăng
User	Post	comment	n-m	Người dùng bình luận bài đăng
User	Notification	create	1-n	Người dùng tạo thông báo: theo dõi người dùng, công ty...
User	Company	follow	n-m	Người dùng theo dõi công ty
User	Job	follow	n-m	Người dùng theo dõi công việc
User	Job	create	1-n	Nhà tuyển dụng tạo công việc
User	Job	apply	n-m	Người tìm việc ứng tuyển

User	Room	create	m-n	Tạo phòng trao đổi: cá nhân, công việc
Company	User	own	1-1	Chủ sở hữu công ty
Company	User	have	1-n	Thành viên của công ty
Company	Post	create	1-n	Tạo mới bài đăng công ty
Company	Notification	create	1-1	Thông báo tạo mới công ty
Company	Job	own	1-n	Công ty sở hữu công việc
Job	Notification	create	1-n	Thông báo của công việc

**Bảng 4.16:** Bảng thiết kế liên kết giữa các thực thể**b, Xây dựng biểu đồ cơ sở dữ liệu phi quan hệ****Hình 4.18:** Biểu đồ cơ sở dữ liệu phi quan hệ

### 4.3 Xây dựng ứng dụng

#### 4.3.1 Thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL	Phiên bản
IDE lập trình	Visual Studio Code	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>	1.68.1
Vẽ biểu đồ UML	Astah	<a href="https://astah.net/">https://astah.net/</a>	8.4.1
Vẽ biểu đồ cơ sở dữ liệu	Draw.IO	<a href="https://app.diagrams.net/">https://app.diagrams.net/</a>	
Môi trường lập trình Back-end	NodeJS	<a href="https://nodejs.org/en/">https://nodejs.org/en/</a>	14.17.0
Framework BackEnd	Express	<a href="https://expressjs.com/">https://expressjs.com/</a>	4.17.3
Thư viện xây dựng Frontend	ReactJS	<a href="https://reactjs.org/">https://reactjs.org/</a>	18.0.0
Driver mongo	Mongoose	<a href="https://mongoosejs.com/">https://mongoosejs.com/</a>	6.3.0
Hệ quản trị cơ sở dữ liệu	Mongo Compass	<a href="https://www.mongodb.com/">https://www.mongodb.com/</a>	1.32.2
Test API	Postman	<a href="https://www.postman.com/">https://www.postman.com/</a>	9.23.3
Test socket	Postman	<a href="https://www.postman.com/">https://www.postman.com/</a>	9.23.3
Tạo dữ liệu	Mockaroo	<a href="https://www.mockaroo.com/">https://www.mockaroo.com/</a>	
Thiết kế giao diện	Figma	<a href="https://www.figma.com/">https://www.figma.com/</a>	
Thống kê mã nguồn	cloc	<a href="http://cloc.sourceforge.net/">http://cloc.sourceforge.net/</a>	

**Bảng 4.17:** Danh sách thư viện và công cụ sử dụng

#### 4.3.2 Kết quả đạt được

Hệ thống được đi vào hoạt động và đáp ứng được yêu cầu đề ra: tạo một môi trường cho phép người tìm kiếm việc làm và nhà tuyển dụng tìm kiếm nhau một cách dễ dàng, hiệu quả. Hệ thống được phát triển dựa trên nền tảng ứng dụng website theo mô hình client – server sử dụng bộ công nghệ MERN [2]. Chính

vì vậy, hệ thống đóng gói hai phần riêng biệt tách rời nhau đảm bảo tính loose coupling cho hệ thống.

Phần thứ nhất, FrontEnd là phần được chạy trên ứng dụng website ở máy khách. FrontEnd được phân chia thành các trang mỗi trang sẽ có các thành phần của chúng được phân cấp một cách hợp lý, nhất quán đảm bảo tính dễ bảo trì và tái sử dụng. FrontEnd được cung cấp hệ thống quản lý cache được tổ chức một cách khoa học giúp tối ưu trải nghiệm người dùng, giảm lời gọi đến phía Backend (server)...

Phần thứ hai, Backend là phần được chạy ở phía server, dưới môi trường NodeJS và được tổ chức hoạt động theo framework ExpressJS [18]. Backend chịu trách nhiệm truy xuất dữ liệu từ cơ sở dữ liệu phi quan hệ MongoDB và trả dữ liệu về cho client thông qua giao thức RESTful API. Các lời gọi API được phân luồng định hướng sử dụng framework ExpressJS thông qua một loại các middleware để có thể giao tiếp với cơ sở dữ liệu. Hệ thống có sử dụng SocketIO để lập trình thời gian thực. Để tăng tái sử dụng, Backend có sử dụng gói services tiếp nhận cả hai kiểu lời gọi từ phía client: WebSocket của SocketIO [24] và RESTful API...

Hệ thống được đi vào hoạt động đã đáp ứng được nhu cầu của người tìm việc khi họ có thể thoải mái tìm việc làm dựa trên các hồ sơ của bản thân. Hơn nữa, hồ sơ của người tìm việc hệ thống có thể dễ dàng đề xuất các ứng viên tiềm năng cho nhà tuyển dụng được giám sát bởi quản trị viên.

Các thông số của hệ thống (sử dụng thư viện cloc 4.17):

Ngôn ngữ	Số lượng tệp	Số dòng trắng	Chú thích	Mã nguồn
Javascript	94	882	404	10231
Tổng	94	882	404	10231

**Bảng 4.18:** Thống kê gói src server

Ngôn ngữ	Số lượng tệp	Số dòng trắng	Chú thích	Mã nguồn
JSX	136	631	262	12575
Javascript	77	447	279	4941
CSS	2	44	8	292
SVG	3	0	0	23
Tổng	218	1122	549	17831

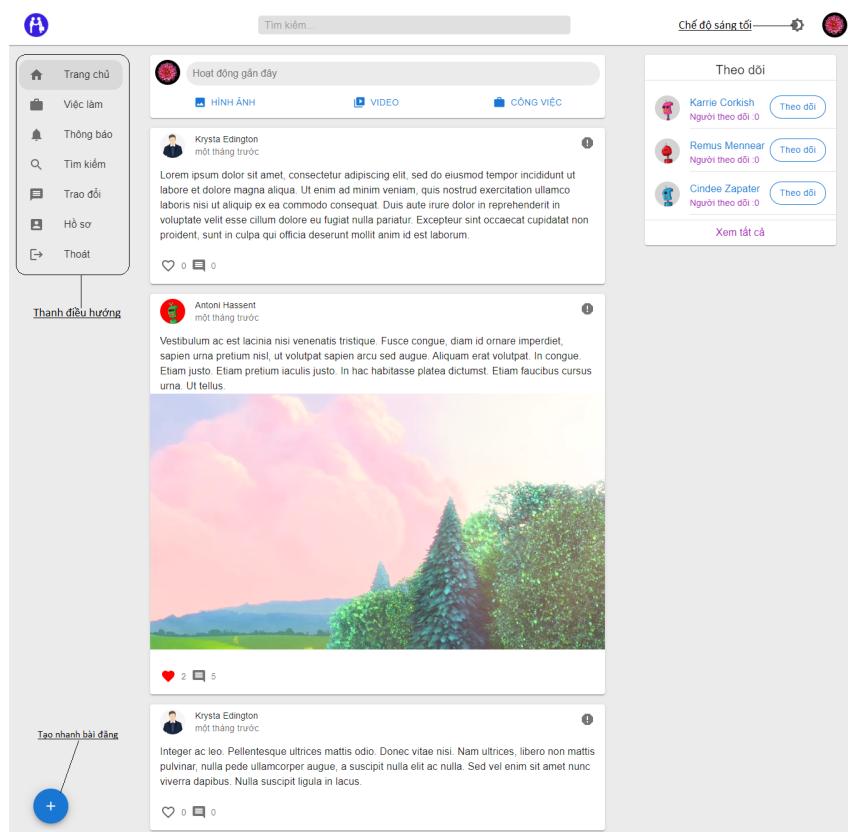
**Bảng 4.19:** Thống kê gói src client



### 4.3.3 Minh hoạ các chức năng chính

#### a, Giao diện trang chủ người dùng

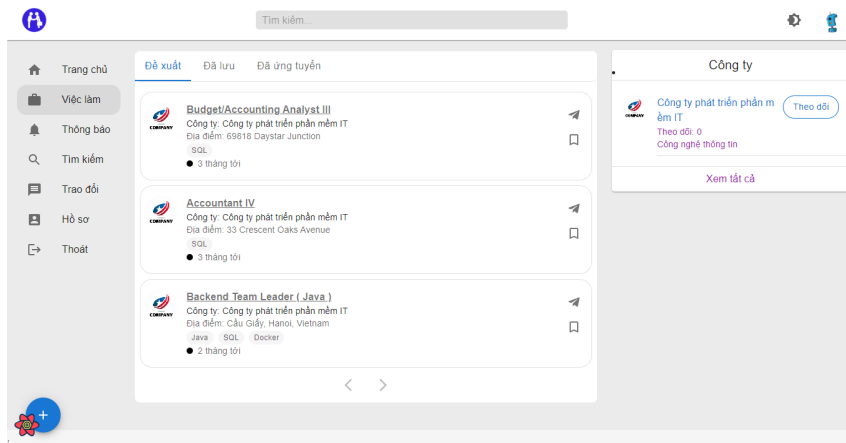
Giao diện trang chủ cho phép người dùng có thể xem danh sách các bài đăng của người dùng mình theo dõi hoặc công ty mình theo dõi. Từ đó, người dùng có thể nắm bắt các hoạt động của họ. Đồng thời, người dùng có thể tương tác, bình luận với bài đăng.



