**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**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: Công nghệ thông tin**

**Chuyên ngành: Công nghệ thông tin**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Nguyễn Tiến Thành |
| **Bộ môn:** | Công nghệ phần mềm |
| **Trường:** | 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 Đồ án tốt nghiệp  Thành  Nguyễn Đức Thành |

# Lời cảm ơn

Trước hết em xin chân thành cảm ơn thầy Thạc Sỹ Nghuyễn 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 em rất nhiều trong suốt quá trình tìm hiểu làm đồ án tốt nghiệp. Đồng thời, em xin cám ơn các thầy cô trong trường đã cung cấp cho em những kiến thức cơ bản, cần thiết để em hoàn thành đồ án tốt nghiệp này.

Em 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 em hoàn thành tốt học phần.

Sau cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè đã tạo mọi điều kiện để em xây dựng thành công học phần này.

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

# Tóm tắt

Ngày nay, các ứng dụng công nghệ thông tin đã được áp dụng trong mọi lĩnh vực nghiên cứu khoa học kĩ 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 mẩu tin trên báo. Người tim 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 dụng tuyển dụng được đăng khắp mọi nơi trên Internet: mạng xã hội, trang website của công ty, mẩu 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 thiếu khuyết điểm của việc áp dụng công nghệ thông tin đối với lĩnh vực tìm kiếm việc làm, em 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 lao động 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.

# Mục lục

[Lời cam kết ii](#_Toc108374968)

[Lời cảm ơn iii](#_Toc108374969)

[Tóm tắt iv](#_Toc108374970)

[Mục lục v](#_Toc108374971)

[Danh mục hình vẽ ix](#_Toc108374972)

[Danh mục bảng xi](#_Toc108374973)

[Danh mục công thức xii](#_Toc108374974)

[Danh mục các từ viết tắt xiii](#_Toc108374975)

[Danh mục thuật ngữ xiv](#_Toc108374976)

[Chương 1 Giới thiệu đề tài 1](#_Toc108374977)

[1.1 Đặt vấn đề 1](#_Toc108374978)

[1.2 Mục tiêu và phạm vi đề tài 1](#_Toc108374979)

[1.3 Định hướng giải pháp 2](#_Toc108374980)

[1.4 Bố cục đồ án 3](#_Toc108374981)

[Chương 2 Khảo sát và phân tích yêu cầu 4](#_Toc108374982)

[2.1 Khảo sát hiện trạng 4](#_Toc108374983)

[2.2 Tổng quan chức năng 6](#_Toc108374984)

[2.2.1 Biểu đồ use case tổng quan 6](#_Toc108374985)

[2.2.2 Biểu đồ use case phân rã “Quản lý người dùng” 10](#_Toc108374986)

[2.2.3 Biểu đồ use case phân rã “Quản lý bài đăng” 11](#_Toc108374987)

[2.2.4 Biểu đồ use case phân rã “Quản lý công ty” 12](#_Toc108374988)

[2.2.5 Biểu đồ use case phân rã “Quản lý tin tuyển dụng” 13](#_Toc108374989)

[2.2.6 Biểu đồ use case phân rã “Quản lý kỹ năng” 14](#_Toc108374990)

[2.2.7 Biểu đồ use case phân rã “Theo dõi” 15](#_Toc108374991)

[2.2.8 Biểu đồ use case phân rã “Tìm kiếm” 16](#_Toc108374992)

[2.2.9 Biểu đồ use case phân rã “Quản lý công ty của tôi” 17](#_Toc108374993)

[2.2.10 Quy trình nghiệp vụ 18](#_Toc108374994)

[2.3 Đặc tả chức năng 22](#_Toc108374995)

[2.3.1 Đặc tả use case “Quên mật khẩu” 22](#_Toc108374996)

[2.3.2 Đặc tả use case “Theo dõi người dùng” 24](#_Toc108374997)

[2.3.3 Đặc tả use case “Cập nhật thông tin” 25](#_Toc108374998)

[2.3.4 Đặc tả usecase “Đăng bài” 27](#_Toc108374999)

[2.3.5 Đặc tả use case “Trao đổi” 29](#_Toc108375000)

[2.3.6 Đặc tả use case “Ứng tuyển công việc” 29](#_Toc108375001)

[2.3.7 Đặc tả use case “Đăng tin tuyển dụng” 31](#_Toc108375002)

[2.4 Yêu cầu phi chức năng 33](#_Toc108375003)

[2.4.1 Yêu cầu bảo mật 33](#_Toc108375004)

[2.4.2 Yêu cầu vận hành 33](#_Toc108375005)

[2.4.3 Yêu cầu giao diện 34](#_Toc108375006)

[Chương 3 Công nghệ sử dụng 35](#_Toc108375007)

[3.1 Ngôn ngữ lập trình Javascript 35](#_Toc108375008)

[3.2 Công nghệ lập trình giao diện ReactJS 35](#_Toc108375009)

[3.3 Công nghệ Redux 37](#_Toc108375010)

[3.4 Công nghệ React Query 38](#_Toc108375011)

[3.5 Công nghệ lập trình máy chủ NodeJS 39](#_Toc108375012)

[3.6 Công nghệ JWT 40](#_Toc108375013)

[3.7 Công nghệ lập trình thời gian thực SocketIO 40](#_Toc108375014)

[3.8 Cơ sở dữ liệu MongoDB 40](#_Toc108375015)

[Chương 4 Phát triển và triển khai ứng dụng 42](#_Toc108375016)

[4.1 Thiết kế kiến trúc 42](#_Toc108375017)

[4.1.1 Lựa chọn kiến trúc phần mềm 42](#_Toc108375018)

[4.1.2 Thiết kế tổng quan 43](#_Toc108375019)

[4.1.3 Thiết kế chi tiết gói 47](#_Toc108375020)

[4.2 Thiết kế chi tiết 54](#_Toc108375021)

[4.2.1 Thiết kế giao diện 54](#_Toc108375022)

[4.2.2 Thiết kế lớp 58](#_Toc108375023)

[4.2.3 Thiết kế cơ sở dữ liệu 62](#_Toc108375024)

[4.3 Xây dựng ứng dụng 70](#_Toc108375025)

[4.3.1 Thư viện và công cụ sử dụng 70](#_Toc108375026)

[4.3.2 Kết quả đạt được 71](#_Toc108375027)

[4.3.3 Minh hoạ các chức năng chính 72](#_Toc108375028)

[4.4 Kiểm thử 81](#_Toc108375029)

[4.4.1 Kiểm thử chức năng phê duyệt tin tuyển dụng 81](#_Toc108375030)

[4.4.2 Kiểm thử chức năng ứng tuyển công việc 83](#_Toc108375031)

[4.5 Triển khai 85](#_Toc108375032)

[Chương 5 Các giải pháp và đóng góp nổi bật 87](#_Toc108375033)

[5.1 Phân trang và cuộn vô hạn danh sách dữ liệu 87](#_Toc108375034)

[5.1.1 Đặt vấn đề 87](#_Toc108375035)

[5.1.2 Giải pháp 87](#_Toc108375036)

[5.2 Thiết kế Category Collection 88](#_Toc108375037)

[5.2.1 Đặt vấn đề 88](#_Toc108375038)

[5.2.2 Giải pháp 88](#_Toc108375039)

[5.3 Thiết kế Notification Collection 90](#_Toc108375040)

[5.3.1 Đặt vấn đề 90](#_Toc108375041)

[5.3.2 Giải pháp 91](#_Toc108375042)

[5.4 Lưu trữ file 91](#_Toc108375043)

[5.4.1 Đăt vấn đề 91](#_Toc108375044)

[5.4.2 Giải pháp 91](#_Toc108375045)

[5.5 Tạo CV 92](#_Toc108375046)

[5.5.1 Đặt vấn đề 92](#_Toc108375047)

[5.5.2 Giải pháp 92](#_Toc108375048)

[Chương 6 Kết luận và hướng phát triển 93](#_Toc108375049)

[6.1 Kết luận 93](#_Toc108375050)

[6.2 Hướng phát triển 94](#_Toc108375051)

[Tài liệu tham khảo 95](#_Toc108375052)

[Phụ lục 1](#_Toc108375053)

# Danh mục hình vẽ

[**Hình 1** Sơ đồ Usecase tổng quan 6](#_Toc108086976)

[**Hình 2** Phân rã Usecase “Quản lý người dùng” 10](#_Toc108086977)

[**Hình 3** Phân rã Usecase “Quản lý bài đăng” 11](#_Toc108086978)

[**Hình 4** Phân rã Usecase “Quản lý công ty” 12](#_Toc108086979)

[**Hình 5** Phân rã Usecase “Quản lý tin tuyển dụng” 13](#_Toc108086980)

[**Hình 6** Phân rã Usecase “Quản lý kỹ năng” 14](#_Toc108086981)

[**Hình 7** Phân rã Usecase “Theo dõi” 15](#_Toc108086982)

[**Hình 8** Phân rã Usecase “Tìm kiếm” 16](#_Toc108086983)

[**Hình 9** Phân rã Usecase “Quản lý công ty của tôi” 17](#_Toc108086984)

[**Hình 10** Biểu đồ hoạt động đăng tin tuyển dụng 20](#_Toc108086985)

[**Hình 11** Biều đồ hoạt động thay đổi chủ của tin tuyển dụng 22](#_Toc108086986)

[**Hình 12** Cấu trúc, thành phần của ReactJS 36](#_Toc108086987)

[**Hình 13** Redux design pattern. 37](#_Toc108086988)

[**Hình 14** Luồng Redux với middleware 38](#_Toc108086989)

[**Hình 15** MVC Design Pattern 42](#_Toc108086990)

[**Hình 16** MVP Design Pattern 43](#_Toc108086991)

[**Hình 17** Biều đồ phụ thuộc gói Client 44](#_Toc108086992)

[**Hình 18** Tổng quan gói Server 46](#_Toc108086993)

[**Hình 19** Thiết kế gói thành phần giao diện 47](#_Toc108086994)

[**Hình 20** Thiết kế chi tiết gói AuthClient 49](#_Toc108086995)

[**Hình 21** Thiết kế chi tiết gói AuthServer 50](#_Toc108086996)

[**Hình 22** Thiết kế màn hình trang chủ 52](#_Toc108086997)

[**Hình 23** Thiết kế gói nghiệp vụ việc làm 53](#_Toc108086998)

[**Hình 24** Bảng màu với giao diện sáng 55](#_Toc108086999)

[**Hình 25** Bảng màu với giao diện sáng 56](#_Toc108087000)

[**Hình 26** Thiết kế giao diện trang việc làm 57](#_Toc108087001)

[**Hình 27** Thiết kế trang chi tiết công việc 58](#_Toc108087002)

[**Hình 28** Biểu đồ tuần tự createJob 60](#_Toc108087003)

[**Hình 29** Biểu đồ tuần tự createPost 62](#_Toc108087004)

[**Hình 30** Biều đồ thực thể liên kết 63](#_Toc108087005)

[**Hình 31** Biểu đồ cơ sở dữ liệu phi quan hệ 69](#_Toc108087006)

[**Hình 32** Giao diện trang chủ người dùng 73](#_Toc108087007)

[**Hình 33** Giao diện trang việc làm 74](#_Toc108087008)

[**Hình 34** Giao diện trang việc làm 75](#_Toc108087009)

[**Hình 35** Giao diện trang thông báo 76](#_Toc108087010)

[**Hình 36** Giao diện tìm kiếm 76](#_Toc108087011)

[**Hình 37** Giao diện trang trao đổi 77](#_Toc108087012)

[**Hình 38** Giao diện trang hồ sơ người dùng 78](#_Toc108087013)

[**Hình 39** Giao diện trang công ty 79](#_Toc108087014)

[**Hình 40** Giao diện trang chủ quản trị viên 80](#_Toc108087015)

[**Hình 41** Giao diện trang người dùng quản trị viên 81](#_Toc108087016)

[**Hình 42** Mô hình triển khai của hệ thống 85](#_Toc108087017)

[**Hình 43** Mối quan hệ kỹ năng khi tạo tin tuyển dụng 89](#_Toc108087018)

[**Hình 44** Mối quan hệ giữa lĩnh vực tương đồng 89](#_Toc108087019)

[Hình 45 Thiết kế mối quan hệ giữa lĩnh vực tương đồng 90](#_Toc108087020)

# Danh mục bảng

[**Bảng 1** Quy trình đăng tin tuyển dụng 19](#_Toc107061571)

[**Bảng 2** Quy trình thay đổi chủ của tin tuyển dụng 21](#_Toc107061572)

[**Bảng 3** Đặc tả usecase “Quên mật khẩu” 24](#_Toc107061573)

[**Bảng 4** Đặc tả use case “Theo dõi người dùng” 25](#_Toc107061574)

[**Bảng 5** Đặc tả use case “Cập nhật thông tin” 26](#_Toc107061575)

[**Bảng 6** Đặc tả use case “Đăng bài” 28](#_Toc107061576)

[**Bảng 7** Đặc tả use case “Trao đổi” 29](#_Toc107061577)

[**Bảng 8** Đặc tả use case “Ứng tuyển công việc” 30](#_Toc107061578)

[**Bảng 9** Đặc tả use case “Đăng tin tuyển dụng” 32](#_Toc107061579)

[**Bảng 16** Danh sách thư viện và công cụ sử dụng 70](#_Toc107061580)

# Danh mục công thức

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **IT** | Information Technology  Công nghệ thông tin |
| **UI** | User Interface  Giao diện người dùng |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **CNTT** | Công nghệ thông tin |
| **CSS** | Cascading Style Sheets  Là ngôn ngữ định dạng văn bản |
| **SV** | Sinh viên |
| **JWT** | JSON Web Tokens  Chuỗi xác thực người dùng |
| **SSR** | SERVER-SIDE RENDERING  Cơ chế hiển thị trang website |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **Compiler** | Trình biên dịch |
| **Javascript** | Là ngôn ngữ lập trình |
| **ReactJS** | Một thư viện viết bằng **javascript**, dùng để **xây dựng giao diện người dùng**. |
| **MUI** | Thư viện hỗ trợ ReactJS xây dựng giao diện. |
| **NodeJS** | 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++ |
| **MongoDB** | Cơ sơ dữ liệu phi quan hệ |
| **Server** | Là 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. |
| **JWT** | Giao thức xác thực người dùng trên internet |
| **Redux** | Công cụ quản lý trạng thái của giao diện |
| **RSA** | Một thuật toán mật mã hóa khóa công khai. |
| **MVC** | Mô hình kiến trúc ứng dụng |
| **MVVM** | Mô hình kiến trúc ứng dụng |
| **Client** | Máy khách |
| **JSON** | JavaScript Object Notation  Một định dạng trao đổi dữ liệu nhẹ. |
| **base64** | Phương thức chuyển đổi dạng mã hóa 2 chiều từ binary sang string để có thể gửi đi được trong network một cách dễ dàng |
| **Public key** | Khóa công khai khi mã hóa |
| **Private key** | Khóa bí mật |

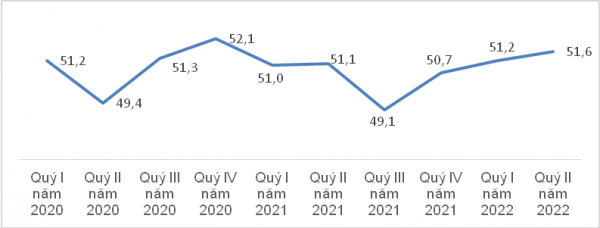
# Giới thiệu đề tài

## 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 mẩu 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 thực 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, 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. Trên trang website chính thức của Tổng cục thống kê Việt Nam (https://www.gso.gov.vn/), 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** Lực lượng lao động theo quý, 2020 – 2022 (Đơn vị tính: Triệu người)

Nhìn vào biểu đồ ta thấy, số lượng người lao động có việc trong các quý gần đây có dấu hiện tăng trưởng. Tuy nhiên, người lao động khi tìm kiếm việc làm sau dịch bệnh cần vô cùng cảnh giác, cẩn sàng lọc các thông tin việc làm tránh bị các đối tượng xấu lợi dụng, dụ dỗi. Gần đây, các công việc như: tuyển cộng tác viên bán hàng online, cộng tác viên 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.

Hiểu được điều đó, người viết đồ 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 tuyể dụng, giúp họ tìm đến nhau một cách dễ dàng.

## Mục tiêu và phạm vi đề tài

Mục tiêu 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 “tới tay” người lao động. Đối với nhà quản lý công ty, hệ thống cung cấp các chức năng cho phép họ giảm 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 và thông báo để giúp các thành viên trong mạng lưới giao tiếp với nhau một cách trực tiếp, dễ dàng và hiệu quả.

## 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 so với các thiết bị sử dụng giao diện cũ hơn. Hệ thống sẽ tránh hoàn toàn việc sử dụng giao diện dòng lệnh, bởi vì, nó khó ghi nhớ câu lệnh và phải nhập câu lệnh văn bản. Ưu điểm của việc sử 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 sẽ được 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 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. Xây dựng giao diện người dùng sử dụng công nghệ ReactJS, 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. cùng với framework Express 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…

## 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 rả 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 liên quan đến bảo mật (phần 2.4.1), vận hành (phần 2.4.2) và giao diện người dùng (phần 2.4.3).

Chương 3 trình bày về các công nghệ được sử dụng trong hệ thống. Thông qua việc phân tích bài toán ở chương 2, người viết đồ án bắt đầu lựa chọn những công nghệ phù hợp để thực thi hệ thống dựa trên 3 phần cơ bản, riêng biệt: back-end, front-end và cơ sở dữ liệu.

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

# ­­Khảo sát và phân tích yêu cầu

## 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 mẩu tin được đăng trên các mặt báo, mẩu 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 điểm hình là Facebook. Với đông đảo số lượng người dùng Facebook hiện nay thông qua các hội nhóm Facebook, người tim việc và nhà tuyển dụng điều 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.

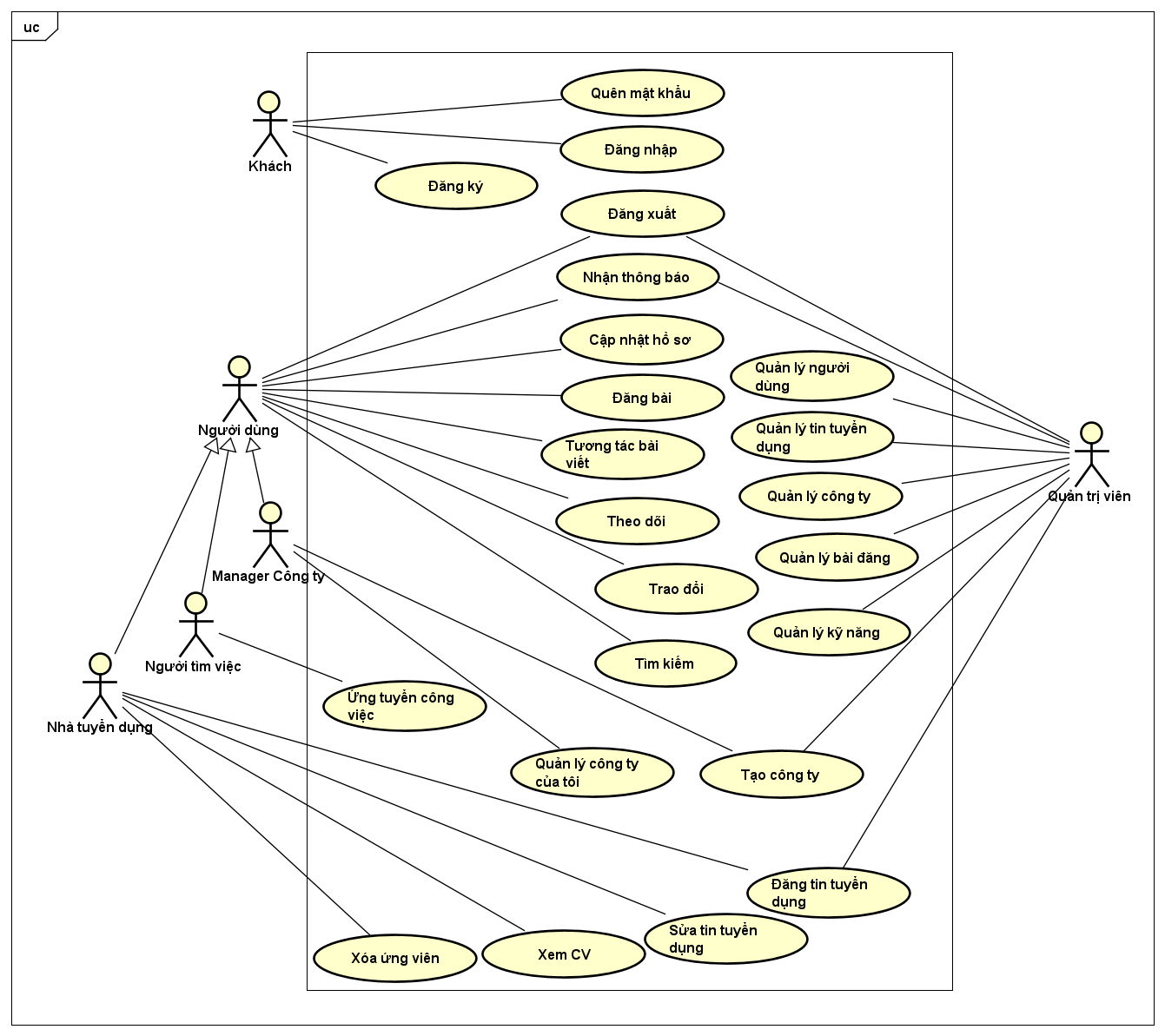
Cũng có một cách khác đó chính là vào các trang tìm kiếm việc làm như: topcv, ITViec, timviecnhanh.com… Các hệ thống website này cung cấp một cách cung cấp đầy đủ các chức năng giúp nhà tuyển dụng đăng tải các công việc của mình, cùng với đó là việc phân chia các công việc theo kỹ năng công ty một cách 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 đấy của công ty,… điều này gây khó khắn 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 đồ 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 đồ 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.

## Tổng quan chức năng

### Biểu đồ use case tổng quan



**Hình 2** Sơ đồ Usecase tổng quan

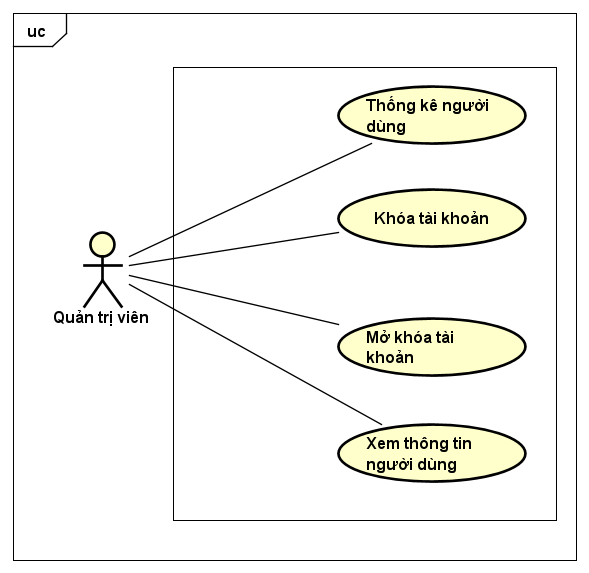
Danh sách tác nhân:

|  |  |  |
| --- | --- | --- |
| 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. |

Mô tả các usecase chính

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên Usecase | 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 hoat 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: ngươì dừng 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 |

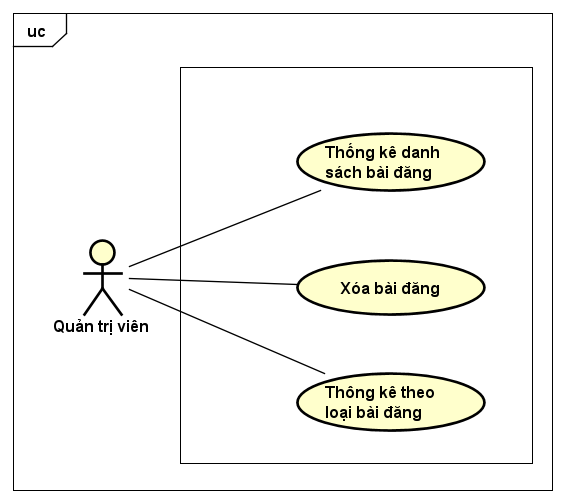
### Biểu đồ use case phân rã “Quản lý người dùng”



**Hình 3** Phân rã Usecase “Quản lý người dùng”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | Mô tả Usecase | Tác nhân tương tác |
| 1 | UC06-1 | Thống kê người dùng | Thông kế 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 |

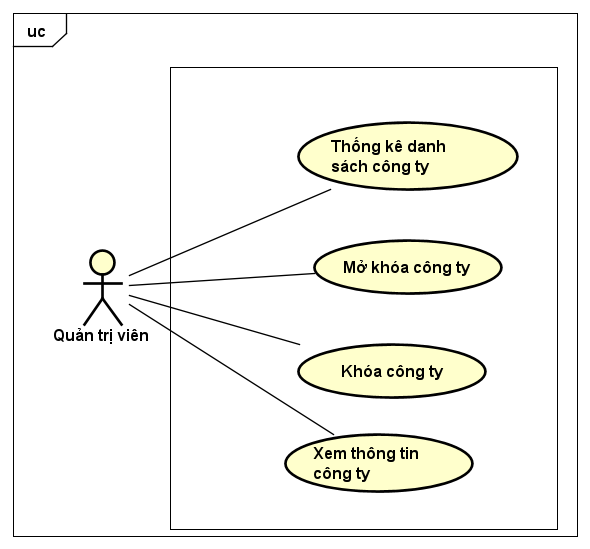
### Biểu đồ use case phân rã “Quản lý bài đăng”



**Hình 4** Phân rã Usecase “Quản lý bài đăng”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | 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 |

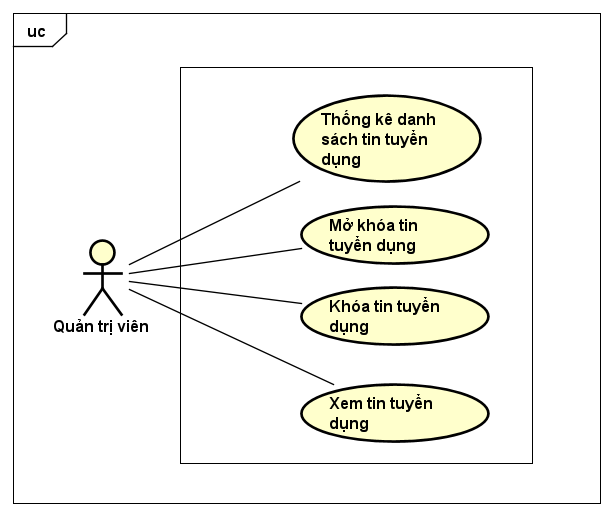
### Biểu đồ use case phân rã “Quản lý công ty”



**Hình 5** Phân rã Usecase “Quản lý công ty”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | 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 |

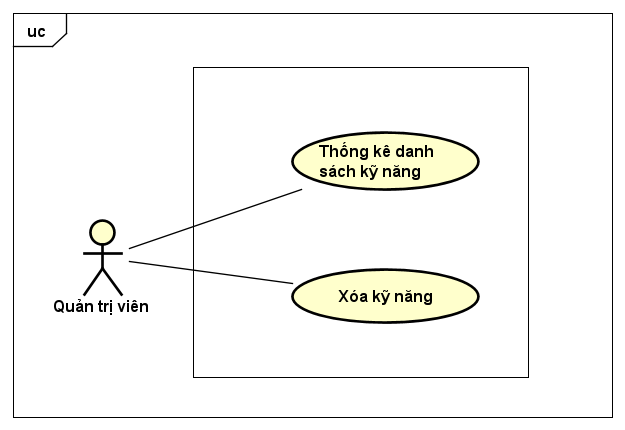
### Biểu đồ use case phân rã “Quản lý tin tuyển dụng”



**Hình 6** Phân rã Usecase “Quản lý tin tuyển dụng”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | 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 |

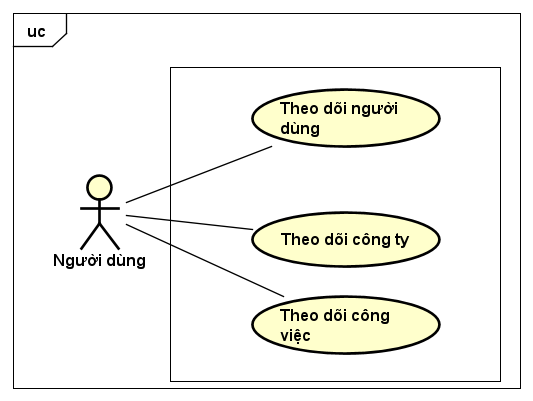
### Biểu đồ use case phân rã “Quản lý kỹ năng”



**Hình 7** Phân rã Usecase “Quản lý kỹ năng”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | 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 |

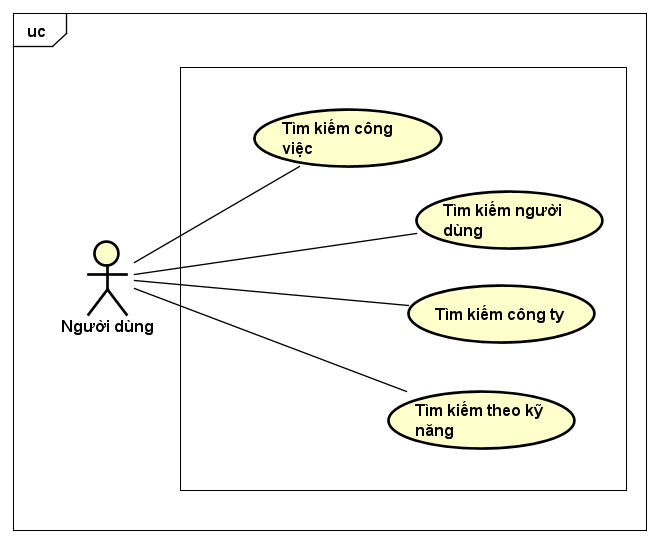
### Biểu đồ use case phân rã “Theo dõi”



**Hình 8** Phân rã Usecase “Theo dõi”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | 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 |

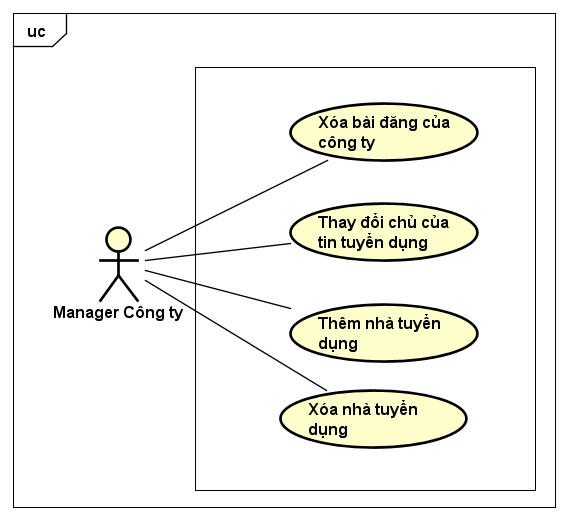
### Biểu đồ use case phân rã “Tìm kiếm”



**Hình 9** Phân rã Usecase “Tìm kiếm”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | 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 |

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



**Hình 10** Phân rã Usecase “Quản lý công ty của tôi”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Usecase | Tên UseCase | 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 |

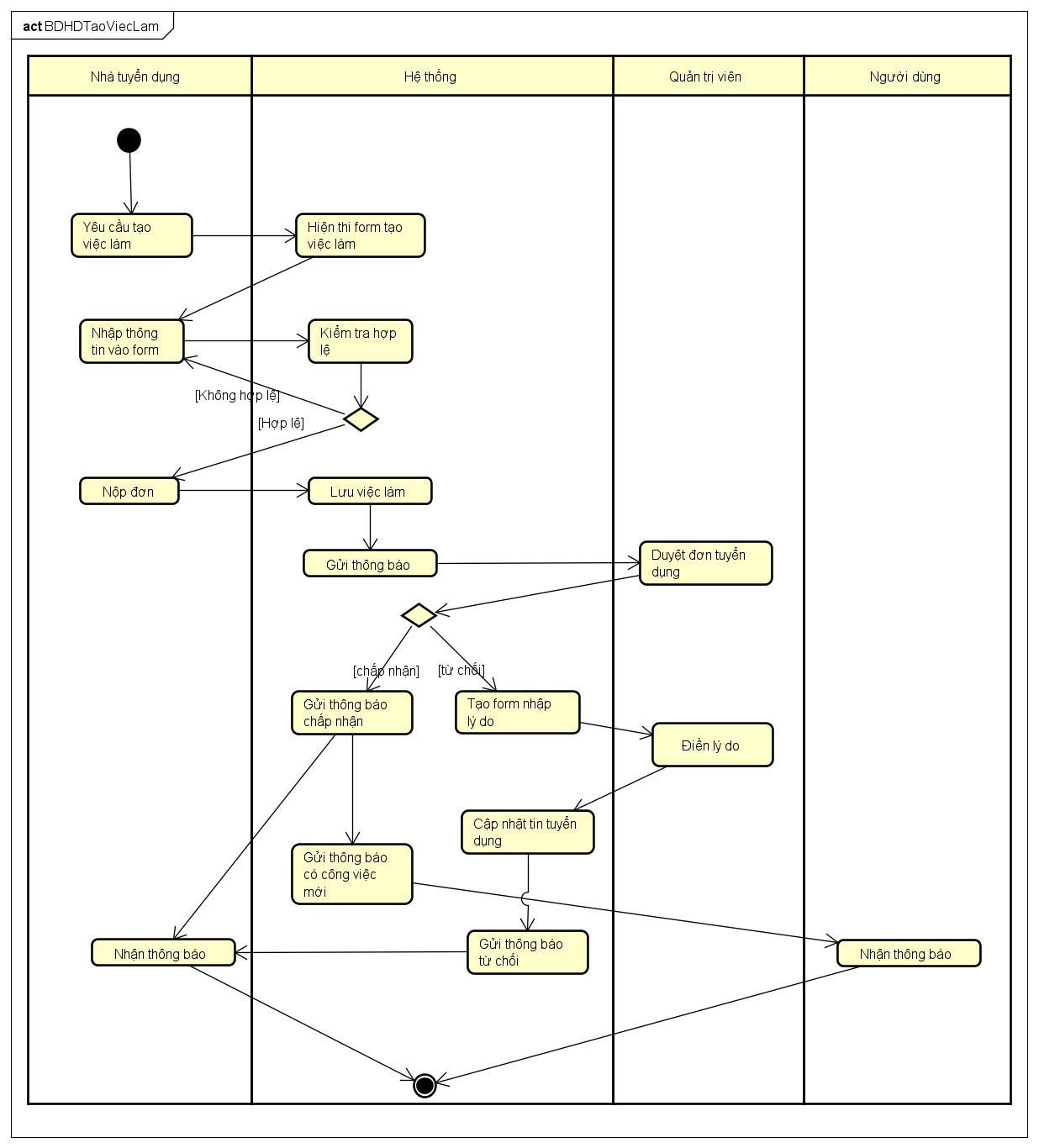
### Quy trình nghiệp vụ

1. 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 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.

|  |  |
| --- | --- |
|  | Đăng tin tuyển dụng |
| **Mục tiêu quy trình** | Đăng tin tuyển dụng việc làm lên hệ thống |
| **Phạm vi quy trình** | Nhà tuyển dụng, quản trị viên, người dùng |
| **Đầu vào quy trình** | * Thông tin tuyển dụng * Thông tin người đăng tin tuyển dụng |
| **Luồng quy trình** | * Nhà tuyển dụng chọn chức năng đăng tin tuyển dụng * Hệ thống hiển thị giao diện đăng tin tuyển dụng * Nhà tuyển dụng nhập các thông tin cần thiết cho việc làm * Hệ thống xác thực đầy đủ thông tin, nếu thiếu, hiện thị lỗi và yêu cầu người dùng sửa lại * 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. * Quản trị viên xác nhận hoặc từ chối tin tuyển dụng sau khi kiểm tra tính đúng đắn của tin * 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 |
| **Đầu ra quy trình** | Gửi thông báo xác nhận hoặc từ chối tin tuyển dụng |
| **Quy trình đặc biệt** | Không có |

**Bảng 1** Quy trình đăng tin tuyển dụng



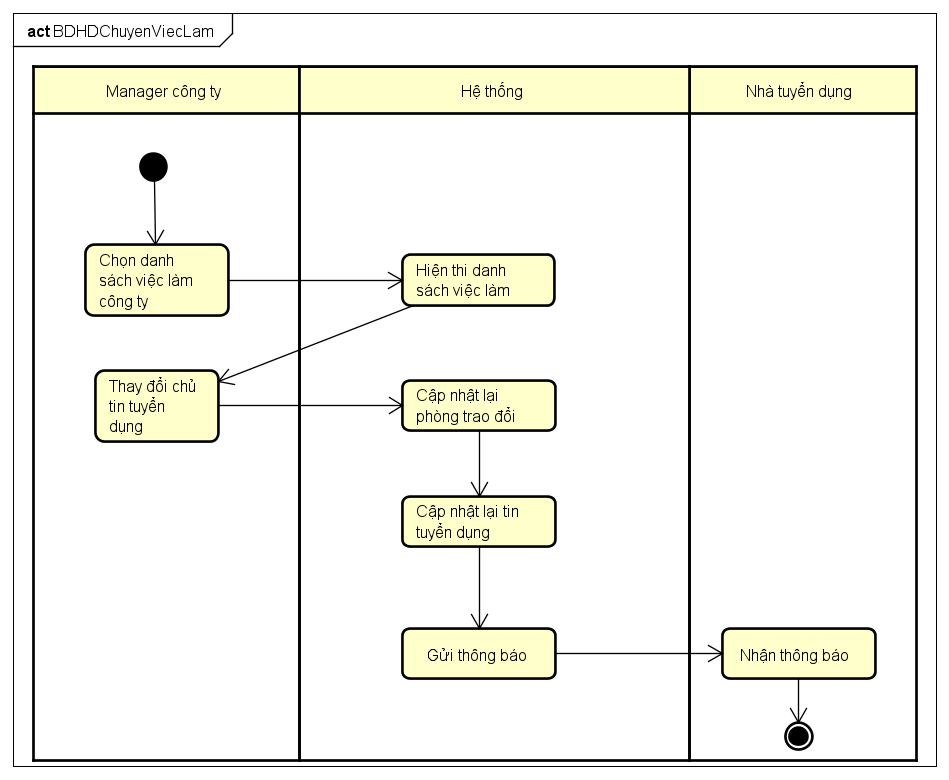
**Hình 11** Biểu đồ hoạt động đăng tin tuyển dụng