**Hình 4.19:** Giao diện trang chủ người dùng

#### b, Giao diện trang việc làm (tin tuyển dụng)

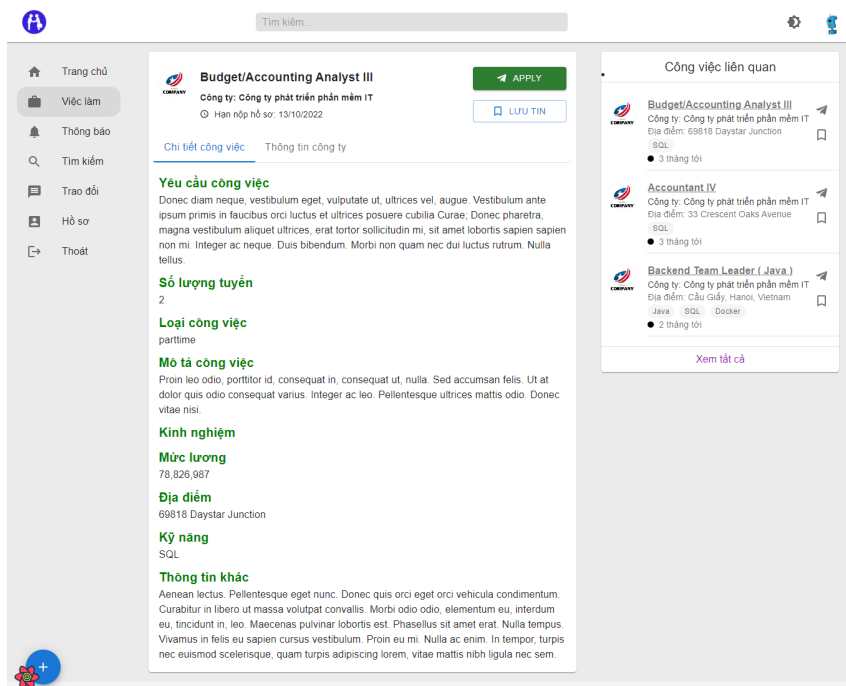
Giao diện cung cấp việc làm ở ba loại: danh sách việc làm được đề xuất cho người tìm việc – dựa trên hồ sơ của người tìm việc, danh sách việc làm mà người tìm việc quan tâm được đánh dấu là đã lưu và danh sách việc làm mà người dùng đã ứng tuyển. Mỗi việc làm bao gồm các thông tin cơ bản: tên nhan đề việc làm, tên công ty sở hữu việc làm, địa điểm làm việc, danh sách kỹ năng cần có của việc làm, thông tin về hạn chót nộp đơn ứng tuyển. Bên cạnh đó, giao diện của trang cũng cung cấp danh sách các công ty mà người dùng có thể quan tâm dựa trên hồ sơ của người dùng khai báo.



Hình 4.20: Giao diện trang việc làm

### c, Giao diện trang chi tiết việc làm

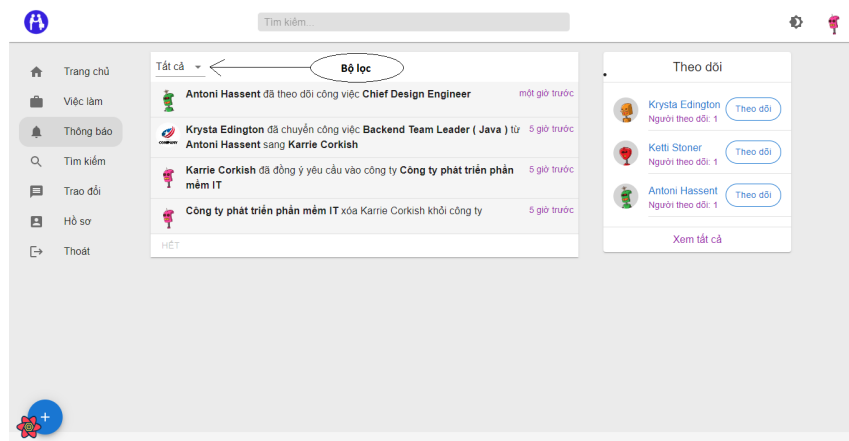
Trang chi tiết công việc cung cấp các thông tin chi tiết của một việc làm, bao gồm: mức lương, số lượng tuyển, yêu cầu đối với ứng viên... Ứng viên có thể lưu tin hoặc có thể ứng tuyển thông qua giao diện nút bấm rõ ràng, trực quan. Đối với nhà tuyển dụng sở hữu việc làm hệ thống cho phép nhà tuyển dụng xem tiến độ của công việc bao gồm: số lượng người theo dõi, số lượng ứng viên đã nộp đơn ứng tuyển. Ngoài ra, giao diện trang ở phía bên phải của màn hình cung cấp các việc làm có liên quan dựa trên danh sách kỹ năng của việc làm hiện tại đang được xem.



Hình 4.21: Giao diện trang chi tiết việc làm

#### d, Giao diện trang thông báo

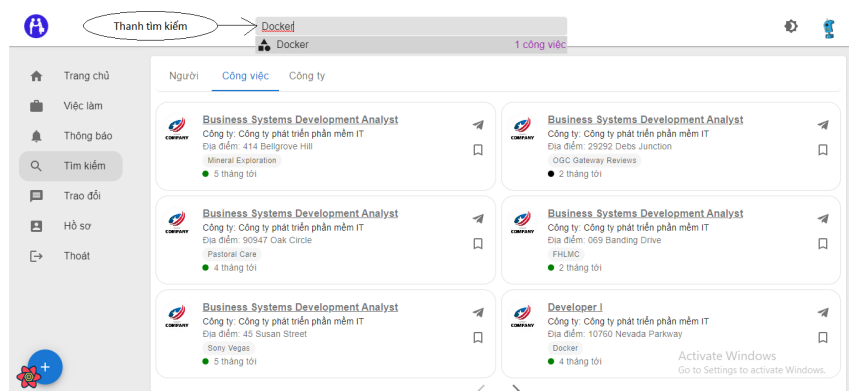
Trang thông báo là nơi mà tác nhân được cung cấp những thông tin, sự kiện nhanh nhất đến mình. Hệ thống cung cấp các sự kiện dựa vào quyền của người sử dụng có trong hệ thống. Ví dụ: đối với quản trị viên các sự kiện có thể nhận là: yêu cầu tạo công ty, yêu cầu tạo việc làm, danh sách đã duyệt việc làm... Đối với người dùng nói chung trong hệ thống: theo dõi người dùng, theo dõi công ty, theo dõi công việc, kết nối người dùng, kết nối công việc... Đối với nhà tuyển dụng: kết quả phê duyệt của công việc, thay đổi quyền sở hữu công việc...



Hình 4.22: Giao diện trang thông báo

#### e, Giao diện trang tìm kiếm

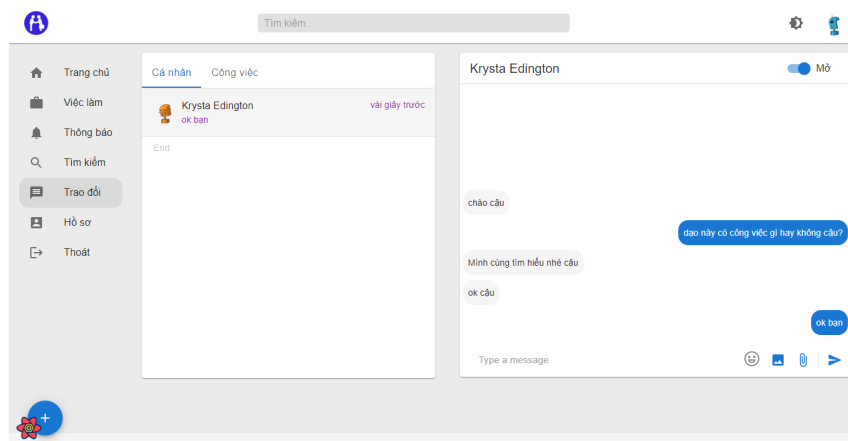
Trang tìm kiếm cung cấp cho người dùng tìm kiếm 3 loại thực thể: người dùng, công việc (tin tuyển dụng) và công ty. Hệ thống tìm kiếm đưa ra kết quả dựa trên truy vấn thông qua thanh tìm kiếm.



Hình 4.23: Giao diện trang tìm kiếm

### f, Giao diện trang trao đổi

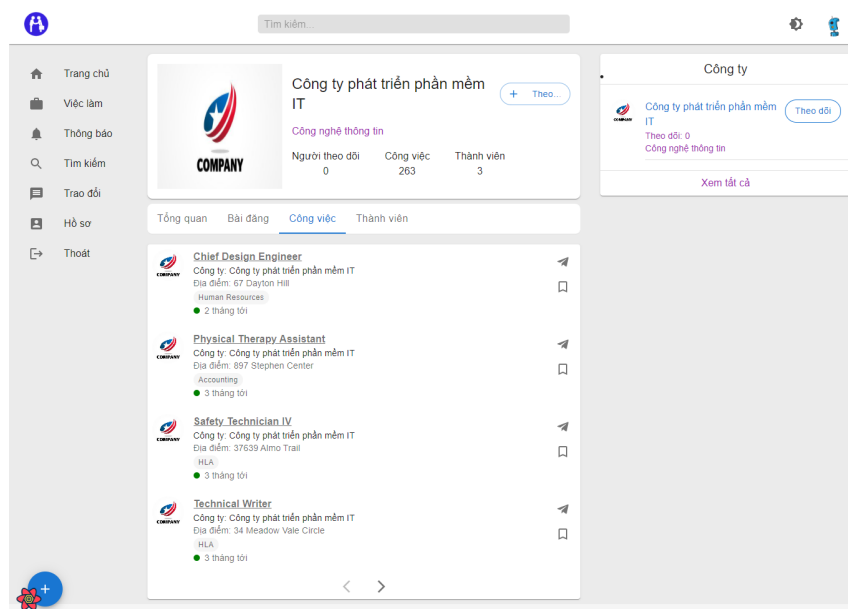
Có 2 loại trao đổi là trao đổi giữa người dùng với nhau hoặc trao đổi giữa người tìm việc và nhà tuyển dụng. Trong khi trao đổi người dùng có thể gửi ảnh, tệp, emoji... nhằm giúp cho cuộc trao đổi dễ hình dung.



Hình 4.24: Giao diện trang trao đổi

### g, Giao diện trang công ty

Giao diện trang công ty cung cấp: thông tin tổng quan của công ty, các hoạt động gần đây của công ty, việc làm của công ty, thành viên của công ty...

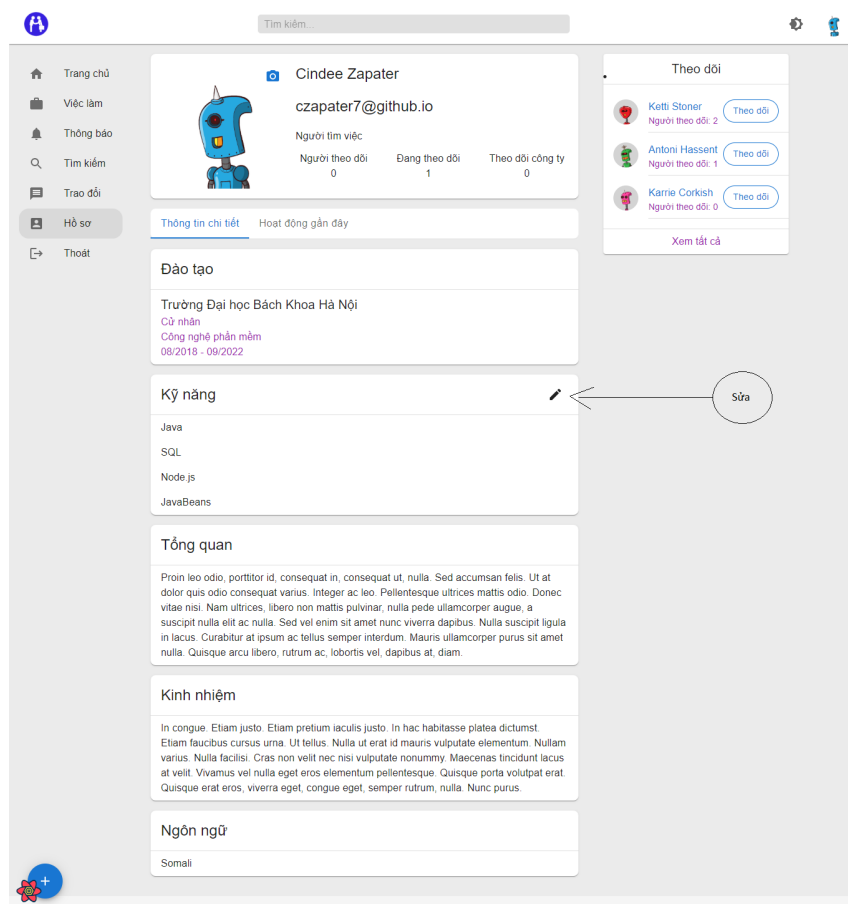


Hình 4.25: Giao diện trang công ty

### h, Giao diện trang hồ sơ người dùng

Giao diện cung cấp cho người dùng có thể xem và sửa (nếu là trang hồ sơ của bản thân): ảnh đại diện, tên người dùng, email người dùng, học vấn, kỹ năng...

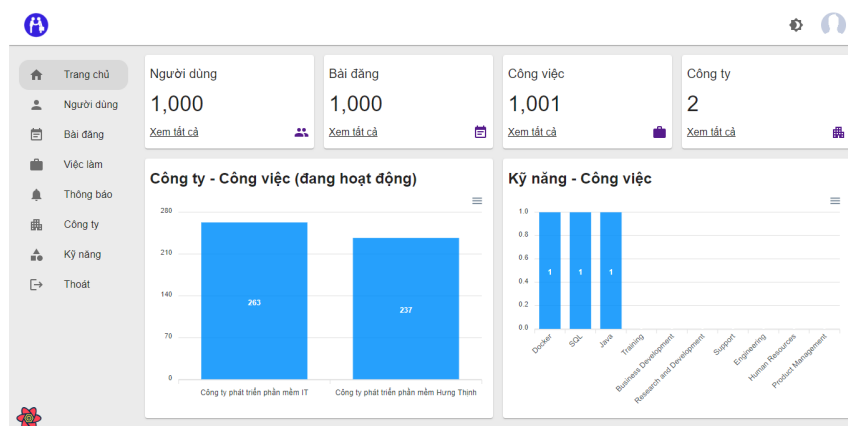
Đồng thời, người dùng có thể xem các bài đăng của mình ở đây.



**Hình 4.26:** Giao diện trang hồ sơ người dùng

### i, Giao diện trang chủ quản trị viên

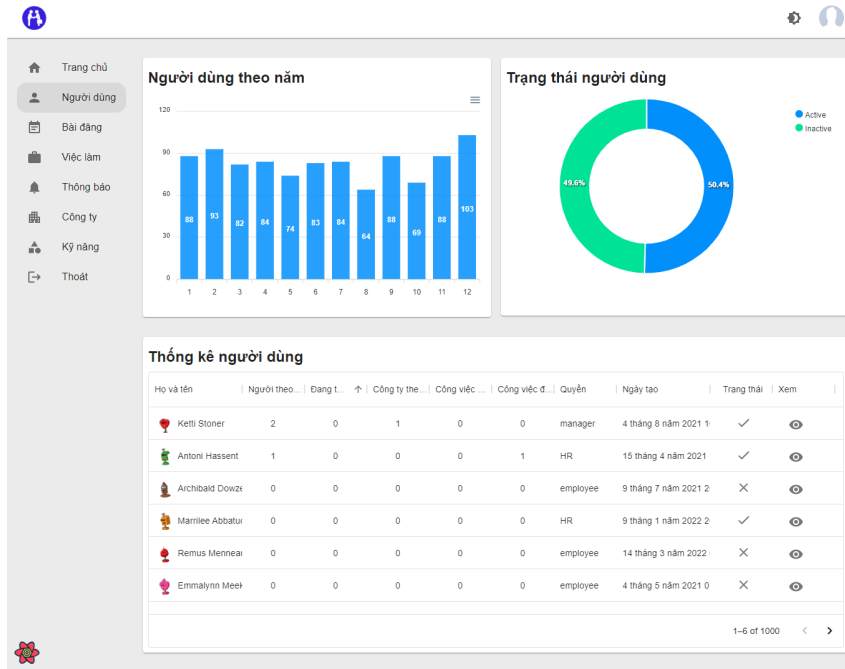
Giao diện trang chủ quản trị viên thông kê tổng quát số lượng các thực thể có trong hệ thống: người dùng, bài đăng, công ty, công việc (tin tuyển dụng). Hơn nữa, giao diện cung cấp biểu đồ thống kê.



**Hình 4.27:** Giao diện trang chủ quản trị viên

### j, Giao diện trang người dùng quản trị viên

Cung cấp các biểu đồ cho phép quản trị viên biết được số lượng người dùng đăng ký theo tháng, phần trăm số lượng người dùng bị khóa và không bị khóa tài khoản. Đồng thời, trang cung cấp bảng giúp người quản trị viên quan sát tình trạng của người dùng, thực hiện thao tác mở hay khóa tài khoản của người dùng.