1. 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 |
| **Mục tiêu quy trình** | Chuyển công việc từ nhà tuyển dụng này đến nhà tuyển dụng khác thuộc cùng một công ty do manager công ty thực hiện |
| **Phạm vi quy trình** | Nhà tuyển dụng, manager công ty |
| **Đầu vào quy trình** | * Tin tuyển dụng * Nhà tuyển dụng cũ * Nhà tuyển dụng mới |
| **Luồng quy trình** | * Manager công ty chọn xem danh sách công việc của công ty * Hiện thị danh sách việc làm của công ty * Manager công ty thực hiện chuyển giao công việc * Hệ thống cập nhật lại phòng trao đổi của công việc đó * Hệ thống cập nhật lại thông tin chủ của tin tuyển dụng * Hệ thống gửi thông báo đến câc nhà tuyển dụng liên quan |
| **Đầu ra quy trình** | Gửi thông báo thay đổi nhà tuyển dụng thành công |
| **Quy trình đặc biệt** | Không có |

**Bảng 2** Quy trình thay đổi chủ của tin tuyển dụng



**Hình 12** Biều đồ hoạt động thay đổi chủ của tin tuyển dụng

## Đặc tả chức năng

### Đặc tả use case “Quên mật khẩu”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC03 | Tên Use case | Quên mật khẩu |
| Tác nhân | Khách | | |
| Tiền điều kiện | Đã vào trang quên mật khẩu | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 đãn 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 biều 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 | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 3a | Hệ thống | Thông báo lỗi: cần nhập đầy đủ cá 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 | | | |
| Hậu điều kiện | Không có | | |

**Bảng 3** 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@gmai.com](mailto:alice@gmai.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](mailto:alice@gmai.com) |
| 2 | Confirmpassword | Kiểm tra lại mật khẩu | Có | Đúng với trường password | [aclie12312](mailto:alice@gmai.com) |

### Đặc tả use case “Theo dõi người dùng”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC14-1 | Tên Use case | Theo dõi người dùng |
| Tác nhân | Người dùng | | |
| Tiền điều kiện | Hiện ở trang profile của người dùng muốn theo dõi | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1. | Ngưởi 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 | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 | | | |
| Hậu điều kiện | Cập nhật lại Feed của người theo dõi | | |

**Bảng 4** Đặc tả use case “Theo dõi người dùng”

### 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 | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1. | Người dùng | Chọn mục cần cập nhật | | 2. | Hệ thống | Hiện 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 | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 5** Đặc tả use case “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 trị trên hệ thống | Có | Phải là duy nhất trong hệ thống | [phongvu](mailto:alice@gmai.com) |
| 2 | email | Email của người dùng | Có | Phải là duy nhất trong hệ thống | [alice@gmail.com](mailto:alice@gmail.com) |
| 3 | avatar | ảnh đại diện của người dùng | Không | Chuỗi string | http://localhost:3000/ 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: “IT”,  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 | experience | Kinh nhiệm của người dùng | Không | Chuỗi string | Tôi có 5 năm kinh nghiệp về C# |
| 8 | languages | Ngoại ngữ của người dùng | Không | Chuỗi string | English, Viẹtnamese |
| 9 | social | Đường liên kết đến các trang website của người dùng | Không | Object | {youtube: <http://localhost>, facebook: <http://localhost>,} |

### Đặc tả usecase “Đăng bài”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC12 | Tên Use case | Đăng bài |
| Tác nhân | Người dùng | | |
| Tiền điều kiện | Đang thao tác với hệ thống | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 dể tạo đối với bài đăng công ty | | | |
| Hậu điều kiện | Cập nhật lại danh sách bài đăng | | |

**Bảng 6** Đặc tả use case “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 | images | Hình ảnh minh hoạt của bài viết | Không | Đường dẫn đến ảnh | <http://abc.com> |
| 3 | videos | Video của bài viết | Không | Chứa 1 file video | http://abc.com |

### Đặc tả use case “Trao đổi”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC15 | Tên Use case | Trao đổi |
| Tác nhân | Người dùng | | |
| Tiền điều kiện | Đã ở trang profile của người dùng cần trao đổi | | |
| Luồng sự kiện chính  (Thành công) |  | | |
| Luồng sự kiện thay thế |  | | |
| Hậu điều kiện | Không | | |

**Bảng 7** Đặc tả use case “Trao đổi”

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

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC19 | Tên Use case | Ứng tuyển công việc |
| Tác nhân | Người tìm việc | | |
| Tiền điều kiện | Đang ở trang xem chi tiết công việc hoặc tìm kiếm, đề xuất công việc | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 | | | |
| Hậu điều kiện | Cập nhật danh sách đã ứng tuyển | | |

**Bảng 8** Đặc tả use case “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 | [Alize@gmail.com](mailto:Alize@gmail.com) |
| 2 | phone | Số điên thoại hiện tại người tìm việc sử dụng | Có | Định dạng 10 chữ số | [1234567890](http://abc.com) |
| 3 | CV | Đơn ứng tuyển của ứng viên | Có | Đường dẫn | http://abc.com |

### Đặc tả use case “Đăng tin tuyển dụng”

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC20 | Tên Use case | Đăng tin tuyển dụng |
| Tác nhân | Nhà tuyển dụng | | |
| Tiền điều kiện | Đang ở trang chủ của hệ thống | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 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 | | | |
| 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: sai hạn chót, công ty bị khóa… | | | |
| Hậu điều kiện | Không | | |

**Bảng 9** Đặc tả use case “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 | [lạlafjl](mailto:Alize@gmail.com) |
| 2 | salary | Lương của công việc | Có | Định dạng số hoặc khoảng lương | [2000000](http://abc.com) |
| 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 nghiệp đã trải qua | Có | Chuỗi | 1-3 năm kinh nghiệp |
| 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#] |

## Yêu cầu phi chức năng

### 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, các thông tin cá nhân phải được đảm bảo tính riêng tư.

### Yêu cầu vận hành

Hệ thống được xây dựng dựa trên nền tảng ứng dụng website, chính vì vậy, hệ thống cần đẩm 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.

### Yêu cầu giao diện

Hệ thống có giao diện đồ họa, đảm bảo dễ sử dụng, thân thiện với sử dụng. Bố 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 biều, thanh điều hướng chú trọng sử dụng các hình vẽ và những từ ngữ cô đọc, dễ hiều, 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.

# Công nghệ sử dụng

## Ngôn ngữ lập trình Javascript

Javascript viết tắt là JS được bắt đầu phát triền từ năn 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, Javascript giờ đây không những có thể biên dịch và chạy ở browser, mà nó còn có thể chạy được ở phía server-side thông qua trình biên dịch là Node.js.

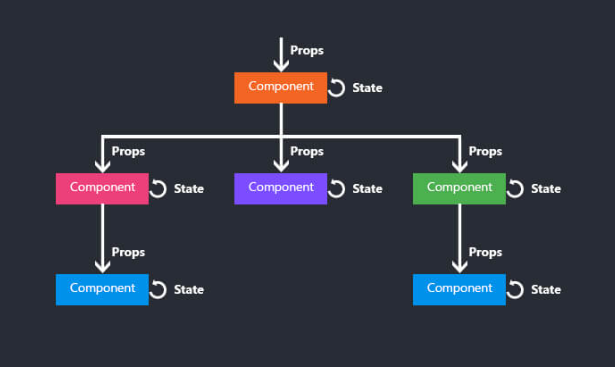
Cho đến ngày nay, JS đã trở thành ngôn ngữ phổ biến, có một cộng đồng phát triển to lớn nhờ các ưu điểm của nó đem lại.JS ở phía client giúp chung ta xử lý nhiều việc như: validation, filter và pagination… Từ đó, phía server sẽ tiết kiệm được lượng lớn lưu lượng khi client giao tiếp với server. JS được đánh giá là ngôn ngữ dữ 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.

## 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. React được phát triển và duy trì bởi công ty Facebook (hiện nay là Meta). 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 hình Model-View-Controller (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 13** Cấu trúc, thành phần của ReactJS

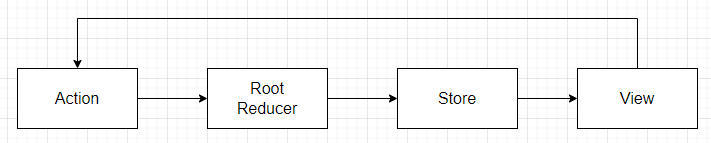
Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến. State: thể hiện trạng thái của ứng dụng, khi state thay đồi thì component đồng thời render lại để cập nhật UI. Về cơ bản, việc xây dựng một ứng dụng MVC phía client với giàng 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 hay Angular.js, tuy nhiên bạn sẽ thấy các hạn chế của chúng khi dự án của bạn ngày càng lớn. 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.

## Công nghệ Redux

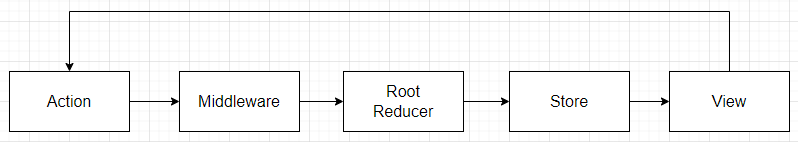
Như đã phân tích ở trên, React xây dựng từ các thành phần riêng rẽ được gọi là component. Các state hoặc props có nhiệm vụ kiểm soát trạng thái của component. Tuy nhiên khi hệ thống ngày càng mở rộng, việc điều phối các state và truyền props xuống các component con trở thành một nhiệm vụ phức tạp. Điều này nêu lên 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, apolloGraphQL, ContextAPI, Pullstate, and RxJs. Tuy nhiên với ứng dụng React, Redux là một thư viện được sử dụng phổ biến.

Cách Redux hoạt động rất đơn giản. 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, action là 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 dung 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. Store giữ trạng thái ứng dụng. Chỉ có một Store trong bất kỳ ứng dụng Redux nào.



**Hình 14** Redux design pattern.

Redux khi thực hiện asynchronous action (hành động bất đồng bộ) nhưng 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 15** 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.

## 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 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 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 nhiệm người sử dụng và tiết kiệm băng thông khi giao tiếp với máy chủ.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | React Query | SWR | Apollo Client | RTK-Query | React Router |
| 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 10** Bảng so sánh các thư viện tìm nạp dữ liệu

## Công nghệ lập trình máy chủ 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. 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 diểm. 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 công 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 độ (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.

Với đồ án tốt nghiệp, em sử dụng NodeJS và ExpressJS là một framework được xây dựng trên nền tảng của NodeJS. ExpressJS cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. ExpressJS hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng. hiết lập các lớp trung gian để trả về các HTTP request. Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.Cho phép trả về các trang HTML dựa vào các tham số.

## Công nghệ JWT

Hệ thống sử dụng JWT 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. JSON Web Token (JWT) là một chuẩn mở (RFC 7519) 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. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA. 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.

## 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... Tuy nhiên, sử dụng các cách này có độ trễ lớn và gây tốn tài nguyên. Bởi vậy, SocketIO 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. 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.

## 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. 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. 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. Một document trong MongoDB có thể hiểu như một bản ghi record trong cơ sở dữ liệu quan hệ. 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 insert, xóa hay update 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.

Trong MongoDB có hai khái niệm cần nắm rõ: Normalization và Denormalization. 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.

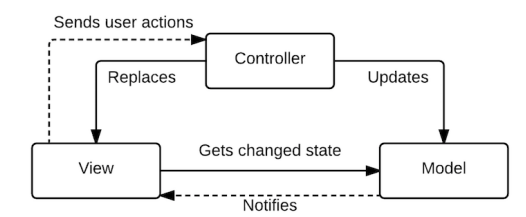
# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

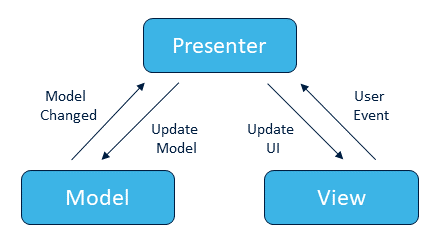
Hệ thống được triển khai trên nền tảng ứng dụng website, dựa trên kiến trúc client - server theo mẫu thiết kế 3 lớp MVP. Đề hiểu MVP là gì, trước hết, ta cần tìm hiểu mẫu thiết kế MVC là gì? Vì MVP được phát triển dựa trên MVC.

MVC từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code 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 16** MVC 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 tránh nhiệm cho việc tiếp nhận tương tác từ View (V) giao tiếp với Model (M) xử lý dữ liệu và trả về kết quả.



**Hình 17** MVP Design Pattern

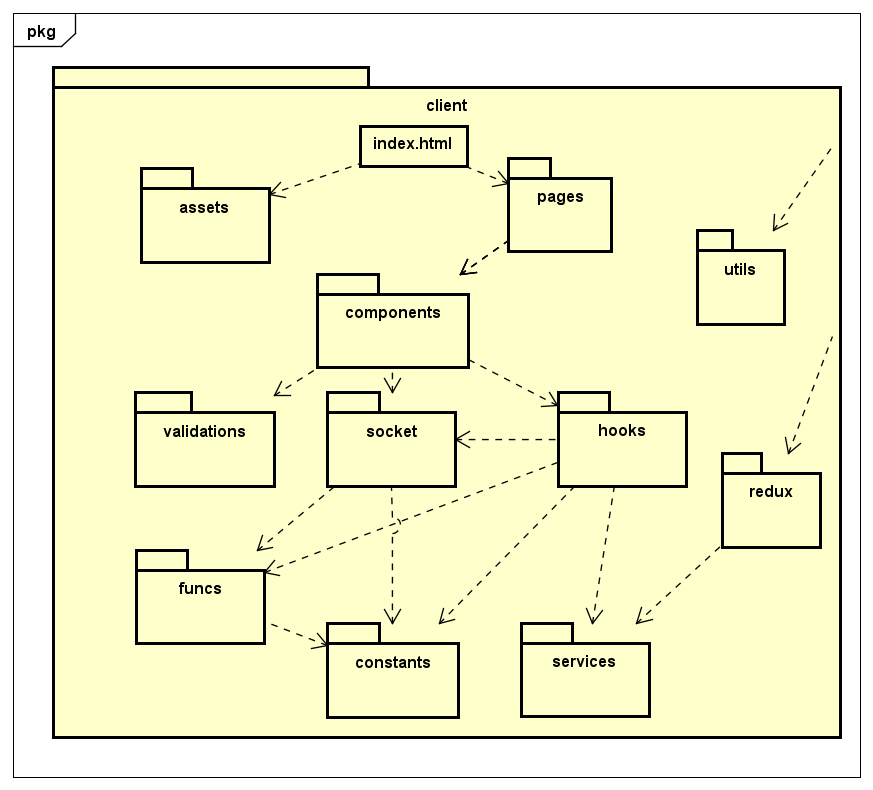
So sánh hai mô hình thiết kế MVP và MVC, ta thấy, MVP có nhiều ưu điểm: View không sử dụng trực tiếp Model trong MVP mà giao tiếp giữa chúng được thực hiện bởi Presenter (Controller trong MVC). Trong MVC, View sẽ đọc dữ liệu trực tiếp từ Model thay vì thông qua Controller. Khi View cần được sửa đổi, 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, và hiệu quả của việc tách và tái sử dụng tốt hơn.

* Tài liệu tham khảo

Hệ thống sử dụng bộ công nghệ MERN (MongoDB, Express, ReactJS, Node.js) được triển khai theo mô hình MVP. Trong đó, phần V (View) sử dụng ReactJS chia ứng dụng thành pages (các trang) được tách riêng đảm bảo tính 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.