**Hình 4.28:** Giao diện trang người dùng quản trị viên

#### 4.4 Kiểm thử

##### 4.4.1 Kiểm thử chức năng phê duyệt tin tuyển dụng

STT	Kịch bản test	Mô tả test case	Các bước test	Dữ liệu test	Kết quả mong đợi	Kết quả thực tế	Trạng thái
1.	Hiển thị danh sách việc làm cần xác thực	Quản trị viên muốn xem danh sách tin tuyển dụng cần duyệt	<ul style="list-style-type: none"> <li>Chọn chức năng “Thông báo tạo việc làm”</li> </ul>		Hiển thị danh sách tin tuyển dụng cần duyệt	Như mong đợi	Pass
2.	Phê duyệt việc làm	Quản trị viên chấp nhận việc làm	<ul style="list-style-type: none"> <li>Chọn chức năng “Thông báo tạo việc làm”</li> <li>Nhấn nút “Chấp nhận”</li> </ul>		<ul style="list-style-type: none"> <li>Tin tuyển dụng được mở</li> <li>Gửi thông báo đến chủ sở hữu</li> <li>Gửi thông báo đến người theo dõi công ty</li> </ul>	Như mong đợi	Pass

3.	Từ chối việc làm không thành công do thiếu trường lý do	Quản trị viên từ chối tin tuyển dụng nhưng điền thiếu trường note	<ul style="list-style-type: none"> <li>Chọn chức năng “Thông báo tạo việc làm”</li> <li>Nhấn nút “Từ chối”</li> <li>Hiện thị giao diện từ chối</li> <li>Nộp</li> </ul>	note: “”	Thông báo lỗi: “note được yêu cầu”	Như mong đợi	Pass
4.	Từ chối thành công	Quản trị viên từ chối tin tuyển dụng	<ul style="list-style-type: none"> <li>Chọn chức năng “Thông báo tạo việc làm”</li> <li>Nhấn nút “Từ chối”</li> <li>Hiện thị giao diện từ chối</li> <li>Điền đơn</li> <li>Nộp</li> </ul>	note: “yêu cầu của công việc quá mức sơ sài gây khó hiểu cho ứng viên”	<ul style="list-style-type: none"> <li>Cập nhật công việc</li> <li>Gửi thông báo từ chối đến chủ sở hữu</li> </ul>	Như mong đợi	Pass
5.	Xem danh sách đã duyệt tin tuyển dụng	Quản trị viên muốn xem danh sách các tin đã duyệt	<ul style="list-style-type: none"> <li>Chọn chức năng “Thông báo xác nhận công việc”</li> </ul>		Hiện thị danh sách tin tuyển dụng đã duyệt	Như mong đợi	Pass

**Bảng 4.20:** Kiểm thử chức năng duyệt tin tuyển dụng

#### 4.4.2 Kiểm thử chức năng ứng tuyển công việc



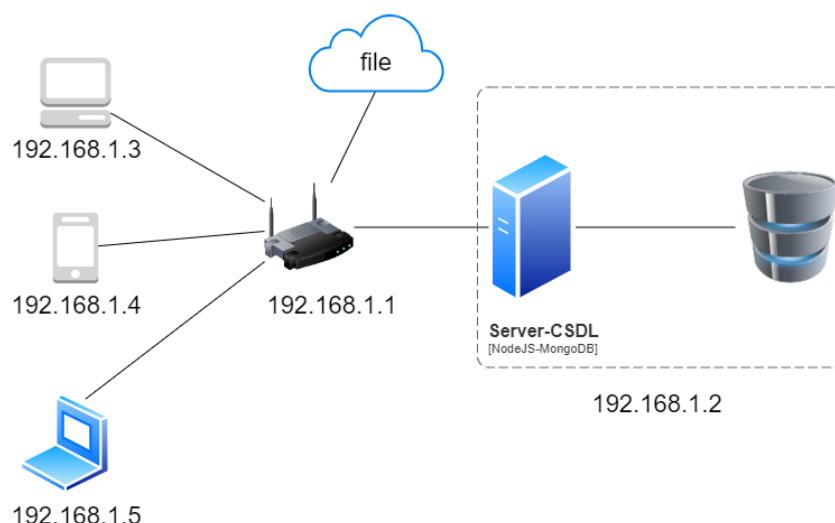
STT	Kịch bản test	Mô tả test case	Các bước test	Dữ liệu test	Kết quả mong đợi	Kết quả thực tế	Trạng thái
1.	Không thành công vì nhập sai định dạng email	Người tìm việc muốn ứng tuyển công việc nhưng điền sai định dạng email	<ul style="list-style-type: none"> <li>Nhấn nút “Apply”</li> <li>Nhập thông tin trường email</li> </ul>	email: abc	Thông báo lỗi: “email sai định dạng”	Như mong đợi	Pass
2.	Không thành công vì nhập sai số điện thoại	Người tìm việc muốn ứng tuyển công việc nhưng điền sai định dạng phone	<ul style="list-style-type: none"> <li>Nhấn nút “Apply”</li> <li>Nhập thông tin trường phone</li> </ul>	email: thanh.nd183991@sis.hust.edu.vn, phone:012345	Thông báo lỗi: “Số điện thoại không đúng định dạng”	Như mong đợi	Pass

3.	Không thành công do thiếu thông tin CV	Người tìm việc muốn ứng tuyển công việc nhưng không nộp CV	<ul style="list-style-type: none"> <li>• Nhấn nút “Apply”</li> <li>• Điền form ứng tuyển</li> <li>• Nộp đơn</li> </ul>	email: thanh.nd183991@sis.hust.edu.vn, phone:0386851140	Thông báo lỗi: “Vui lòng chọn CV”	Như mong đợi	Pass
4.	Không thành công do đã hết hạn ứng tuyển	Người tìm việc muốn ứng tuyển công việc nhưng đã quá hạn ứng tuyển	<ul style="list-style-type: none"> <li>• Nhấn nút “Apply”</li> <li>• Điền form ứng tuyển</li> <li>• Nộp đơn</li> </ul>	email: thanh.nd183991@sis.hust.edu.vn, phone:0386851140, CV: <a href="https://orimi.com/pdf-test.pdf">https://orimi.com/pdf-test.pdf</a>	Thông báo lỗi: “Hết hạn ứng tuyển”	Như mong đợi	Pass
5.	Không thành công do đã ứng tuyển	Người tìm việc muốn ứng tuyển công việc nhưng đã ứng tuyển trước đó	<ul style="list-style-type: none"> <li>• Nhấn nút “Apply”</li> <li>• Điền form ứng tuyển</li> <li>• Nộp đơn</li> </ul>	email: thanh.nd183991@sis.hust.edu.vn, phone:0386851140, CV: <a href="https://orimi.com/pdf-test.pdf">https://orimi.com/pdf-test.pdf</a>	Thông báo lỗi: “Bạn đã ứng tuyển rồi”	Như mong đợi	Pass

6.	Ứng tuyển thành công	Người tìm việc ứng tuyển thành công	<ul style="list-style-type: none"> <li>• Nhấn nút “Apply”</li> <li>• Điền form ứng tuyển</li> <li>• Nộp đơn</li> </ul>	email: thanh.nd183991@sis.hust.edu.vn, phone:0386851140, CV: https://orimi.com/pdf-test.pdf	<ul style="list-style-type: none"> <li>• Cập nhật danh sách ứng tuyển</li> <li>• Gửi thông báo đến nhà tuyển dụng</li> <li>• Thông báo: “Apply thành công”</li> </ul>	Như mong đợi	Pass
----	----------------------	-------------------------------------	--	---	---	--------------	------

**Bảng 4.21:** Bảng kiểm thử chức năng ứng tuyển công việc

#### 4.5 Triển khai

**Hình 4.29:** Mô hình triển khai của hệ thống

Hệ thống được đi vào thử nghiệm và chạy ổn định trên mạng LAN (Local Area Network). Máy chủ và cơ sở dữ liệu được chạy trên cùng một máy có thể giao tiếp với máy khách ở trong cùng một mạng cục bộ. Máy chủ chạy ở cổng 5000 và cơ sở dữ liệu chạy ở cổng 27017. Các file của hệ thống được lưu trên cloud điều này giúp làm giảm các tác vụ trên máy server. Các máy khách có thể truy cập vào hệ thống thông qua đa dạng các loại thiết bị: desktop, laptop, mobile. Các thiết bị sẽ được hệ thống phản hồi giao diện phù hợp với từng loại kích cỡ mà loại thiết bị có đảm bảo hiệu quả, tối ưu trải nghiệm người sử dụng.

Thông tin cấu hình	Mô tả
Laptop	ASUS S551LN
Hệ điều hành	Windows 10 Pro 64-bit operating system, x64-based processor
RAM	8G
Màn hình	1366 x 768 (32bit) (60Hz)

**Bảng 4.22:** Bảng thông số cấu hình server

## CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

### 5.1 Phân trang và cuộn vô hạn danh sách dữ liệu

#### 5.1.1 Đặt vấn đề

Trong quá trình hoạt động của hệ thống, dữ liệu của các bảng trong cơ sở dữ liệu tăng lên một cách nhanh chóng. Chính vì vậy, hệ thống cần có một cơ chế truy vấn dữ liệu hợp lý từ client đến server vì nó sẽ ảnh hưởng nhiều đến tốc độ tải trang và trải nghiệm người dùng.

#### 5.1.2 Giải pháp

Có hai giải pháp chính được sử dụng hiện nay khi muốn truy xuất danh sách dữ liệu:

Phương pháp 1: Lấy tất cả dữ liệu từ cơ sở dữ liệu, rồi gửi trả lại client qua lời gọi API.

Phương pháp 2: Lấy một phần của danh sách phù hợp với các truy vấn từ phía client.

Đối với phương pháp thứ 1, việc truy vấn danh sách rất dễ dàng và dễ thực hiện. Các lời gọi từ client đến server sẽ chỉ cần 1 lời gọi để lấy toàn bộ danh sách dữ liệu cần tìm nạp. Phía client tùy vào mục đích có thể hiển thị toàn bộ dữ liệu hoặc là phân trang dựa trên danh sách dữ liệu tìm nạp được từ server để không phá vỡ bố cục của giao diện. Tuy nhiên, việc truy vấn toàn bộ danh sách dữ liệu cần tìm gặp khó khăn khi dữ liệu cần tìm lên đến hàng nghìn, thậm chí hàng triệu bản ghi. Điều này làm tăng mất mát dữ liệu khi truyền, tăng độ trễ khi tải, dẫn đến làm giảm trải nghiệm người dùng.

Đối với phương pháp thứ 2, lời gọi truy vấn lấy danh sách dữ liệu cần tìm nạp sẽ được phân thành các trang. Phương pháp này là hợp lý đối với dữ liệu lớn, bởi vì, người sử dụng có thể chỉ xem một phần dữ liệu lên việc phân trang này sẽ tránh dư thừa dữ liệu. Hơn thế nữa, phương pháp này giúp điều khiển dữ liệu được trả về client từ server làm giảm tính mất mát dữ liệu và giúp việc lập trình bên phía client tự do thể hiện bố cục mà mình muốn.

Phương pháp thứ 2 được thiết lập dựa trên hai tham số: page (trang) và limit (giới hạn). Page là thông số cho biết số thứ tự mà trang muốn lấy, trong khi đó, limit là số lượng bản ghi mà mình muốn lấy.

Trong đề tài, hệ thống có áp dụng cả hai phương pháp nhằm tối ưu hiệu suất của hệ thống và tăng trải nghiệm người sử dụng. Phương pháp 1 được sử dụng khi

nhà tuyển dụng muốn xem danh sách các ứng viên đã ứng tuyển vào công việc mà mình quản lý. Điều này là hoàn toàn phù hợp, bởi vì dựa trên thực tế, số lượng tuyển của một tin tuyển dụng thường là ít (1 - 10 ứng viên) số lượng ứng viên ứng tuyển thường giao động từ 1 – 20 người. Phương pháp 2 được sử dụng đối với các truy vấn tìm nạp dữ liệu lớn như tìm người dùng, tìm việc, công ty... Sau khi nhận được các trang dữ liệu từ người dùng, hệ thống có thể hiện thị dữ liệu theo hai hình thức: phân trang và cuộn vô hạn. Hai hình thức này đều được hỗ trợ bởi thư viện React Query. Đối với phân trang[31], hệ thống áp dụng ở các trang tìm kiếm, trang việc làm của công ty, các trang của quản trị viên... Đối với cuộn vô hạn [32], hệ thống sẽ đọc trang tiếp theo khi người dùng cuộn đến cuối danh sách được áp dụng với các trang: trang chủ, trang trao đổi, trang thông báo. Đặc biệt ở trang chủ, do các bài đăng có thể đính kèm video, để có trải nghiệm tốt nhất cho người dùng hệ thống thiết lập phát video tự động khi video hiển thị trên màn hình và dừng lại khi video được cuộn ra khỏi màn hình.

## 5.2 Tạo CV

### 5.2.1 Đặt vấn đề

CV (Curriculum Vitae) là một bản tóm tắt thông tin cá nhân, kinh nghiệm làm việc, hoạt động, kỹ năng... ứng viên gửi cho nhà tuyển dụng. CV là cơ sở chính để nhà tuyển dụng chọn ứng viên bước vào vòng phỏng vấn. CV là bước đầu tiên mà ứng viên cần chuẩn bị khi ứng tuyển việc làm.

### 5.2.2 Giải pháp

Có rất nhiều giải pháp cho phương pháp này.

Phương pháp 1: tạo CV sử dụng MS Word (Microsoft Word). MS Word giúp chúng ta tự do trong việc tạo ra bố cục cho CV, tự do thêm các thông tin về bản thân trong CV. Tuy nhiên, tạo CV trên MS Word dễ gây thiếu ý, bố cục thiếu cân xứng do được tự do trong việc tổ chức.

Phương pháp 2: tạo CV sử dụng các template có sẵn trên Internet. Cách này giúp ứng viên dễ dàng điền đúng và đủ thông, đồng thời, bố cục của CV cũng được đảm bảo. Tuy nhiên, cách này gây khó khăn khi ứng viên phải mất thời gian điền thông tin vào CV.

Phương pháp 3: Hệ thống hỗ trợ việc tạo CV dựa trên hồ sơ cá nhân mà ứng viên đã kê khai trong hệ thống. Cách này có ưu điểm là tận dụng được tối đa thông tin sẵn có mà người dùng đã kê khai. Đồng thời, việc sử dụng chung một mẫu CV tuân thủ một bố cục sẵn có giúp nhà tuyển dụng dễ dàng tìm kiếm các thông tin mong muốn từ ứng viên.

Hệ thống dựa trên phương pháp 3 sử dụng thư viện react-pdf[33] cho việc tạo CV dựa trên các thông tin được người dùng nhập trên hồ sơ cá nhân của họ.

### 5.3 Thiết kế Category Collection

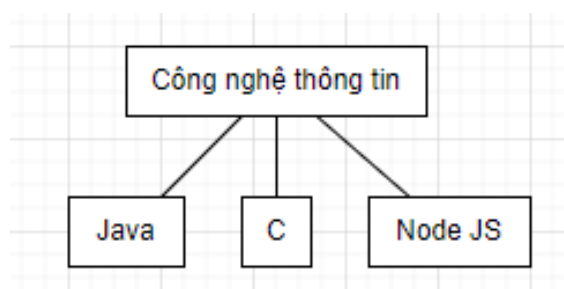
#### 5.3.1 Đặt vấn đề

Khi nhà tuyển dụng tạo một tin tuyển dụng, hệ thống sẽ lưu thông tin tuyển dụng đồng thời lưu danh sách các kỹ năng của công việc đó trong Category Collection. Tuy nhiên, việc lưu thông tin các kỹ năng cần phải được đảm bảo sao cho việc truy vấn gợi ý, tìm kiếm các tin tuyển dụng của người dùng phải chính xác nhất. Đồng thời thông qua Category Collection, hệ thống đề xuất cho nhà tuyển dụng những ứng viên tiềm năng phù hợp với công việc họ quản lý. Chính vì vậy, việc thiết kế Category Collection đóng vai trò quan trọng khi trong việc xây dựng mục đích của hệ thống – tạo ra môi trường giúp nhà tuyển dụng và người lao động (tìm việc) tìm đến nhau dễ dàng hơn.

#### 5.3.2 Giải pháp

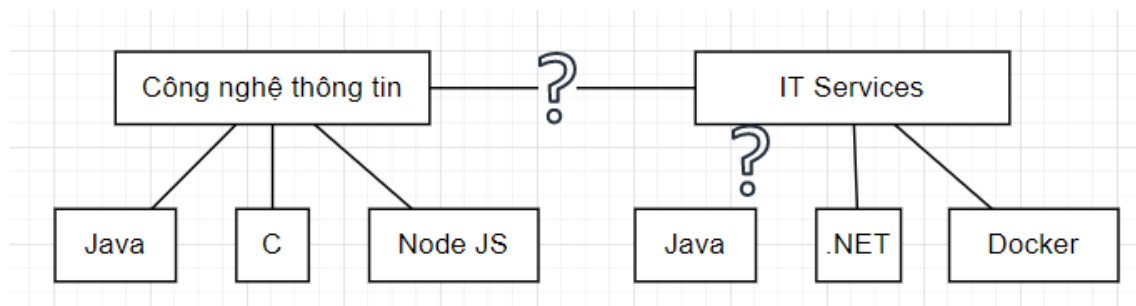
Khi nhà quản trị công ty muốn tham gia vào hệ thống, họ sẽ nộp lên hệ thống và chờ đợi xác nhận của quản trị viên. Hệ thống sẽ căn cứ vào đơn tạo yêu cầu để chèn thêm hoặc tạo mới lĩnh vực của công ty vào hệ thống. Ví dụ: Công ty ABC có lĩnh vực Công nghệ thông tin, hệ thống sẽ căn cứ vào lĩnh vực có trong biểu mẫu tìm trong Category Collection nếu chưa có sẽ tạo mới 1 document cho lĩnh vực này. Việc tìm kiếm lĩnh vực có trong Category Collection sẽ căn cứ vào slug (một chuỗi trong đó khoảng cách giữa các ký tự là các dấu gạch ngang và chỉ sử dụng các chữ cái in thường) giúp đảm bảo tính chính xác, tránh trùng lặp. Ví dụ: Công nghệ thông tin, công nghệ Thông tin, Công nghe thông tin... được chuyển đổi qua slug là: cong-nghe-thong-tin.