### Thiết kế tổng quan

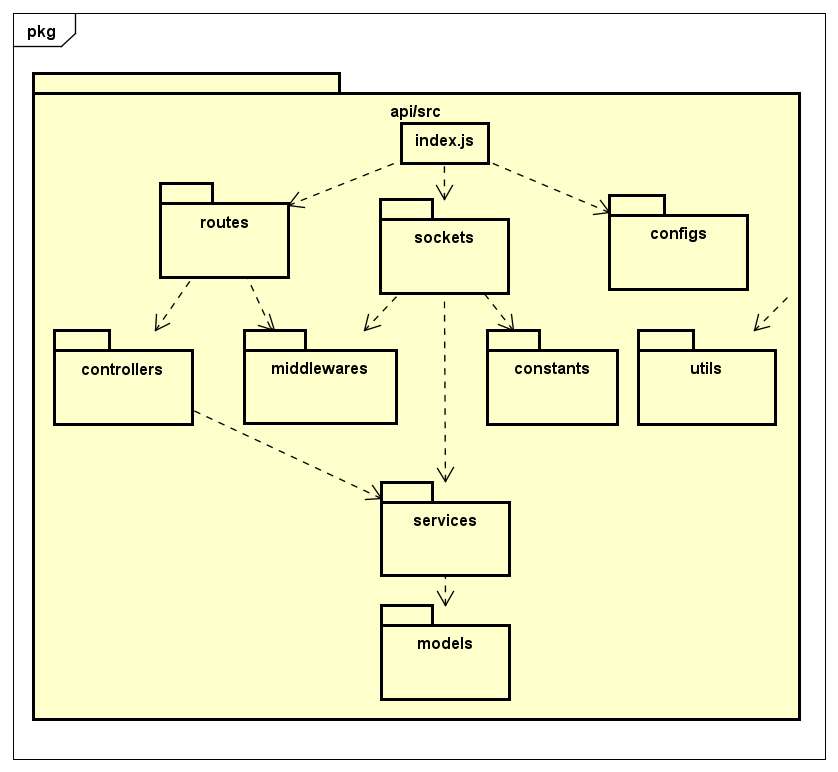
##### Biều đồ phụ thuộc gói client



**Hình 18** Biều đồ phụ thuộc gói Client

|  |  |  |
| --- | --- | --- |
| STT | Gói | Mô tả |
|  | assets | Chứa file logo của hệ thống |
|  | pages | Chứa các trang của hệ thống |
|  | components | Chứa các thành phần của hệ thống có thể tái sử dụng |
|  | validations | Chứa các file xác thực form của hệ thống |
|  | socket | Chứa các file đảm nhận tạo, nhận, sự kiện từ socketIO của hệ thống |
|  | hooks | Chứa file lưu trữ các state của components giúp tái sử dụng, dễ bảo trì hệ thống |
|  | funcs | Xứ lý các dữ liệu từ socketIO nhận được |
|  | constants | Chứa các hằng số của socketIO và hệ thống |
|  | services | Có nhiệm vụ gọi các truyền nhận dữ liệu với server theo giao thức RESTful API |
|  | redux | Chứa các state chung của hệ thống |
|  | utils | Chứa các tiện ích: định dạng ngày, định dạng số, media… được sử dụng trong toàn bộ hệ thống |

##### Biều đồ phụ thuộc gói server

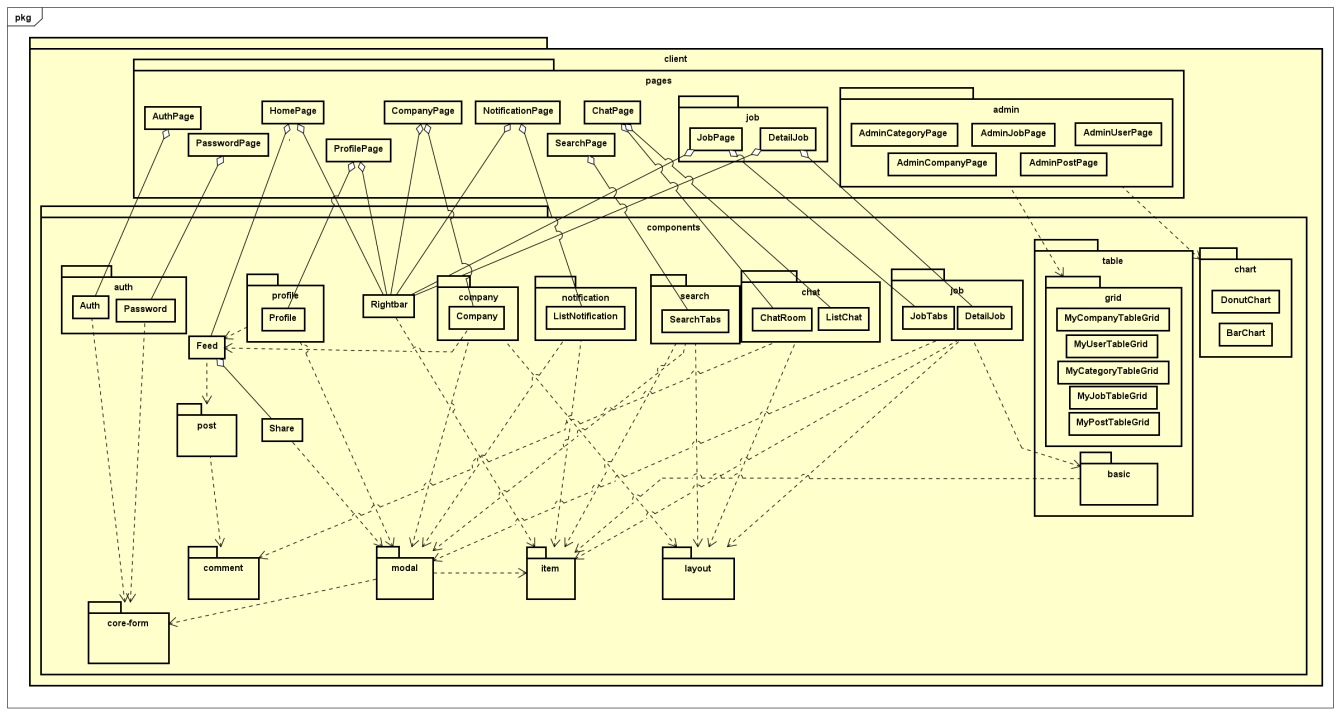


**Hình 19** Tổng quan gói Server

|  |  |  |
| --- | --- | --- |
| STT | Gói | Mô tả |
|  | routes | Xử lý phân định các yêu cầu lời gọi RESTful API đầu vào |
|  | sockets | Tạo, nhận các sự kiện của hệ thống |
|  | configs | Cấu hình của hệ thống |
|  | middlewares | Xác thực tính hợp lệ của lời gọi RESTful API, socketIO, xác định danh tính người dùng |
|  | constants | Chứa các hằng số của socketIO và hệ thống |
|  | controllers | Định dạng dữ liệu trả về, xử lý lỗi |
|  | services | Thực hiện các nghiệp vụ cho hệ thống |
|  | 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. |
|  | utils | Chứa các tiện ích của hệ thống |

### Thiết kế chi tiết gói

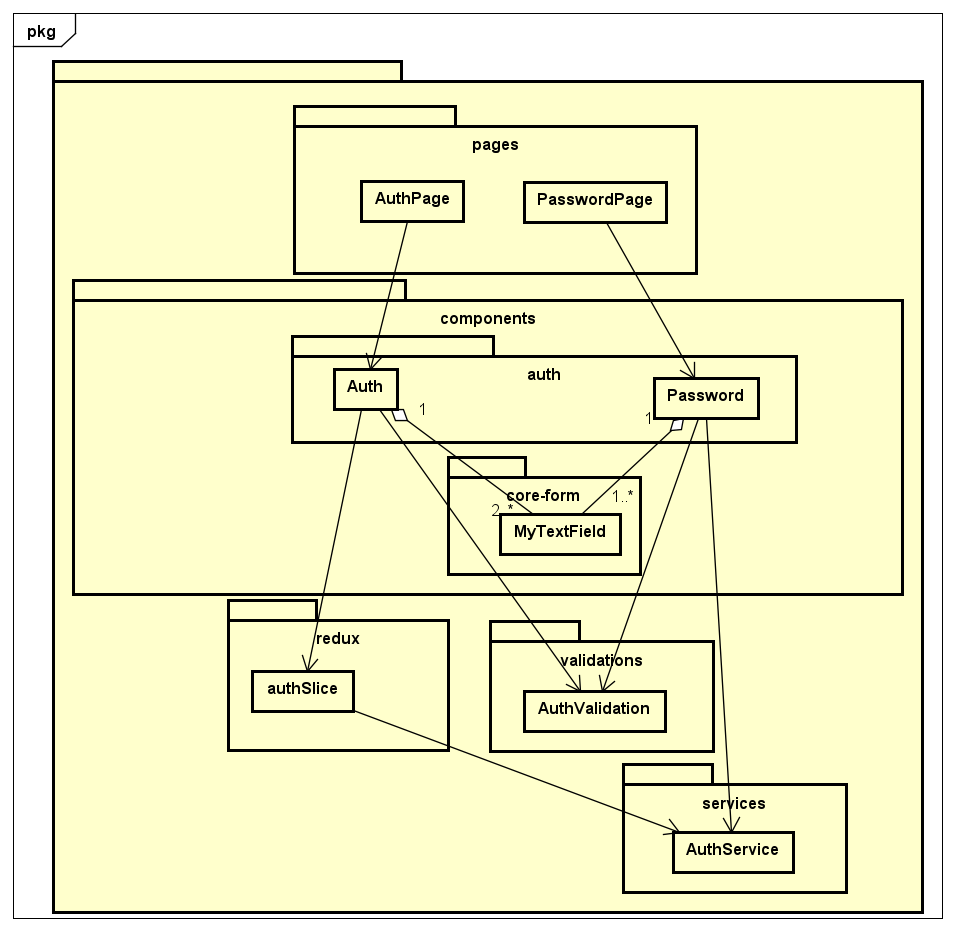
##### Thiết kế chi tiết gói thành phần giao diện



**Hình 20** Thiết kế gói thành phần giao diện

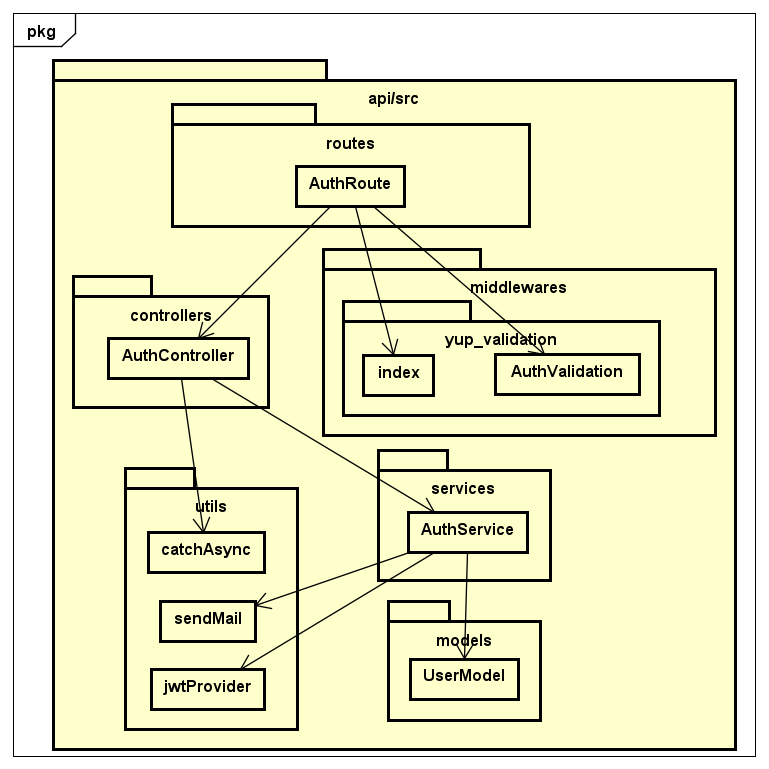
|  |  |  |
| --- | --- | --- |
| STT | Gói/Lớp | Mô tả |
|  | auth | Gói chứa thành phần cho tác nhân khách: đăng ký, đăng nhập, quên mật khẩu |
|  | Feed | Hiện thỉ danh sách bài đăng |
|  | profile | Gói chứa các thành phần chính của hồ sơ: thông tin cá nhân, các bài đăng của họ |
|  | company | Gói chứa các thành phần chính của công ty: hồ sơ công ty, danh sách việc làm… |
|  | Rightbar | Phần được hiện thị bên phải của hệ thống: đề xuất người dùng, công việc, công ty. |
|  | search | Gói để hiện thị giao diện tìm kiếm: người dùng, công ty, công việc. |
|  | table | Gói chứa 2 gói con là: grid (hiện thị bảng cho quản trị viên), basic (hiện thị bảng cho người dùng). |
|  | chart | Gói hiện thị biểu đồ: cột, donut. |
|  | … | … |

##### Thiết kế chi tiết gói auth



**Hình 21** Thiết kế chi tiết gói AuthClient

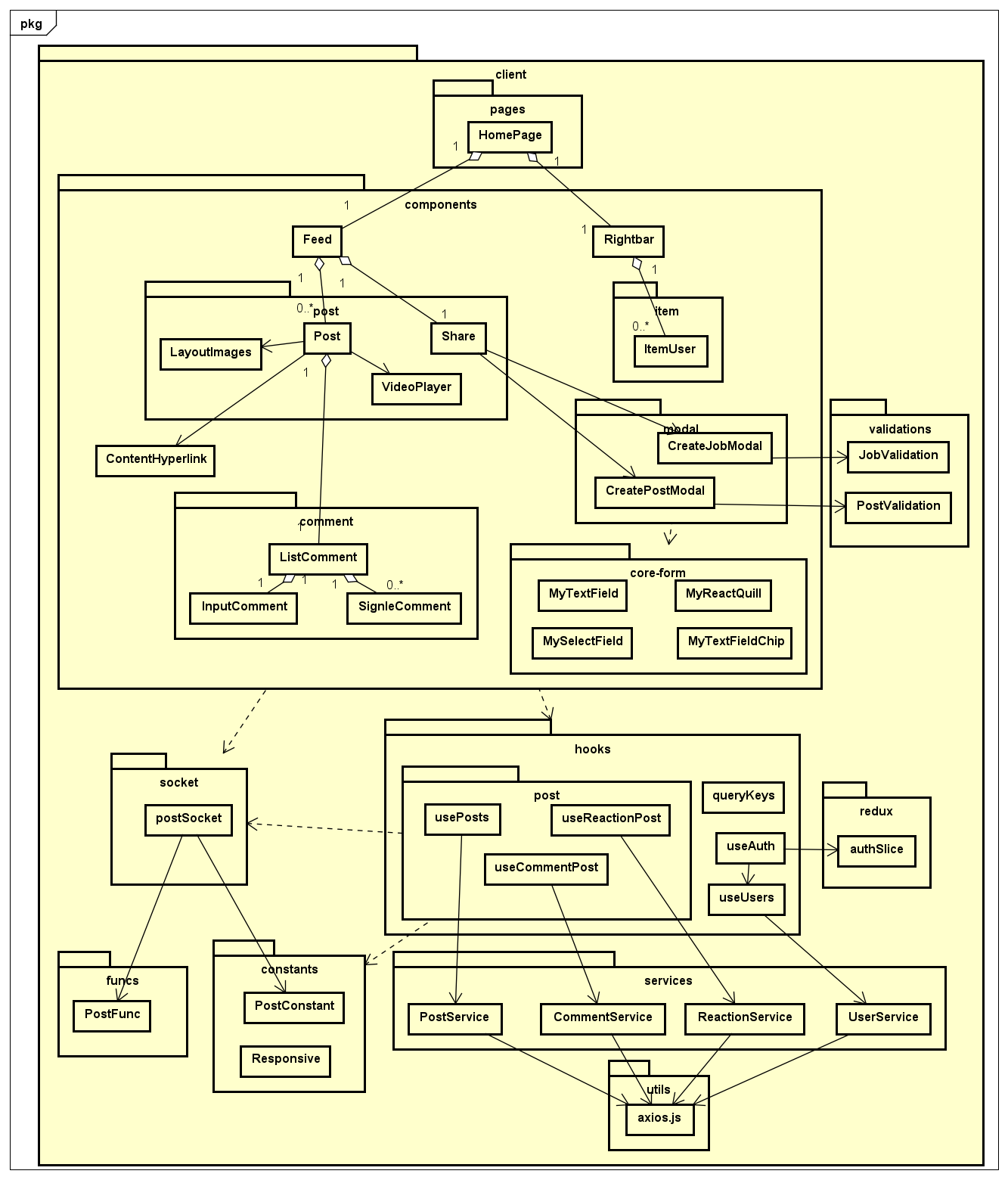
|  |  |  |
| --- | --- | --- |
| STT | Lớp | Mô tả |
|  | AuthPage | Hiện thị trang đăng nhập hoặc đăng ký |
|  | PasswordPage | Hiện thị trang quên mật khẩu hoặc trang thay đổi mật khẩu |
|  | Auth | Tạo ra mẫu đơn đăng nhập hoặc đăng ký sử dụng react-hook-form |
|  | Password | Tạo ra mẫu đơn quên mật khẩu hoặc thay đổi mật khẩu |
|  | MyTextField | Là trường input để điền vào form |
|  | authSlice | Chứa các hành động đăng nhập và đăng ký |
|  | AuthValidation | Chứa các schema giúp xác thực biểu mẫu của form |
|  | AuthService | Giao tiếp nhận dữ liệu từ server |



**Hình 22** Thiết kế chi tiết gói AuthServer

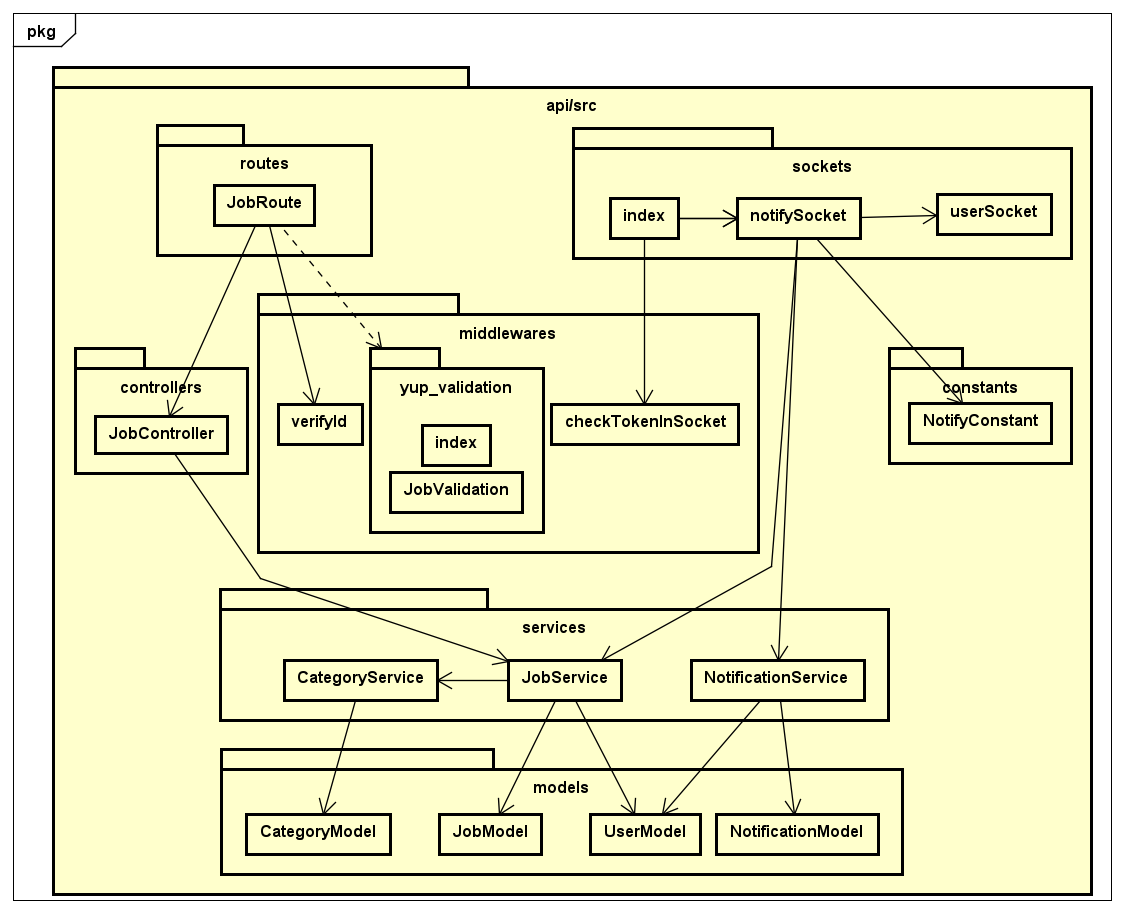
|  |  |  |
| --- | --- | --- |
| STT | Lớp | Mô tả |
|  | AuthRoute | Điều hướng các RESTful API từ client |
|  | index | Xác thực dữ liệu tới từ client: query,params, body |
|  | AuthValidation | Chứa các schema để xác thực dữ liệu từ client |
|  | AuthController | Gọi các nghiệp vụ và định dạng đầu ra của dữ liệu |
|  | catchAsync | Bắt lỗi ngoại lệ |
|  | AuthService | Thực hiện nghiệp vụ của hệ thống: đăng ký, đăng nhập, đổi mật khẩu… |
|  | sendMail | Dịch vụ gửi thư điện tử |
|  | jwtProvider | Ký mã token để xác thực người dùng |
|  | UserModel | Định dạng kiểu mẫu dữ liệu người dùng được lưu trữ trên cơ sở dữ liệu |

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



**Hình 23** Thiết kế màn hình trang chủ

##### Thiết kế gói cho nghiệp vụ việc làm máy chủ



**Hình 24** Thiết kế gói nghiệp vụ việc làm

|  |  |  |
| --- | --- | --- |
| STT | Lớp | Mô tả |
|  | JobRoute | Điều hướng RESTful API request đến từ client, request được thông qua các một loạt các middleware để kiểm tra tính hợp lệ |
|  | index (gói socket) | Khởi tạo không gian tên, middleware cho socket |
|  | notifySocket | Truyền, nhận các sự kiện thông báo: followUser, followCompany, applyJob… |
|  | userSocket | Chứa danh sách người dùng hiện thời trong hệ thống với không gian tên: namespace(“/”) |
|  | verifyId | Xác thực danh tính người dùng dựa vào mã JWT |
|  | index (gói yup\_validation) | Xác thực dữ liệu được truyền từ client theo giao thức RESTful API dựa vào các schema |
|  | JobValidation | Chứa các schema xác thực |
|  | checkTokenInSocket | Kiểm tra tính hợp lệ mã JWT trong socket |
|  | JobController | Định đạng trạng thái,dữ liệu trả về cho client |
|  | NotifyConstant | Chứa hằng số cho các sự kiện thông báo |
|  | CategoryService | Chứa các nghiệp vụ liên quan đến kỹ năng |
|  | JobService | Chứa nghiệp vụ liên quan đến tin tuyển dụng (công việc) |
|  | NotificationService | Chứa nghiệp vụ liên quan đến thông báo |
|  | CategoryModel | Định nghĩa schema, chứa các phương thực thao tác với bảng Category |
|  | UserModel | Định nghĩa schema, chứa các phương thực thao tác với bảng User |
|  | JobModel | Định nghĩa schema, chứa các phương thực thao tác với bảng Job |
|  | NotificationModel | Định nghĩa schema, chứa các phương thực thao tác với bảng Notification |

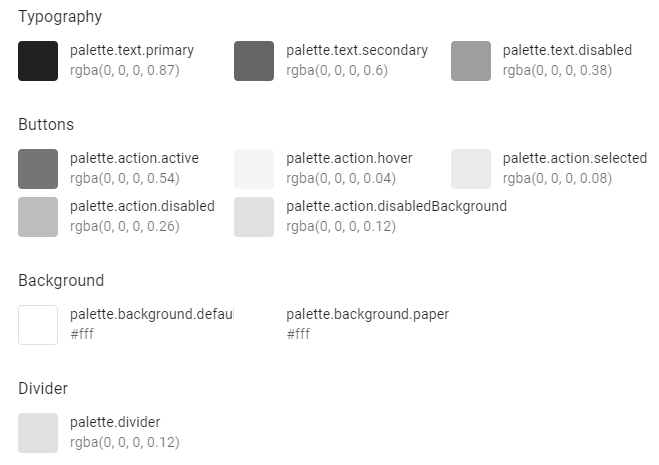
## Thiết kế chi tiết

### Thiết kế giao diện

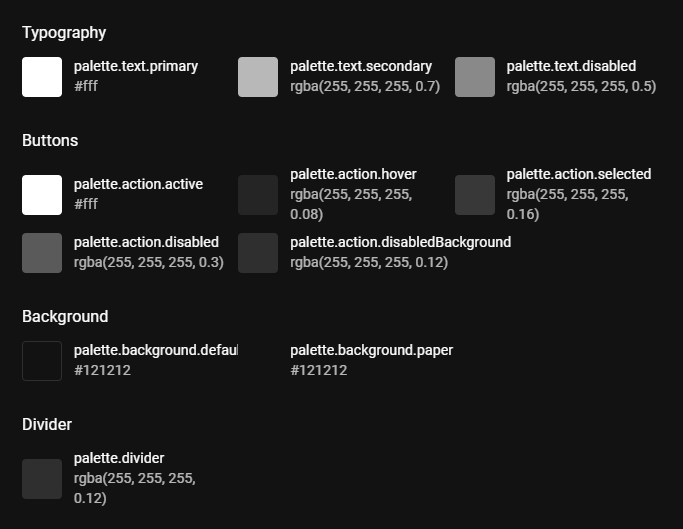
Thiết kế giao diện của người dùng phải đảm bảo tuân thủ các nguyên tắc thiết kế, đáp ứng đầy đủ yêu cầu người dùng. Hệ thống phải đảm bảo đơn giản hóa các tác vụ cho người sử dụng, trình bày bố cục thuận tiện rõ ràng nhất cho người sử dụng dễ dàng hiểu và thao tác

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 25** Bảng màu với giao diện sáng

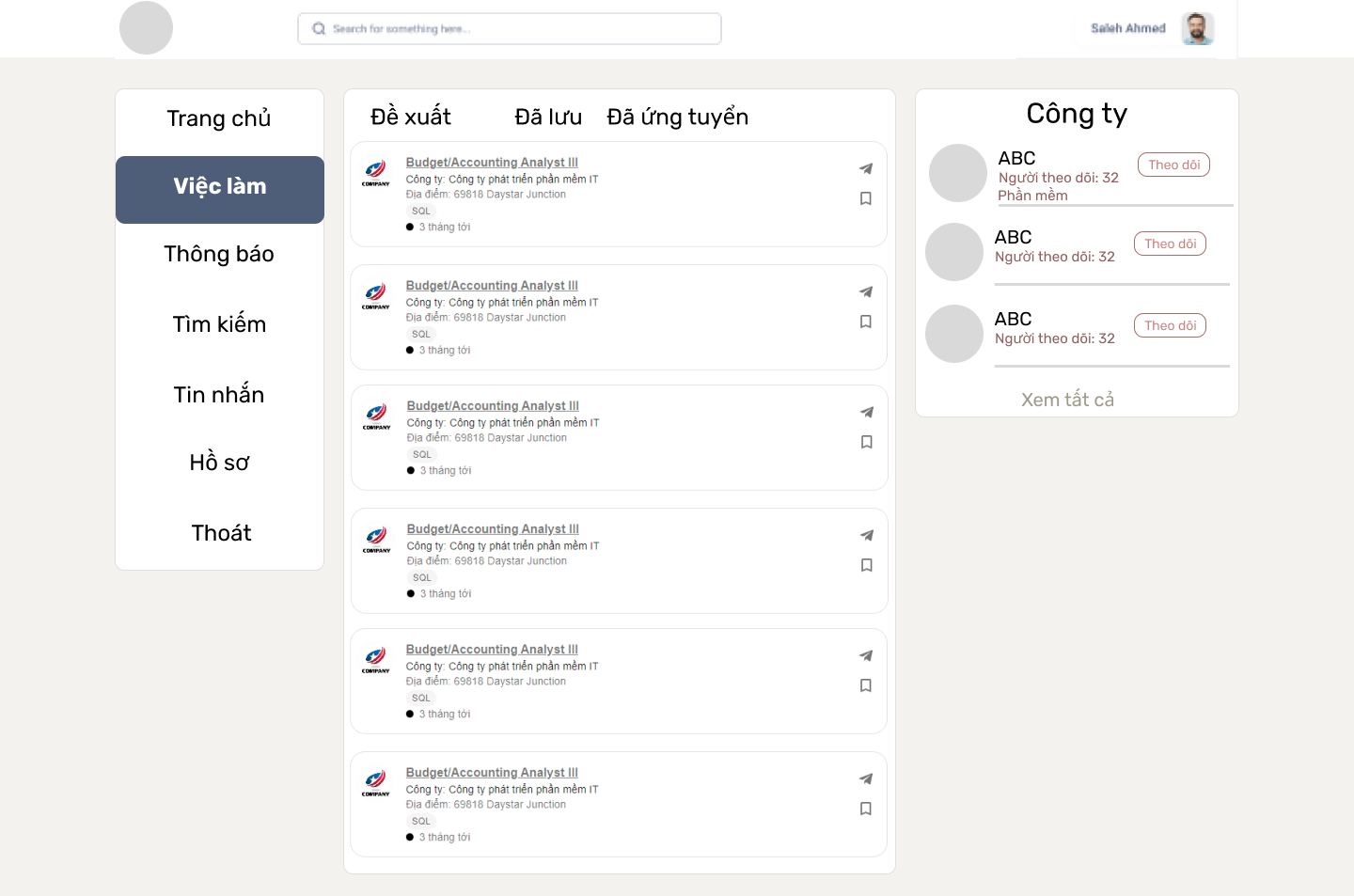


**Hình 26** Bảng màu với giao diện sáng

Một số giao diện minh họa cho chức năng của hệ thống

##### 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.

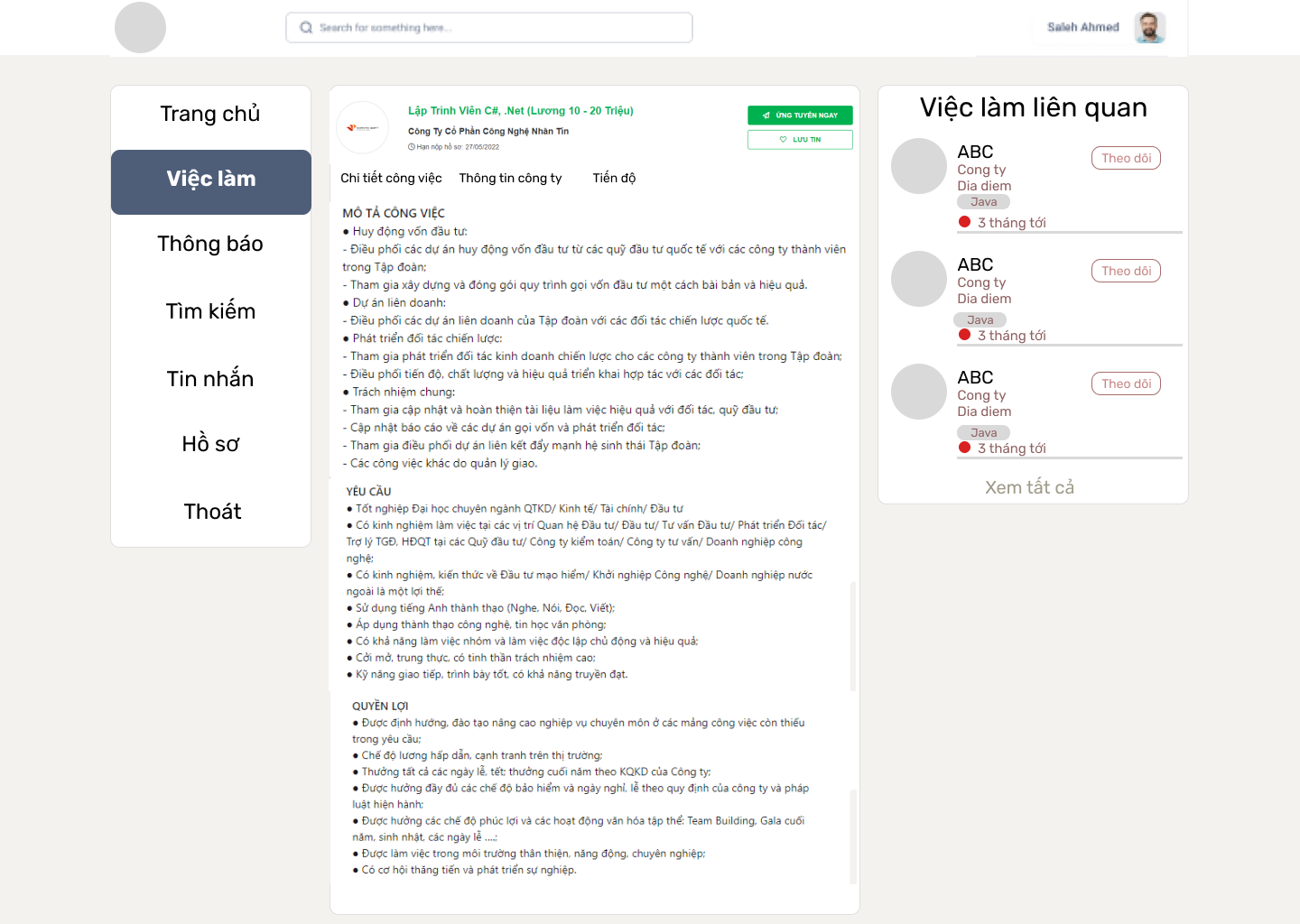


**Hình 27** Thiết kế giao diện trang việc làm

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

Trang chỉ 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 28** Thiết kế trang chi tiết công việc

### Thiết kế lớp

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

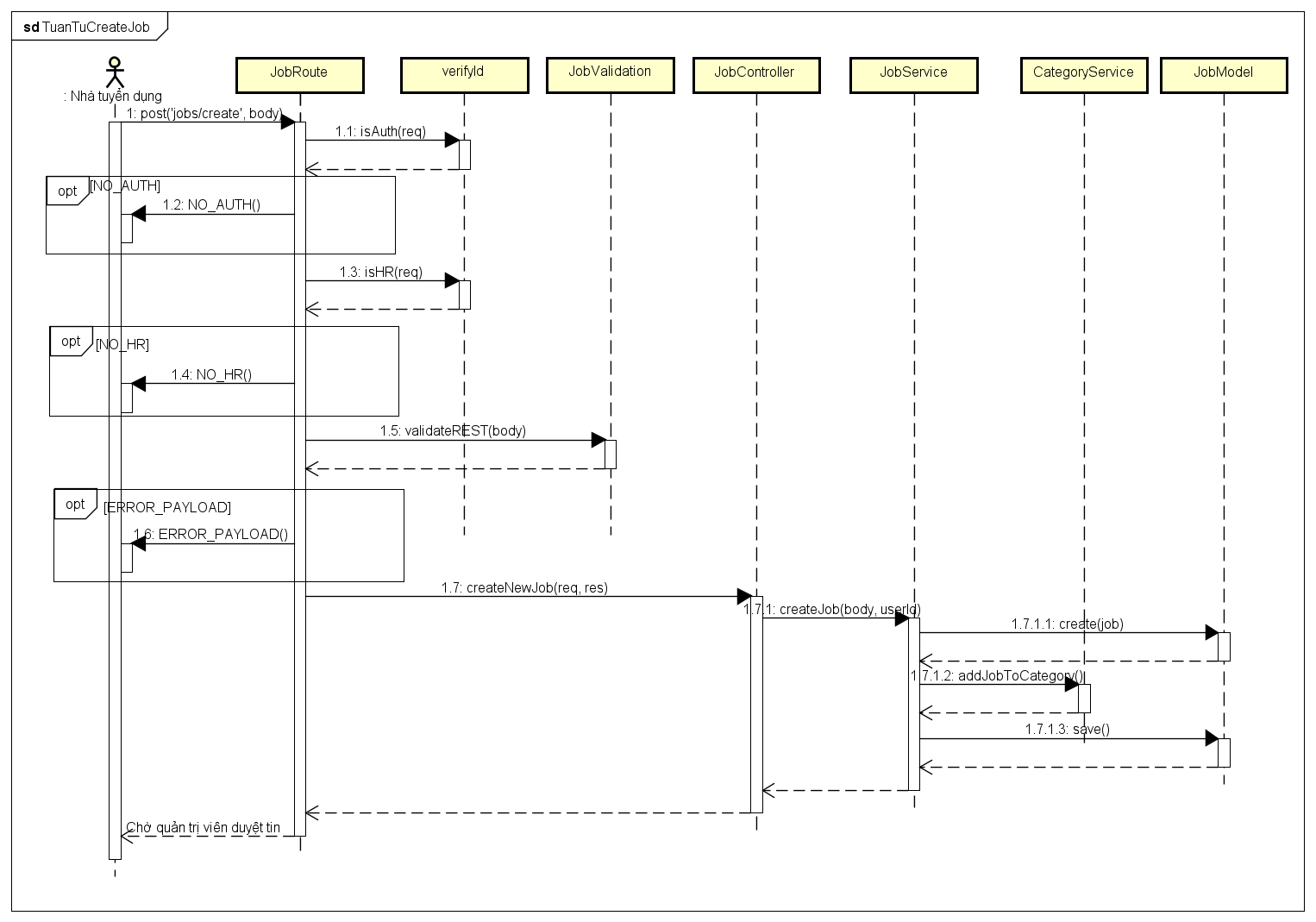
Lớp bao gồm các phương thức và thuộc tính sau đây:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
|  | Category | Category | Một đối tượng của CategoryModel chứa phương thức thao tác với bảng Category |
|  | Company | Company | Chứa các phương thức thao tác với bảng Company |
|  | Job | Job | Chứa các phương thức thao tác với bảng Job |
|  | User | User | Chứa các phương thức thao tác với bảng User |
|  | convertToSlug | convertToSlug | Chuyển chuỗi thành chuỗi slug |
|  | CategoryService | CategoryService | Nghiệp vụ của category |
|  | RoomService | RoomService | Nghiệp vụ của room |

**Bảng 11** Mô tả thuộc tính của lớp JobService

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Mô tả |
|  | createJob | Object | Tạo mới công việc |
|  | updateActiveJobs | Object | Cập nhật trạng thái của công việc theo companyId |
|  | updateApplicantJob | Object | Người tìm kiếm việc làm ứng tuyển công việc |
|  | deleteApplicantJob | Object | Chủ sở hữu tin tuyển dụng xóa ứng viên |
|  | updateJob | Object | Nhà tuyển dụng cập nhật công việc hoặc quản trị viên thêm lý do từ chối công việc |
|  | confirmJob | Object | Quàn trị viên từ chối hoặc chấp nhận công việc |
|  | getJobByCategoryService | Object | Tìm công việc theo tên kỹ năng |
|  | listJobRelateCurrentJob | Object | Tìm các công việc liên quan theo mã công việc |
|  | infoJob | Object | Hiện 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. |
|  | recommendJobByUser | Object | Gợi ý các công việc dựa vào hồ sơ kỹ năng của người dùng |
|  | viewApplyJobService | Object | Xem đơn ứng tuyển |
|  | listJobByCompanyService | Object | Danh sách công việc của công ty |
|  | listMyJobHR | Object | Danh sách công việc của nhà tuyển dụng |
|  | getJobStatsService | Object | Thống kê danh sách việc làm trong năm theo tháng |
|  | getJobStatusService | Object | Thống kê trạng thái của công việc |
|  | changeOwnerJob | Object | Nhà quản lý công ty thay đổi quyền sở hữu của tin tuyển dụng |