Khi nhà tuyển dụng của một công ty muốn tạo một bài viết liên quan đến công ty đó, hệ thống coi lĩnh vực của công ty đó như là nút cha của các kỹ năng trong tin tuyển dụng mà nhà tuyển dụng tạo. Hệ thống cũng sử dụng slug cho việc tìm kiếm và chèn tập các kỹ năng của công việc (tin tuyển dụng).



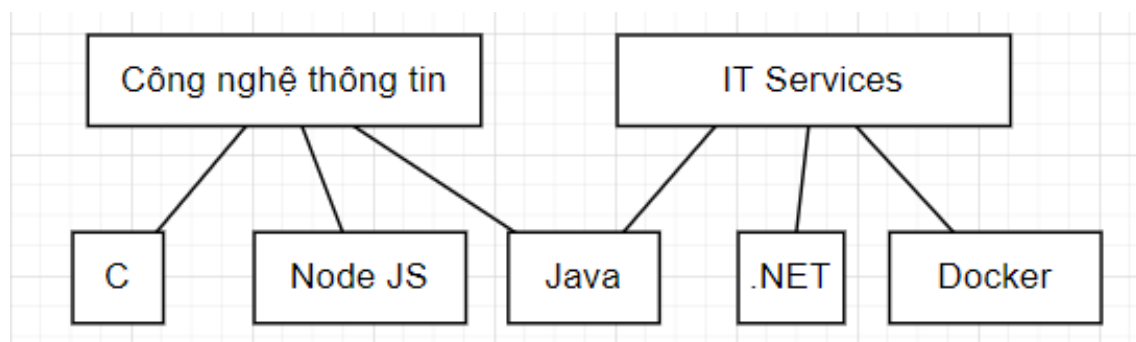
**Hình 5.1:** Mối quan hệ kỹ năng khi tạo tin tuyển dụng

Ở mô hình này 5.1, ta có thể thấy việc truy vấn để tìm danh sách kỹ năng của một công ty, tìm danh sách công việc theo kỹ năng, tìm danh sách công ty có kỹ năng phù hợp với bản thân là hoàn toàn được. Theo mô hình này 5.1, một kỹ năng chỉ có thể thuộc một lĩnh vực cụ thể. Điều này là hoàn toàn đúng, tuy nhiên trên thực tế, một lĩnh vực có thể có nhiều tên gọi khác nhau tùy thuộc vào vùng miền nơi họ sinh sống. Ví dụ: ở Việt Name, Công nghệ thông tin sẽ được phổ biến hơn là gọi IT Services hoặc IT Service Support and Solutions và ngược lại. Điều đó gây khó khăn khi truy vấn, gợi ý các việc làm hoặc ứng viên mà các kỹ năng của họ có các lĩnh vực tương đồng 5.2.



**Hình 5.2:** Vấn đề mối quan hệ giữa lĩnh vực tương đồng

Từ những điều đã phân tích ở trên, hệ thống cần thiết kế Category Collection sao cho đáp ứng các kỹ năng phải thuộc nhiều lĩnh vực (các lĩnh vực tương đồng chỉ khác nhau tên gọi), mối quan hệ giữa các lĩnh vực được ngầm hiểu là cùng lĩnh vực chỉ khác tên gọi. Ví dụ: Java thuộc hai lĩnh vực Công nghệ thông tin và IT Services đều này được ngầm hiểu là Công nghệ thông tin là tên gọi khác của IT Services.



**Hình 5.3:** Thiết kế mối quan hệ giữa lĩnh vực tương đồng

Nhìn vào hình vẽ 5.3, ta có thể dễ dàng liên kết mạng lưới kỹ năng với lĩnh vực. IT Services có kỹ năng Java lên IT Services là lĩnh vực tương đồng với Công nghệ thông tin. Từ đó, IT Services có các kỹ năng: C, NodeJS, Java, .NET, Docker. Việc xây dựng danh sách kỹ năng cũng như danh sách lĩnh vực ban đầu rất quan trọng, hệ thống có sử dụng trang website: <https://www.mockaroo.com/> trong việc tạo dữ liệu cho Category Collection. Điều này giúp hình thành lên mạng lưới công việc,



đảm bảo tính đúng đắn, chính xác của dữ liệu. Từ đó, hệ thống đề xuất, truy vấn công việc, ứng viên một cách hiệu quả, chính xác.

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
name	String	Unique	Tên của kỹ năng, lĩnh vực
slug	String	Unique	Chuỗi thực hiện truy vấn
parents	Array		Danh sách mã lĩnh vực của kỹ năng
jobs	Array		Danh sách công việc của kỹ năng
users	Array		Danh sách người dùng sở hữu kỹ năng

**Bảng 5.1:** Thiết kế Category Collection

## 5.4 Thiết kế Notification Collection

### 5.4.1 Đặt vấn đề

Hệ thống này là một mạng xã hội hướng việc làm, vì vậy, hệ thống cần có một mạng lưới thông báo nhanh chóng, chính xác những thông tin mới nhất về công ty (doanh nghiệp), người dùng. Dịch vụ thông báo của hệ thống phải bao gồm đa dạng các loại thông báo xung quanh các thực thể của hệ thống: người sử dụng, bài đăng, công việc, công ty.

### 5.4.2 Giải pháp

Dịch vụ thông báo tối thiểu phải đảm bảo 3 trường: người gửi (sender), người nhận (receiver), đối tượng truy cập (access - đối tượng được nhắc đến trong thông báo) và kiểu thông báo. Trường người gửi phải đảm bảo có thể được gửi từ hai thực thể: User và Company. Trường người nhận phải đảm bảo có thể được nhận từ hai thực thể: User và Company. Trường đối tượng truy cập có thể nhận các thực thể: User, Company, Job, Post, Room. Trường kiểu thông báo cung cấp loại thông báo giúp xác định các hành động đối với thông báo đó (ví dụ: thông báo có kiểu là tạo công ty để tạo trải nghiệm tốt nhất thì quản trị viên có thể chấp nhận/từ chối ngay trên chính thông báo mà không phải lọc rồi mới duyệt công ty).

Notification Collection được thiết kế để giải quyết các vấn đề. Thông báo 1 – 1 từ người gửi đến người nhận trong các trường hợp: theo dõi người dùng, yêu cầu trao đổi, theo dõi công việc (tin tuyển dụng)... Thông báo 1 – n từ người gửi đến người nhận trong các trường hợp: đăng bài viết cá nhân, đăng bài viết công ty, tin tuyển dụng mới với người dùng... Đồng thời, các thông báo 1 – n hoặc 1 – n cần có cơ chế chấp nhận hoặc từ chối thông báo (bảng 4.15). Ví dụ: từ chối lời mời vào công ty, từ chối kết nối trao đổi đối với 1 – 1, hoặc từ chối/ chấp nhận tạo mới tin tuyển dụng đối với 1 – n.

## 5.5 Lưu trữ file

### 5.5.1 Đặt vấn đề

Trong bất kỳ hệ thống này việc upload file lên server cũng là một chức năng quan trọng, việc lưu trữ file ở đâu có thể gây ảnh hưởng nhiều đến hiệu suất, tốc độ thực thi của hệ thống.

### 5.5.2 Giải pháp

Có khá nhiều phương pháp để xem xét giải quyết vấn đề này.

Phương pháp 1: Lưu file trực tiếp trên database dưới dạng base64 hoặc tệp nhị phân (blob). Cách này có ưu điểm là sẽ tiết kiệm chi phí cho việc mua/thuê các dịch vụ lưu trữ trên Cloud. Tuy vậy, phương pháp này gây ảnh hưởng đến hiệu suất đọc ghi của Backend và cơ sở dữ liệu.

Phương pháp 2: Lưu file trên thư mục của server. Khi client gửi file nhị phân thì server sẽ lưu file này với tên thường được tạo ngẫu nhiên (ví dụ: tên được tạo bởi thư viện uuid) rồi sau đó lưu tên file trong cơ sở dữ liệu. Ưu điểm của phương pháp này là việc truy xuất file rất nhanh vì file được đặt luôn tại server. Tuy nhiên, phương pháp này chỉ áp dụng đối với hệ thống có số lượng file nhỏ vì khi file có số lượng lớn dễ dẫn đến quá tải bộ nhớ ở server.

Phương pháp 3: Lưu file trên Cloud Storage qua backend. Client gửi file lên server sử dụng định dạng request multipart form data. Phía server chuyển tiếp file lên Cloud Storage lưu trữ. Cloud Storage gửi trả đường dẫn file cho server lưu vào cơ sở dữ liệu. Phương pháp này tốt hơn hai phương pháp trước do file được lưu trữ remote không gây ảnh hưởng gì đến server và cơ sở dữ liệu. Tuy nhiên, phương pháp này có một hạn chế là server trở thành trung gian giữa client và Cloud Storage gây mất thời gian truyền file.

Phương pháp 4: Client trực tiếp upload file lên Cloud Storage. Phương pháp này dựa trên việc khi client thực hiện upload file không cần truyền qua server. Tuy nhiên để đảm bảo tính bảo mật, client đầu tiên sẽ yêu cầu server cấp cho một chữ ký. Chữ ký này sẽ giúp Cloud Storage xác minh tính hợp lệ. Sau đó, client sẽ gửi đường dẫn file đã upload về server.