**Bảng 12** Mô tả phương thức của lớp JobService



**Hình 29** Biểu đồ tuần tự createJob

##### Lớp PostService chứa các nghiệp vụ lên quan đến bài đăng của hệ thống

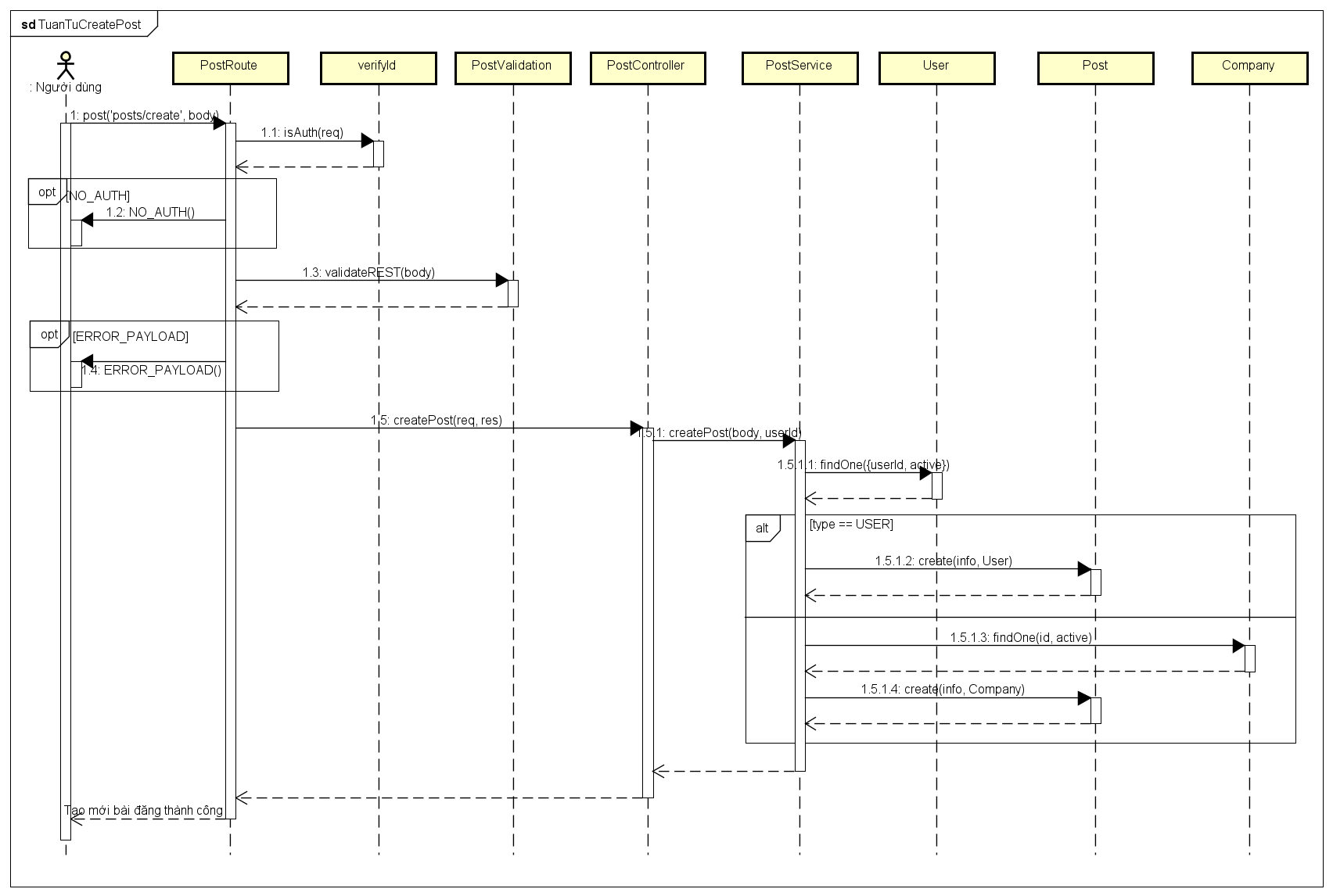
Lớp bao gồm các phương thức và thuộc tính sau đây:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Mô tả |
|  | Post | Post | Là một đối tượng của PostModel chứa phương thức giao tiếp với bảng Post |
|  | Company | Company | Chứa các phương thức thao tác với bảng Company |
|  | Reaction | Reaction | Chứa các phương thức thao tác với bảng Reaction |
|  | Comment | Comment | Chứa các phương thức thao tác với bảng Comment |
|  | User | User | Chứa các phương thức thao tác với bảng User |

**Bảng 13** Mô tả thuộc tính của lớp PostService

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

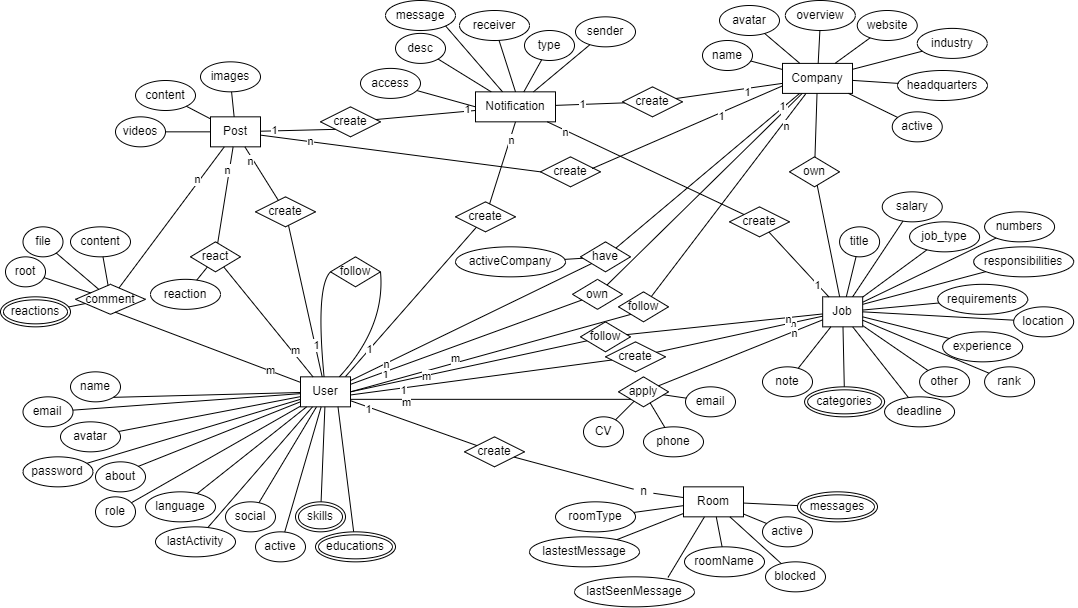
**Bảng 14** Mô tả phương thức của lớp PostService



**Hình 30** Biểu đồ tuần tự createPost

### Thiết kế cơ sở dữ liệu

##### Xây dụng biểu đồ thực thể liên kết



**Hình 31** Biều 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 |
| language | String |  | 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 |

**Bảng 15** 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 |
| images | Array |  | Danh sách hình ảnh cho bài đăng |
| videos | Array |  | Danh sách videos cho bài đăng |

**Bảng 16** 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 17** 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: fulltime, 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 | Ví 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… |

**Bảng 18** Thiết kế thực thể Job

Thực thể Room: lưu trữ phòng trao đổi giữa người dùng với người dùng hoặc người dùng với nhà tuyển dụng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| roomName | String |  | Tên của phòng trao đổi |
| roomType | String | 1 trong 2 giá trị: Personal, Company | Kiểu của phòng trao đổi |
| lastestMessage | Object |  | Tin nhắn mới nhất |
| lastSeenMessage | Object |  | Tin nhắn xem gần nhất |
| messages | Array |  | Danh sách tin nhắn |
| blocked | Boolean |  | Người khóa phòng |
| active | Boolean |  | Trạng thái của phòng |

**Bảng 19** Thiết kế thực thể Room

Thực thể Notification: lưu trữ thông báo của các thực thể khác

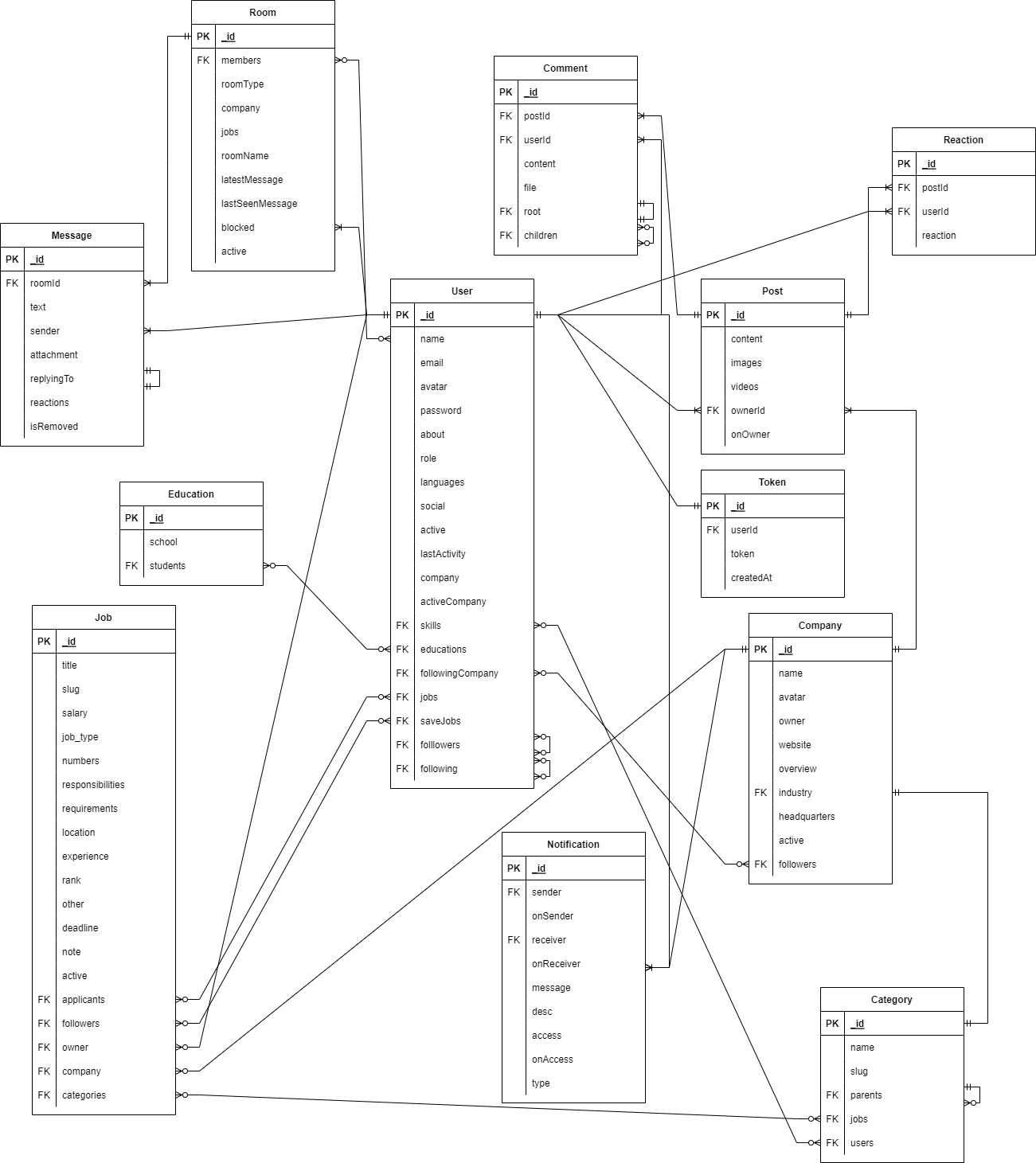
|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| sender | ObjectId |  | Mã người gửi |
| receiver | Array |  | Danh sách người nhận |
| message | String |  | Thông điệp |
| desc | String |  | Chi tiết thông điệp |
| access | ObjectId |  | Đối tượng truy cập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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thực thể nguồn | Thực thể dích | 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 | 1-n | Tạo phòn 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 20** Bảng thiết kế liên kết giữa các thực thể

##### Xây dựng biểu đồ cơ sở dữ liệu phi quan hệ



**Hình 32** Biểu đồ cơ sở dữ liệu phi quan hệ

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

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

**Bảng 21** Danh sách thư viện và công cụ sử dụng

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

Phần thứ nhât, 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 quản 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. 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 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. Hệ thống được đảm bảo cung cấp chính xác các thông tin việc làm thông qua việc kiểm duyệt công việc của quản trị viên…

Các thông số của hệ thống:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ngôn ngữ | Số lượng tệp | Số dòng trắng | Chú thích | Mã nguồn |
| Javascript | 84 | 808 | 336 | 9322 |
| Tổng | 84 | 808 | 336 | 9322 |

Bảng 22 Thống kê gói src của server

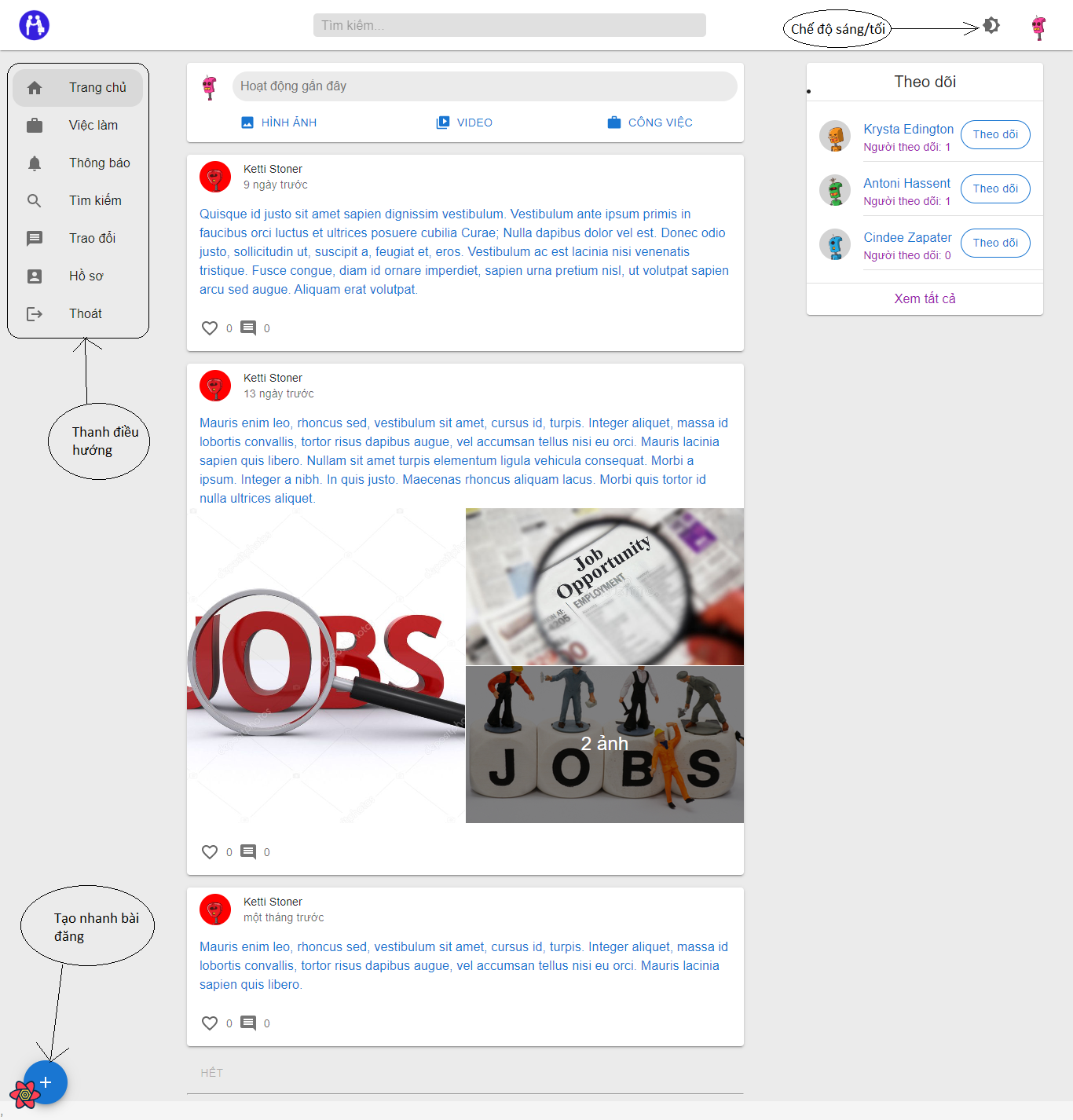
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ngôn ngữ | Số lượng tệp | Số dòng trắng | Chú thích | Mã nguồn |
| JSX | 121 | 544 | 258 | 11210 |
| Javascript | 90 | 435 | 239 | 10811 |
| CSS | 1 | 12 | 4 | 43 |
| SVG | 2 | 0 | 0 | 22 |
| Tổng | 214 | 991 | 501 | 22086 |

**Bảng 23** Thống kê gói src của client

### Minh hoạ các chức năng chính

##### Giao diện trang chủ người dùng

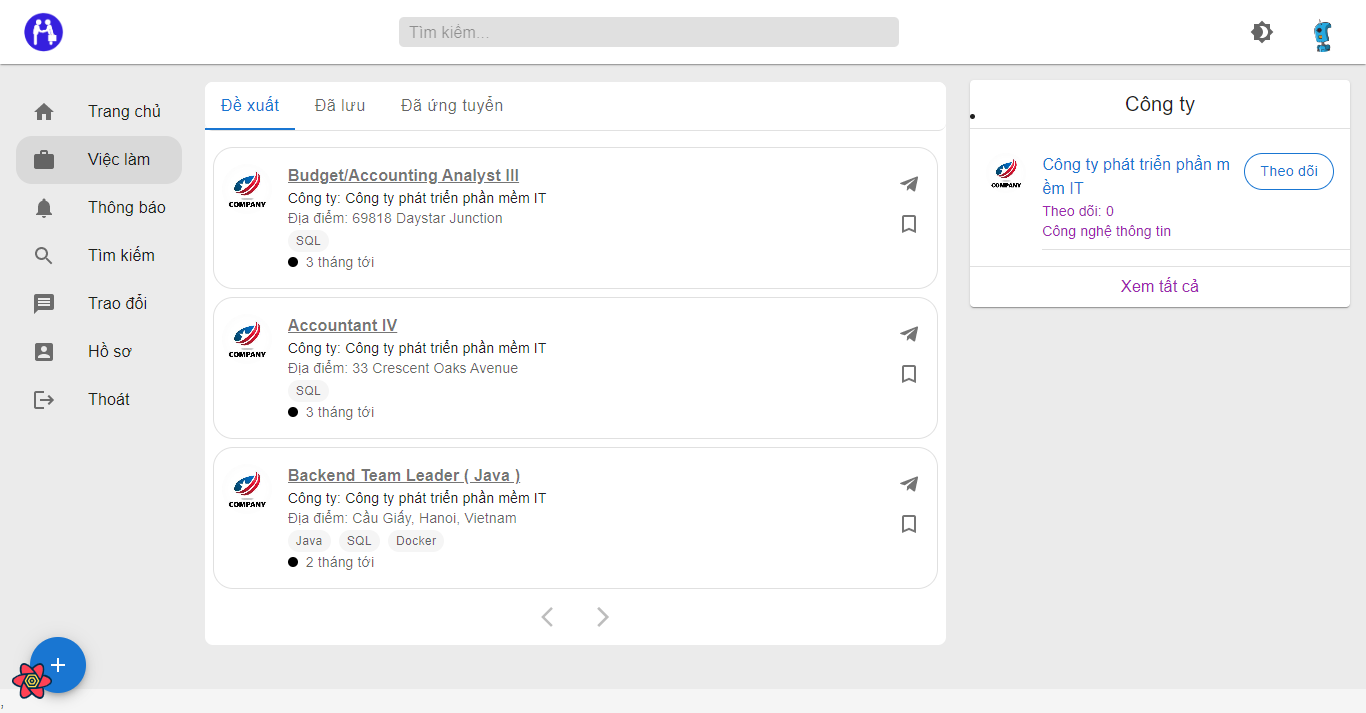
Giao diện trang chủ cho phép người dùng có thể xem danh sách các bài đăng của người dùng mình theo dõi hoặc công ty mình theo dõi. Từ đó, người dùng có thể nắm bắt các hoạt động của họ. Đồng thời, người dùng có thể tương tác, bình luận với bài đăng.



**Hình 33** Giao diện trang chủ người dùng

##### 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.

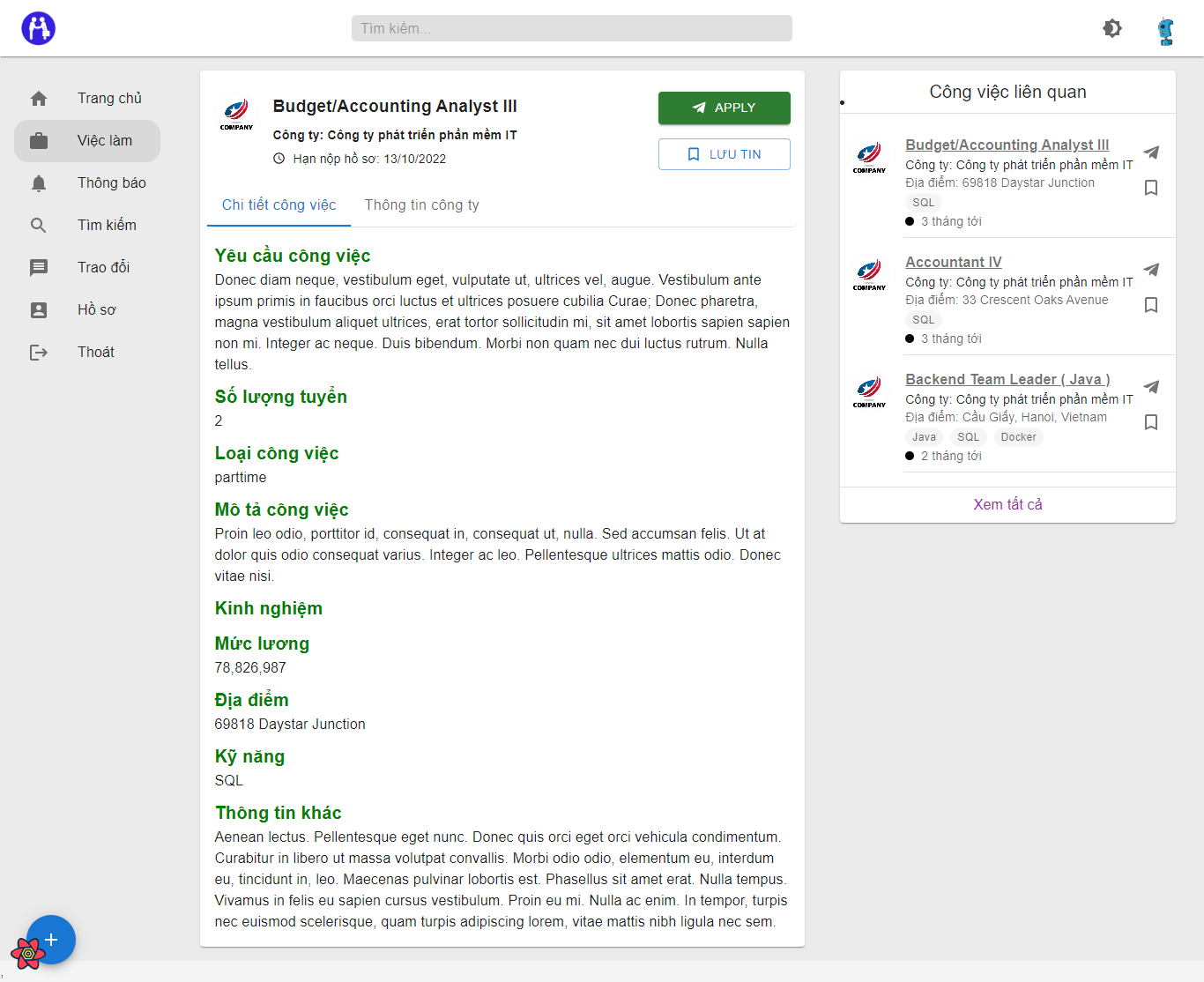


**Hình 34** Giao diện trang việc làm

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

Trang chỉ 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.

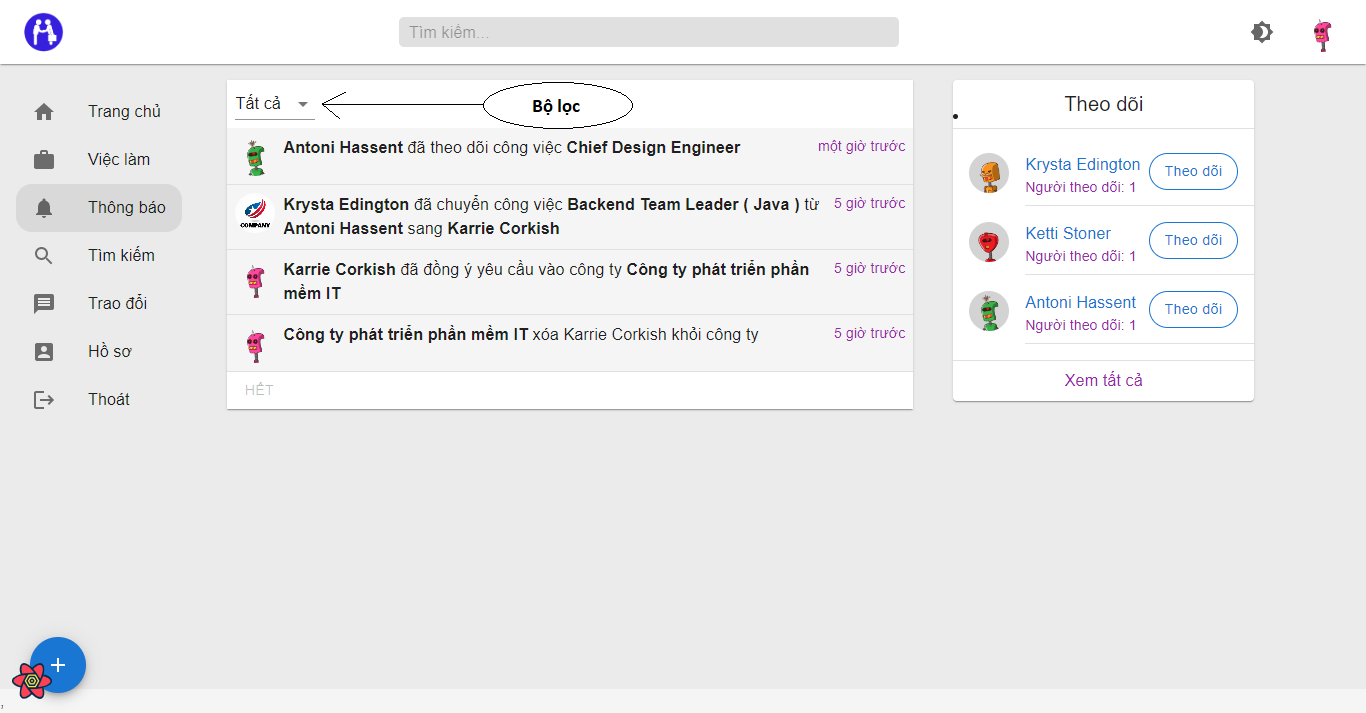
Ngoài ra, giao diện trang ở phía bên phải của màn hình cung cấp các việc làm có liên quan dựa trên danh sách kỹ năng của việc làm hiện tại đang được xem.



**Hình 35** Giao diện trang việc làm

##### Giao diện trang thông báo

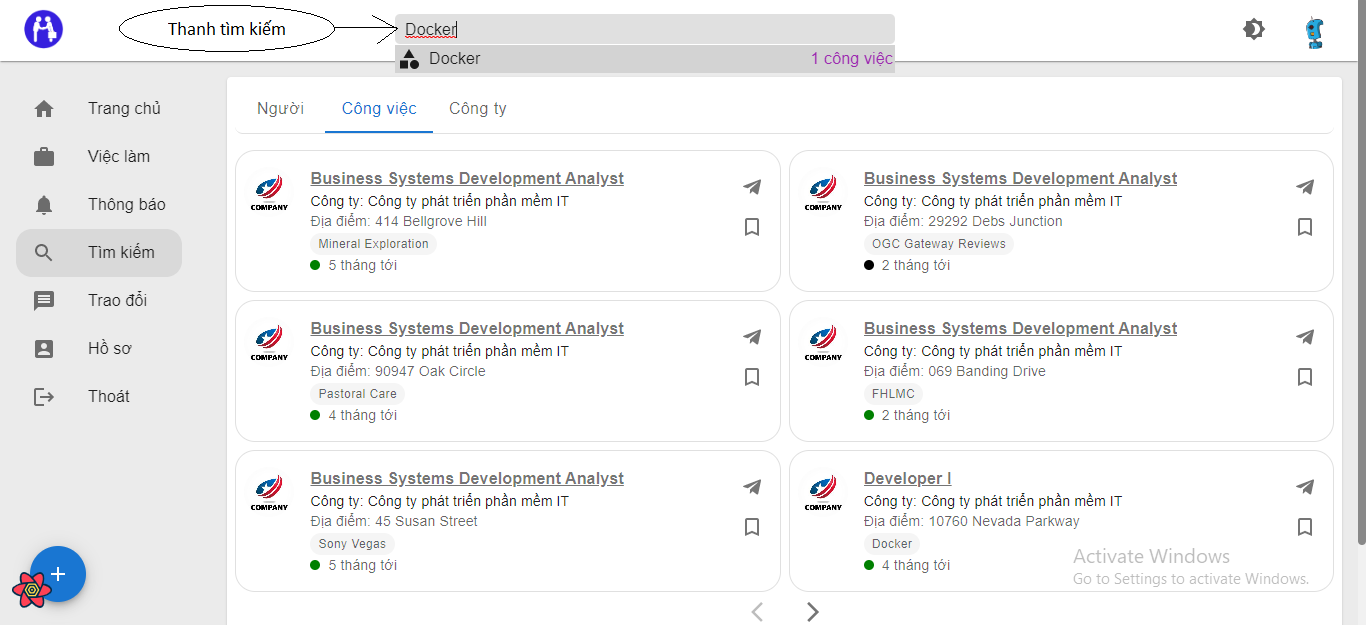
Trang thông báo là nơi mà tác nhân được cung cấp những thông tin, sự kiên nhanh nhất đến mình. Hệ thống cung cấp các sự kiện dựa vào quyền của người sử dụng có trong hệ thống. Ví dụ: đối với quản trị viên các sự kiện có thể nhận là: yêu cầu tạo công ty, yêu cầu tạo việc làm, danh sách đã duyệt việc làm… 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 36** Giao diện trang thông báo

##### Giao diện trang tìm kiếm

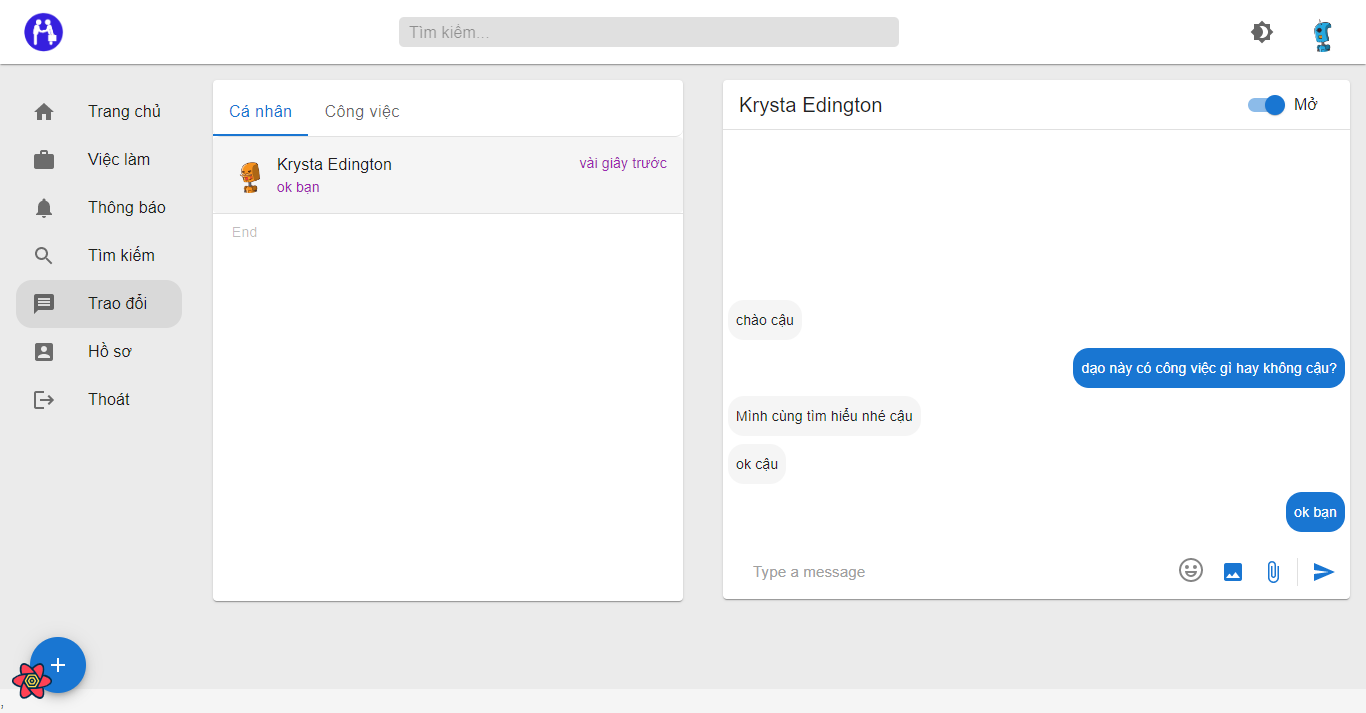
Trang tìm kiếm cung cấp cho người dùng tìm kiếm 3 loại thực thể: người dùng, công việc (tin tuyển dụng) và công ty. Hệ thống tìm kiếm đưa ra kết quả dựa trên truy vấn thông qua thanh tìm kiếm.



**Hình 37** Giao diện tìm kiếm

##### Giao diện trang trao đổi

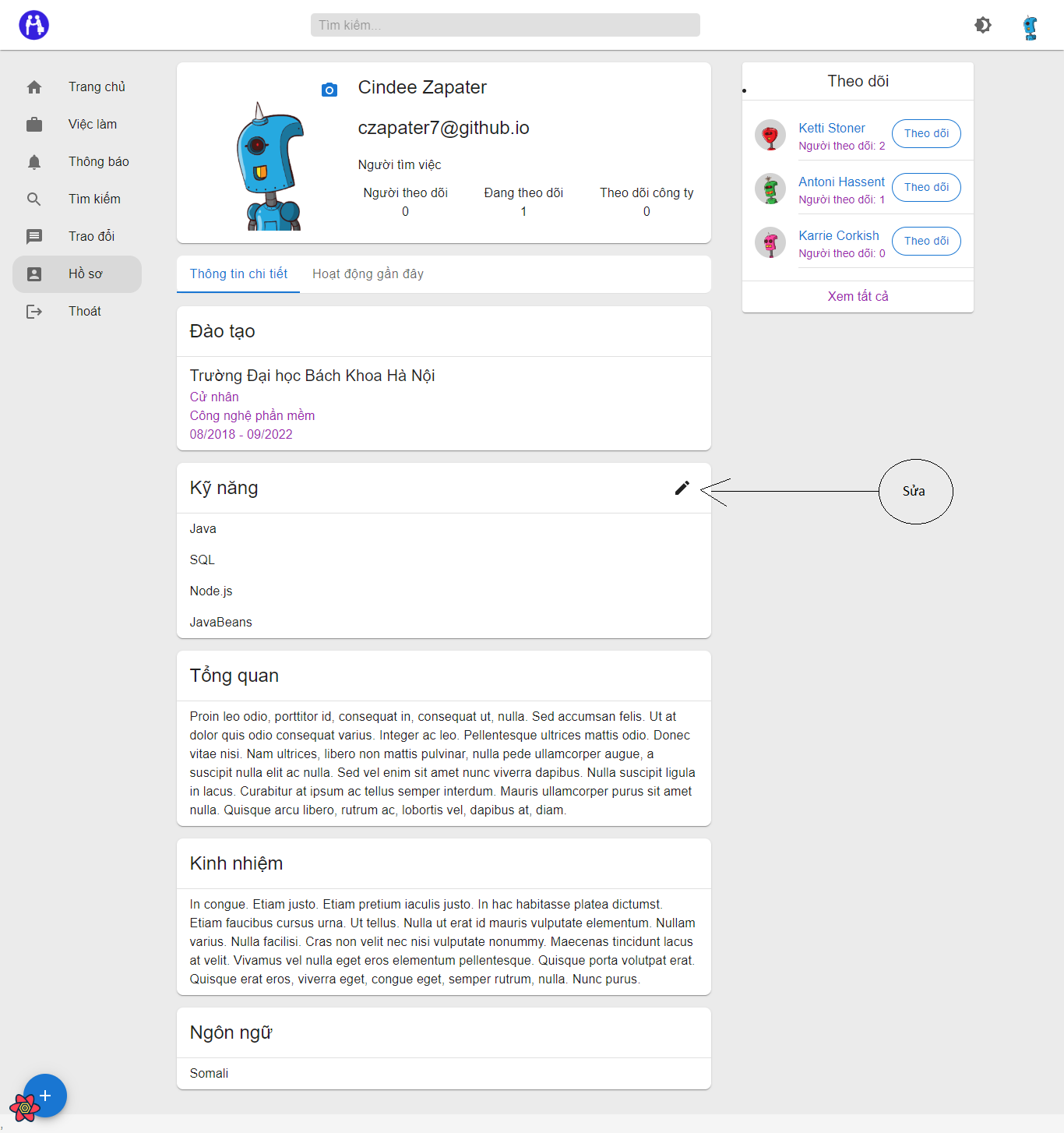
Có 2 loại trao đổi là trao đổi giữa người dùng với nhau hoặc trao đổi giữa người tìm việc và nhà tuyển dụng. Trong khi trao đổi người dùng có thể gửi ảnh, tệp, emoji… nhằm giúp cho cuộc trao đổi dễ hình dung.