Phương pháp 4 được sử dụng trong đồ án với dịch vụ lưu trữ file của clouinary [34] giúp lưu các file: ảnh đại diện, CV, ảnh bài đăng, video...

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Hệ thống xây dựng đã cung cấp được các chức năng cơ bản giúp nhà tuyển dụng và người lao động (tìm việc) có một kênh thông tin có thể giao tiếp với nhau một cách hiệu quả và đáng tin cậy. Nhà tuyển dụng có thể đăng tin tuyển dụng, tìm kiếm các ứng viên tiềm năng và có thể quảng bá cho công ty của mình qua kênh thông tin chung của hệ thống,... Người lao động được cung cấp một mạng lưới các doanh nghiệp tiềm năng dựa trên việc khai báo hồ sơ cá nhân của mình từ đó tìm được các công việc phù hợp với bản thân.

Hệ thống là một mạng xã hội hướng việc làm vì vậy so với Facebook hay mạng các trang mạng xã hội khác thì hệ thống cung cấp một cách chính xác, hiệu quả các công việc cho người lao động không khiến cho họ phải tìm kiếm vất vả như các trang mạng xã hội thông thường.

Hệ thống cung cấp dịch vụ thông báo hai chiều giữa người lao động và nhà tuyển dụng giúp họ có thể tìm kiếm nhau dễ dàng hơn. Hơn nữa, các thông tin trong hệ thống đều được quản trị viên kiểm duyệt đảm bảo tính đúng đắn. Chính vì vậy, hệ thống sẽ cung cấp các thông tin hữu ích so với các trang thông tin tìm việc hiện nay như [timviecnhanh.vn](http://timviecnhanh.vn), [vietnamworks.com](http://vietnamworks.com),... chỉ cung cấp một chiều từ ứng viên đến nhà tuyển dụng.

Hệ thống có một mạng lưới các bảng tin giúp người tìm kiếm việc làm nắm bắt thông tin chính xác về các hoạt động của người dùng và công ty mình quan tâm. Đối với nhà tuyển dụng, bảng tin giúp cho họ nắm bắt thông tin nhanh chóng những hoạt động xung quanh các ứng viên tiềm năng. Người sử dụng không phải mất công phải “chuyển” tab hay lướt web các trang mạng xã hội khác để có thể hiểu hơn về công ty, điều này sẽ cải thiện hơn so với các trang website tìm kiếm việc làm hiện nay như: [topCV](http://topCV), [ITViec](http://ITViec)...

Hệ thống cung cấp một cách rạch ròi về việc trao đổi giữa việc trao đổi cá nhân và công việc. Từ đó, người lao động có thể trao đổi lẫn nhau để hiểu hơn về công ty mà họ quan tâm dựa theo ý kiến chủ quan của họ. Hơn nữa, hệ thống phân chia như vậy sẽ giúp nhà quản lý dễ dàng phân chia công việc của công ty mình một cách hiệu quả, đó là điểm mạnh mà hệ thống đem lại so với các hệ thống mạng xã hội tìm kiếm việc làm hiện nay như [LinkedIn](http://LinkedIn).

Qua quá trình làm đồ án tốt nghiệp, tôi đã học được nhiều công nghệ mới, cũng như kỹ năng viết báo cáo. Việc làm đồ án tốt nghiệp giúp tôi vận dụng những gì đã

học được xuyên suốt quá trình học đại học: phân tích hệ thống, thiết kế giao diện, thiết kế cơ sở dữ liệu, thiết kế hệ thống, kiểm thử sản phẩm... đó sẽ là hành trang vững chắc trong công việc sau này.

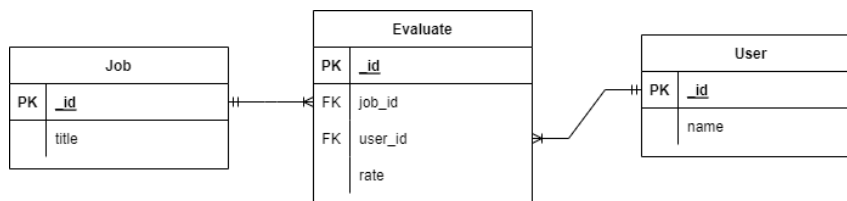
Do thời gian làm đồ án có hạn, cũng như hạn chế của năng lực bản thân nên hệ thống xây dựng vẫn còn nhiều thiếu sót. Hệ thống vẫn chưa có chức năng đăng nhập bằng bằng mạng xã hội Facebook, tài khoản Google, việc thiết kế cơ sở dữ liệu còn thiếu sót...

## 6.2 Hướng phát triển

Để chương trình có thể hoạt động hiệu quả, dữ liệu phải được lưu trữ bằng hệ quản trị cơ sở dữ liệu có tính bảo mật hơn. Hệ thống cần có sự giao tiếp song song giữa quản trị viên và người dùng một cách chặt chẽ và nhiều hơn để tối ưu trải nghiệm của người dùng.

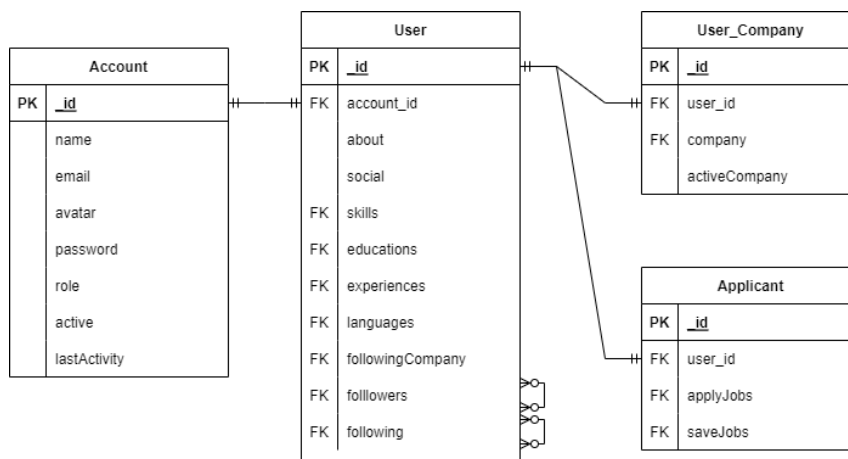
Hệ thống cần mở rộng chức năng đăng nhập cho người dùng sử dụng thư viện PassportJS để người dùng có thể đăng nhập từ các bên thứ 3 giúp làm giảm thời gian nhập liệu cho người dùng.

Hệ thống cần cải thiện thêm tính năng đánh giá công việc thông qua các phiếu đánh giá từ chính người tìm việc của hệ thống. Để làm được điều đó, ta cần tạo thêm một Evaluate Collection trong database được liên kết khóa ngoại với id của công việc mà người tìm việc đánh giá.



**Hình 6.1:** Thiết kế Evaluate Collection

Hiện tại trong bảng User Collection đang chứa tất cả các thông tin về người dùng 4.11. Cách làm này có hiệu quả ban đầu khi người dùng còn ít và tận dụng được lợi thế của MongoDB khi có thể lưu linh động nhiều loại dữ liệu, truy vấn nhanh chóng. Tuy nhiên thiết kế như vậy gây ra khó khăn trong việc mở rộng hệ thống. Ví dụ: tăng thêm tác nhân trong hệ thống, dữ liệu thiếu sự chuẩn hóa gây lỗi tiềm tàng, tăng thêm nghiệp vụ đối với người dùng hệ thống (livestream, video call)...

**Hình 6.2:** Chuẩn hóa User Collection

Theo hình 6.2, User Collection đã được chia làm 4 collection bao gồm: Account Collection phụ trách việc lưu trữ các thông tin cơ bản khi người dùng đăng nhập vào hệ thống, User Collection lưu trữ các thông tin hồ sơ, thông tin theo dõi. User\_Company Collection lưu trữ thông tin cho 2 tác nhân: Manager công ty và nhà tuyển dụng. Applicant Collection lưu trữ thông tin của ứng viên.

Hiện tại hệ thống được xác thực người dùng thông qua JWT đảm bảo được mỗi người dùng chỉ được phép truy cập và sử dụng những thông tin mà mình được phép. Tuy nhiên, hệ thống vẫn chưa có cơ chế bảo mật trong dịch vụ trao đổi, trò chuyện giữa người dùng và công việc. Hệ thống cần phải khắc phục điều này sử dụng Double Ratchet Algorithm [35] để bảo mật các cuộc trò chuyện của người dùng tránh rò rỉ dữ liệu ra bên ngoài.

Hệ thống cần được chỉnh sửa để có thể tương tác dễ dàng qua thiết bị màn hình như: di động, iPad, iWatch. Đồng thời, hệ thống cần phải chỉnh sửa lại giao diện phối màu, bố cục... để tăng thêm trải nghiệm của người dùng.

## TÀI LIỆU THAM KHẢO

- [1] H. Subramanian and P. Raj, *Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs*. Packt Publishing Ltd, 2019.
- [2] *What is the mern stack? introduction examples mongodb*. [Online]. Available: <https://www.mongodb.com/mern-stack> (visited on 07/14/2022).
- [3] A. Biryukov, D. Dinu, and D. Khovratovich, "Argon2: New generation of memory-hard functions for password hashing and other applications," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2016, pp. 292–302.
- [4] P. Wilton, *Beginning JavaScript*. John Wiley & Sons, 2004.
- [5] D. Herron, *Node.js Web Development: Server-side development with Node 10 made easy*. Packt Publishing Ltd, 2018.
- [6] E. Wang, S. Zurowski, O. Duffy, T. Thomas, and I. Baggili, "Juicing v8: A primary account for the memory forensics of the v8 javascript engine," *Forensic Science International: Digital Investigation*, vol. 42, p. 301 400, 2022.
- [7] C. Gackenheim and A. Paul, *Introduction to React*. Springer, 2015, vol. 52.
- [8] R. Caers, T. De Feyter, M. De Couck, T. Stough, C. Vigna, and C. Du Bois, "Facebook: A literature review," *New media & society*, vol. 15, no. 6, pp. 982–1002, 2013.
- [9] R. F. Grove and E. Ozkan, "The mvc-web design pattern," in *International Conference on Web Information Systems and Technologies*, SCITEPRESS, vol. 2, 2011, pp. 127–130.
- [10] J. Sugrue, *Beginning Backbone.js*. Apress, 2014.
- [11] B. Green and S. Seshadri, *AngularJS*. "O'Reilly Media, Inc.", 2013.
- [12] P. A. Kumar, N. S. Kumar, and N. C. Shekar, "Structure–compressibility correlation among mobx," *Journal of Physics and Chemistry of Solids*, vol. 163, p. 110 599, 2022.
- [13] A. GraphQL, *Apollo graphql | supergraph: Unify apis, microservices, databases in a composable graph*. [Online]. Available: <https://www.apollographql.com/> (visited on 07/29/2022).
- [14] *Lostpebble/pullstate: Simple state stores using immer and react hooks - reuse parts of your state by pulling it anywhere you like!* [Online]. Available: <https://github.com/lostpebble/pullstate> (visited on 07/29/2022).

- [15] *Rxjs*. [Online]. Available: <https://rxjs.dev/> (visited on 07/29/2022).
- [16] D. T. Kudiabor, “State management with react-redux,” 2020.
- [17] *Tanstack query | react query, solid query, svelte query, vue query*. [Online]. Available: <https://tanstack.com/query/v4> (visited on 07/14/2022).
- [18] A. Mardan, *Express.js Guide: The Comprehensive Book on Express.js*. Azat Mardan, 2014.
- [19] R. Fielding, J. Gettys, J. Mogul, *et al.*, *Rfc2616: Hypertext transfer protocol—http/1.1*, 1999.
- [20] *Rfc 7519 - json web token (jwt)*. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7519> (visited on 07/14/2022).
- [21] *Rfc 2104 - hmac: Keyed-hashing for message authentication*. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc2104> (visited on 07/14/2022).
- [22] *Rfc 8017 - pkcs 1: Rsa cryptography specifications version 2.2*. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc8017> (visited on 07/14/2022).
- [23] *Giới thiệu kỹ thuật long-polling, websockets, server-sent events, comet*. [Online]. Available: <https://techmaster.vn/posts/33693/ky-thuat-long-polling-websockets-server-sent-events-comet> (visited on 07/15/2022).
- [24] *Socket.io*. [Online]. Available: <https://socket.io/> (visited on 07/15/2022).
- [25] A. Boicea, F. Radulescu, and L. I. Agapin, “Mongodb vs oracle—database comparison,” in *2012 third international conference on emerging intelligent data and web technologies*, IEEE, 2012, pp. 330–335.
- [26] *Introduction to mongodb — mongodb manual*. [Online]. Available: <https://www.mongodb.com/docs/manual/introduction/> (visited on 07/15/2022).
- [27] D. Chauhan and K. Bansal, “Using the advantages of nosql: A case study on mongodb,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 2, pp. 90–93, 2017.
- [28] *Normalization vs denormalization in database modeling*. [Online]. Available: <https://www.linkedin.com/pulse/normalization-vs-denormalization-rohit-prasad/> (visited on 07/15/2022).
- [29] M. Potel, “Mvp: Model-view-presenter the taligent programming model for c++ and java,” *Taligent Inc*, vol. 20, 1996.

- [30] D. D. Li and X. Y. Liu, “Research on mvp design pattern modeling based on mda,” *Procedia Computer Science*, vol. 166, pp. 51–56, 2020.
- [31] *Paginated/lagged queries | tanstack query docs*. [Online]. Available: <https://tanstack.com/query/v4/docs/guides/paginated-queries> (visited on 07/16/2022).
- [32] *Infinite queries | tanstack query docs*. [Online]. Available: <https://tanstack.com/query/v4/docs/guides/infinite-queries> (visited on 07/16/2022).
- [33] *React-pdf*. [Online]. Available: <https://react-pdf.org/> (visited on 07/16/2022).
- [34] *Upload api reference | cloudinary*. [Online]. Available: [https://cloudinary.com/documentation/image\\_upload\\_api\\_reference](https://cloudinary.com/documentation/image_upload_api_reference) (visited on 07/16/2022).
- [35] T. Perrin and M. Marlinspike, “The double ratchet algorithm,” *GitHub wiki*, 2016.



# PHỤ LỤC

## A. ĐẶC TẢ USE CASE

### A.1 Đặc tả use case “Theo dõi người dùng”

<b>Mã Use case</b>	UC14-1	<b>Tên Use case</b>	Theo dõi người dùng
<b>Tác nhân</b>	Người dùng		
<b>Tiền điều kiện</b>	Hiện ở trang profile của người dùng muốn theo dõi		
<b>Luồng sự kiện chính(Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Click nút “Theo dõi” người dùng
	2.	Hệ thống	Kiểm tra người dùng hợp lệ
	3.	Hệ thống	Thông báo theo dõi người dùng thành công
	4.	Hệ thống	Gửi thông báo đến người dùng được theo dõi
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	2a.	Hệ thống	Người dùng hiện đang bị khóa
	3a.	Hệ thống	Bỏ theo dõi người dùng
	3b	Hệ thống	Đã theo dõi người dùng thì chuyển đến phòng trao đổi
<b>Hậu điều kiện</b>	Cập nhật lại Feed của người theo dõi		

**Bảng A.1:** Đặc tả usecase “Theo dõi người dùng”

### A.2 Đặc tả use case “Đăng bài”

<b>Mã Use case</b>	UC12	<b>Tên Use case</b>	Đăng bài
<b>Tác nhân</b>	Người dùng		
<b>Tiền điều kiện</b>	Đang thao tác với hệ thống		

Luồng sự kiện chính (Thành công)	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Nhấn nút đăng bài
	2	Hệ thống	Hiển thị giao diện đăng bài
	3.	Người dùng	Điền đầy đủ các trường thông tin (*)
	4.	Người dùng	Nhấn nút “Nộp”
	5.	Hệ thống	Kiểm tra xem các trường thông tin nhập vào đã hợp lệ hay chưa
	6.	Hệ thống	Lưu bài đăng vào hệ thống
	7.	Hệ thống	Hiển thị thông báo tạo bài đăng thành công
	8	Hệ thống	Gửi thông báo đến các người dùng quan tâm
Luồng sự kiện thay thế	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	5a.	Hệ thống	thông báo lỗi: Hiển thị thông báo các trường thông tin còn thiếu hoặc không hợp lệ
	6a.	Hệ thống	thông báo lỗi: Lỗi upload ảnh khi bài đăng có file đính kèm
	6b	Hệ thống	thông báo lỗi: Lỗi tạo bài đăng khi không có đủ quyền để tạo đối với bài đăng công ty
Hậu điều kiện	Cập nhật lại danh sách bài đăng		

**Bảng A.2:** Đặc tả usecase “Đăng bài”

Dữ liệu đầu vào (\*)

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	content	Nội dung của bài viết	Có	Chuỗi string	Viết gì đó

2	files	Tệp đính kèm mỗi bài đăng	Không	Mảng đường dẫn đến tệp đính kèm	[{ file: http://ab.com, re-source_type: "image", original_filename: "abc.jpg" }]
---	-------	---------------------------	-------	---------------------------------	--

### A.3 Đặc tả use case “Trao đổi”

Mã Use case	UC15	Tên Use case	Trao đổi
<b>Tác nhân</b>	Người dùng		
<b>Tiền điều kiện</b>	Đã ở trang profile của người dùng cần trao đổi		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	1.	Người dùng	Nhấn nút “trao đổi”
	2.	Hệ thống	Kiểm tra người dùng hợp lệ
	3.	Hệ thống	Gửi thông báo kết nối đến người được trao đổi
	4	Hệ thống	Thông báo chờ người dùng khác duyệt trao đổi
	5	Hệ thống	Thông báo chấp nhận hoặc từ chối trao đổi
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	2a.	Hệ thống	Báo không tìm thấy người dùng
	3a	Hệ thống	Chuyển hướng đến phòng đã được mở giữa hai người dùng.
<b>Hậu điều kiện</b>	Không		

**Bảng A.4:** Đặc tả usecase “Trao đổi”