**Hình 38** Giao diện trang trao đổi

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

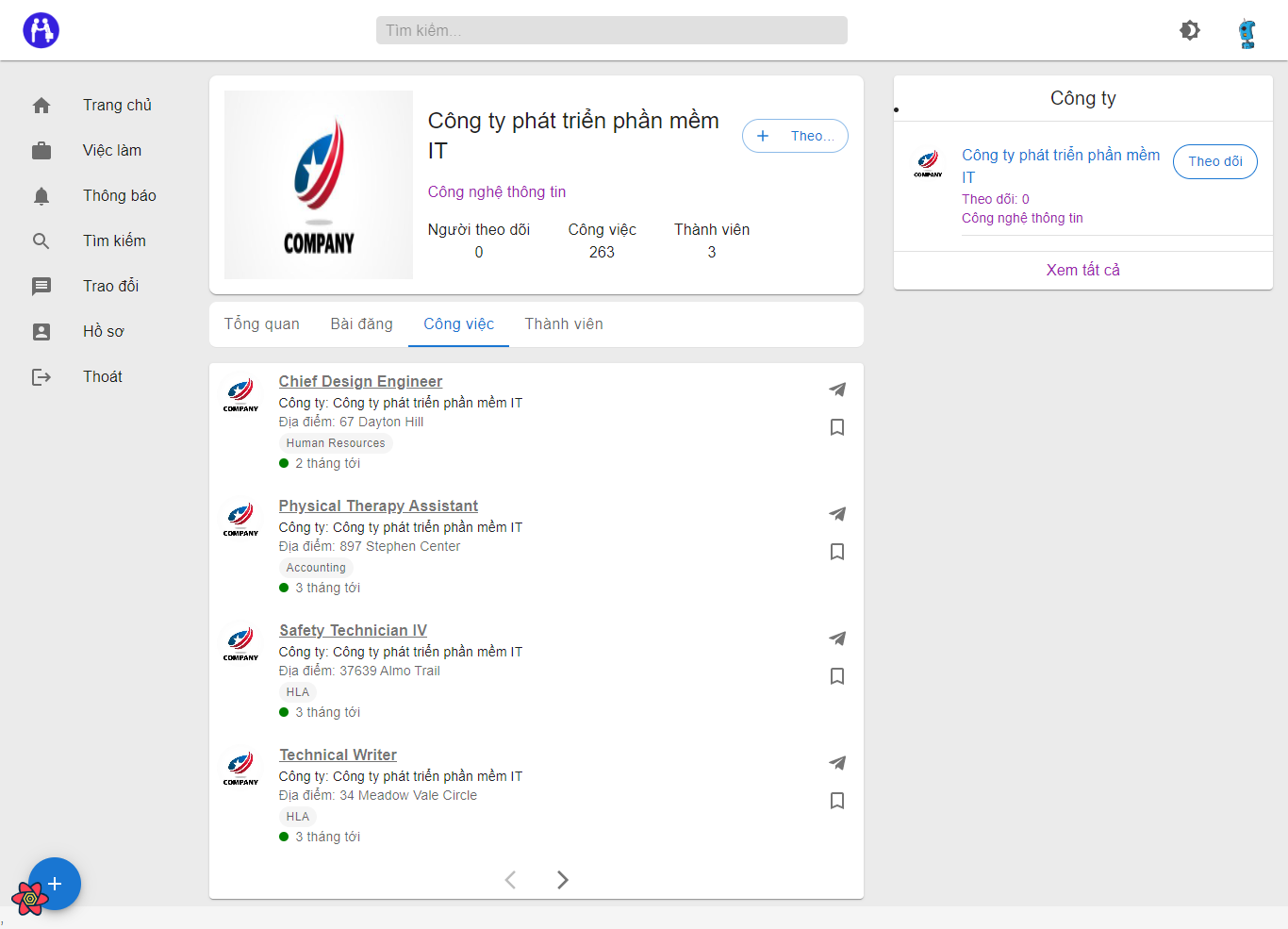
Giao diện cung cấp cho người dùng có thể xem và sửa (nếu là trang hồ sơ của bản thân): ảnh đại diện, tên người dùng, email người dùng, học vấn, kỹ năng… Đồng thời, người dùng có thể xem các bài đăng của mình ở đây.



**Hình 39** Giao diện trang hồ sơ người dùng

##### Giao diện trang công ty

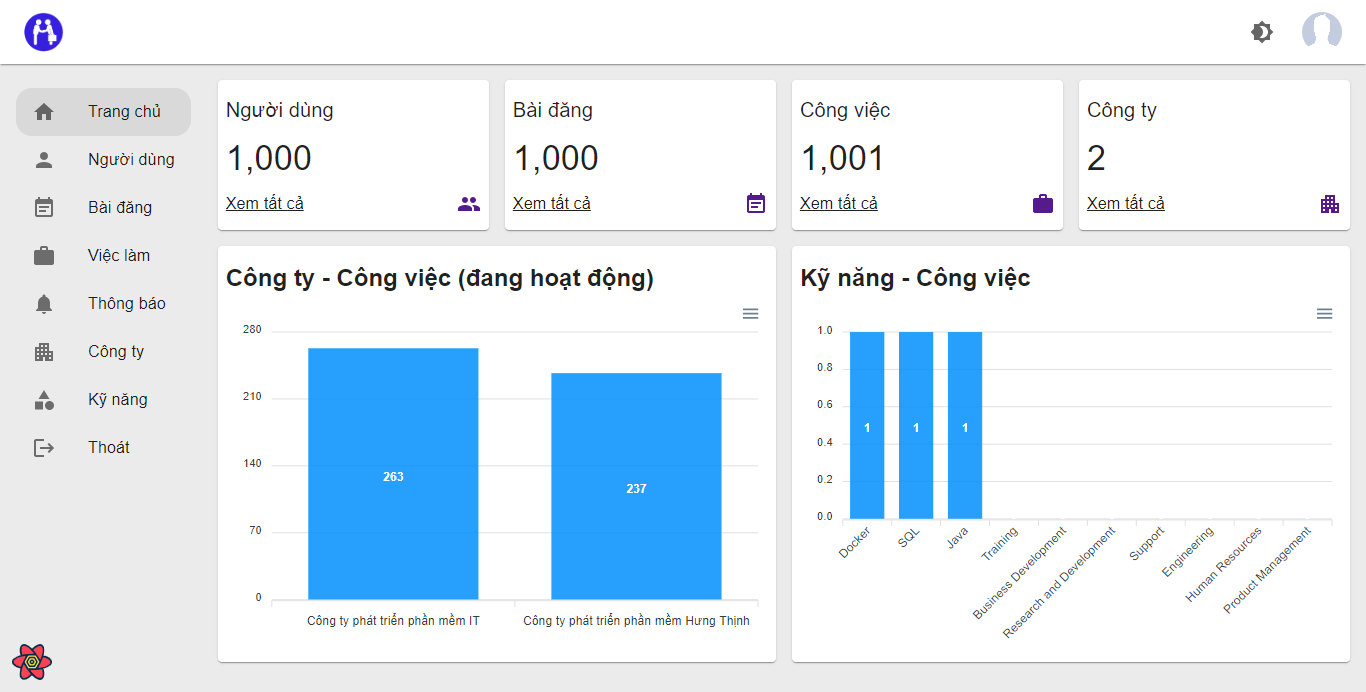
Giao diện trang công ty cung cấp: thông tin tổng quan của công ty, các hoạt động gần đây của công ty, việc làm của công ty, thành viên của công ty….



**Hình 40** Giao diện trang công ty

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

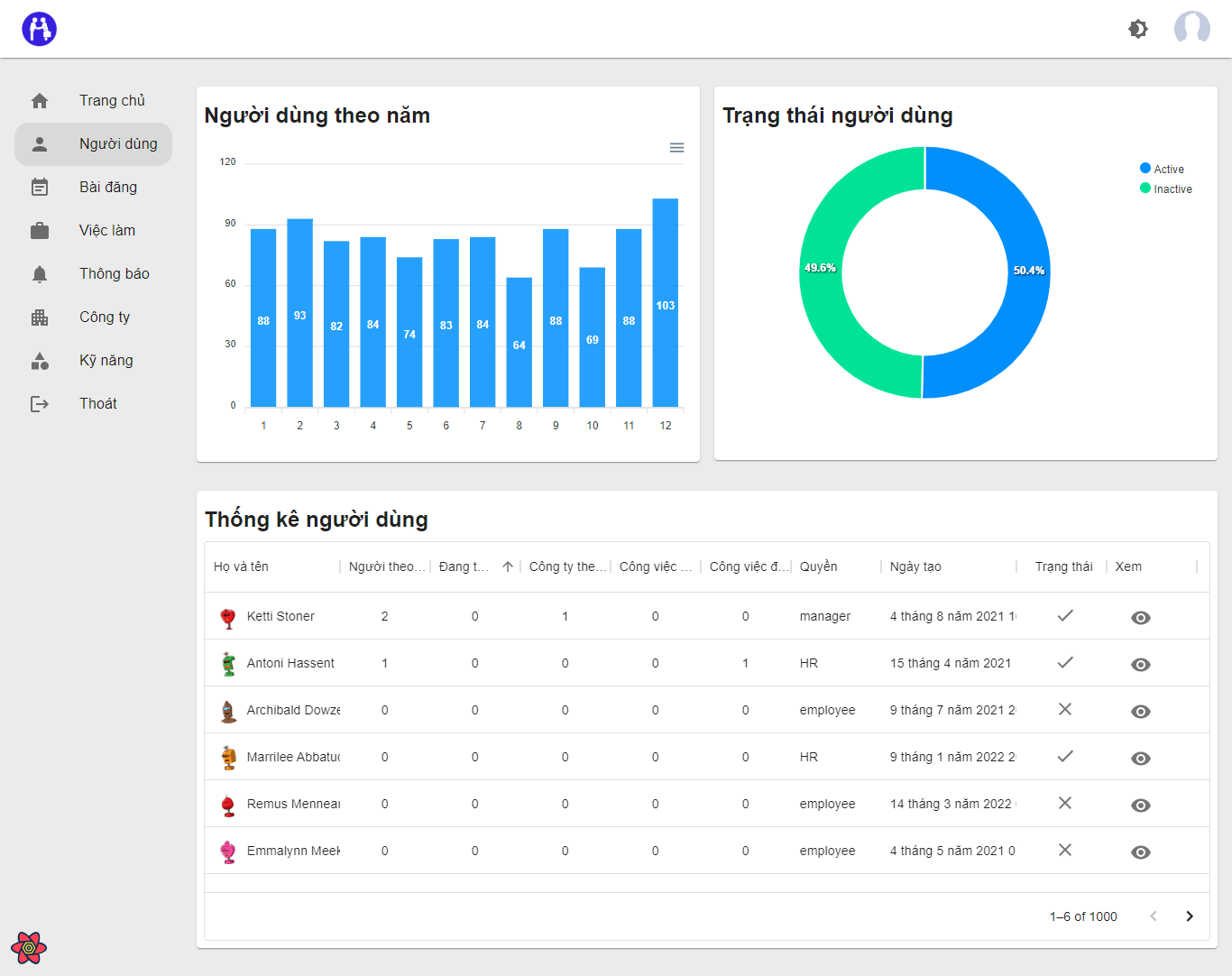
Giao diện trang chủ quản trị viên thông kê tổng quát số lượng các thực thể có trong hệ thống: người dùng, bài đăng, công ty, công việc (tin tuyển dụng). Hơn nữa, giao diện cung cấp biểu đồ thống kê



**Hình 41** Giao diện trang chủ quản trị viên

##### 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 42** Giao diện trang người dùng quản trị viên

## Kiểm thử

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Kịch bản test | Mô tả test case | Các bước test | Dữ liệu test | Kết quản mong đợi | Kết quả thực tế | Trạng thái |
|  | Hiển thị danh sách việc làm cần xác thực | Quản trị viên muốn xem danh sách tin tuyể dụng cần duyệt | Chọn chức năng “Thông báo tạo việc làm” |  | Hiện thị danh sách tin tuyển dụng cần duyệt | Như mong đợi | Pass |
|  | Phê duyệt việc làm | Quản trị viên chấp nhận việc làm | Chọn chức năng “Thông báo tạo việc làm”  Nhấn nút “Chấp nhận” |  | Tin tuyển dụng được mở  Gửi thông báo đến chủ sở hữu  Gửi thông báo đến người theo dõi công ty | Như mong đợi | Pass |
|  | 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 | Chọn chức năng “Thông báo tạo việc làm”  Nhấn nút “Từ chối”  Hiển thị giao diện từ chối  Nộp | note: “” | Thông báo lỗi: “note được yêu cầu” | Như mong đợi | Pass |
|  | Từ chối thành công | Quản trị viên từ chối tin tuyển dụng | Chọn chức năng “Thông báo tạo việc làm”  Nhấn nút “Từ chối”  Hiển thị giao diện từ chối  Điền đơn  Nộp | note: “yêu cầu của công việc quá mức sơ sài gây khó hiểu cho ứng viên” | Cập nhật công việc  Gửi thông báo từ chối đến chủ sở hữu | Như mong đợi | Pass |
|  | Xem danh sách đã duyệt tin tuyển dụng | Quản trị viên muốn xem danh sách các tin đã duyệt | Chọn chức năng “Thông báo xác nhận công việc” |  | Hiển thị danh sách tin tuyển dụng đã duyệt | Như mong đợi | Pass |

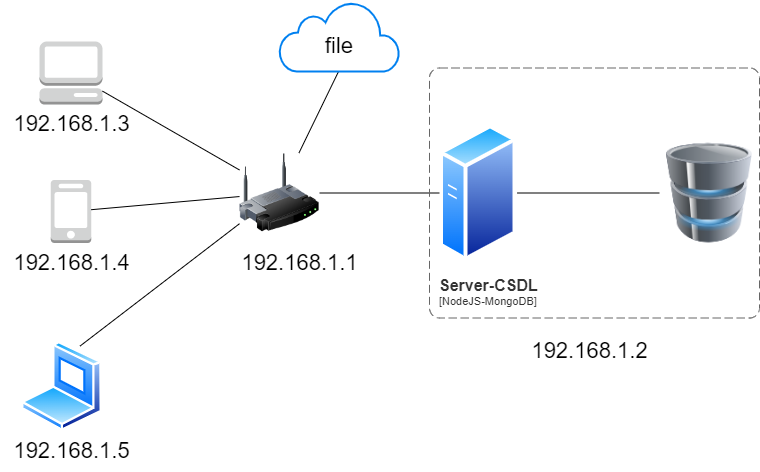
**Bảng 24** Bảng kiểm thử chức năng duyệt tin tuyển dụng

### Kiểm thử chức năng ứng tuyển công việc

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Kịch bản test | Mô tả test case | Các bước test | Dữ liệu test | Kết quản mong đợi | Kết quả thực tế | Trạng thái |
|  | Không thành công vì nhập sai định dạng email | Người tìm việc muốn ứng tuyển công việc nhưng điền sai định dạng email | Nhấn nút “Apply”  Hiện thị đơn ứng tuyển  Nhập thông tin trường email | email: abc | Thông báo lỗi: “email sai định dạng” | Như mong đợi | Pass |
|  | Không thành công vì nhập sai số điện thoại | Người tìm việc muốn ứng tuyển công việc nhưng điền sai định dạng phone | Nhấn nút “Apply”  Hiện thị đơn ứng tuyển  Nhập thông tin trường phone | email: [thanh.nd183991@sis.hust.edu.vn](mailto:thanh.nd183991@sis.hust.edu.vn)  phone:012345 | Thông báo lỗi: “Số điện thoại không đúng định dạng” | Như mong đơi | Pass |
|  | Không thành công do thiếu thông tin CV | Người tìm việc muốn ứng tuyển công việc nhưng không nộp CV | Nhấn nút “Apply”  Hiện thị đơn ứng tuyển  Điền form ứng tuyển  Nộp đơn | email: [thanh.nd183991@sis.hust.edu.vn](mailto:thanh.nd183991@sis.hust.edu.vn)  phone:0386851140 | Thông báo lỗi: “Vui lòng chọn CV” | Như mong đợi | Pass |
|  | Không thành công do đã hết hạn ứng tuyển | Người tìm việc muốn ứng tuyển công việc nhưng đã quá hạn ứng tuyển | Nhấn nút “Apply”  Hiện thị đơn ứng tuyển  Điền form ứng tuyển  Nộp đơn | email: [thanh.nd183991@sis.hust.edu.vn](mailto:thanh.nd183991@sis.hust.edu.vn)  phone:0386851140  CV: https://www.orimi.com/pdf-test.pdf | Thông báo lỗi: “Hết hạn ứng tuyển” | Như mong đợi | Pass |
|  | Không thành công do đã ứng tuyển | Người tìm việc muốn ứng tuyển công việc nhưng đã ứng tuyển trước đó | Nhấn nút “Apply”  Hiện thị đơn ứng tuyển  Điền form ứng tuyển  Nộp đơn | email: [thanh.nd183991@sis.hust.edu.vn](mailto:thanh.nd183991@sis.hust.edu.vn)  phone:0386851140  CV: https://www.orimi.com/pdf-test.pdf | Thông báo lỗi: “Bạn đã ứng tuyển rồi” | Như mong đợi | Pass |
|  | Ửng tuyển thành công | Người tìm việc ứng tuyển thành công | Nhấn nút “Apply”  Điền form tạo việc | email: [thanh.nd183991@sis.hust.edu.vn](mailto:thanh.nd183991@sis.hust.edu.vn)  phone:0386851140  CV: https://www.orimi.com/pdf-test.pdf | Cập nhật danh sách ứn tuyển  Gửi thông báo đến nhà tuyển dụng  Thông báo: “Apply thành công” | Như mong đợi | Pass |

**Bảng 25** Bảng kiểm thử chức năng ứng tuyển công việc

## Triển khai



**Hình 43** Mô hình triển khai của hệ thống

Hệ thống được đi vào thử nghiệm và chạy ổn định trên mạng LAN (Local Area Network). Mảy 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. Mảy 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 26** Bảng thông số cầu hình server

# Các giải pháp và đóng góp nổi bật

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

### 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.

### 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 hiện 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 hằng nghìn, thậm chí hàng triệu bảng 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 dối 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, 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, 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 nhiệm 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.

## Tạo CV

### 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.

### Giải pháp

Có rất nhiều giải pháp cho phương pháp này:

Phương pháp 1: tạo CV sử dụng MS 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 sử dụng phương pháp 3 cho việc tạo CV cho người dùng

## Thiết kế Category Collection

### Đặt vấn đề

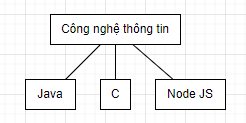
Khi nhà tuyển dụng tạo một tin tuyển dụng, hệ thông sẽ lưu thông tin tuyển dụng đồng thời lưu danh sách các kỹ năng của công việc đó trong Category Collection. Tuy nhiên, việc lưu thông tin các kỹ năng cần phải được đảm bảo sao cho việc truy vấn gợi ý, tìm kiếm các tin tuyển dụng của người dùng phải 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.

### 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 sự dụng các chữ cái in thường) giúp đảm bảo tính chính xác, tránh trùng lặp.

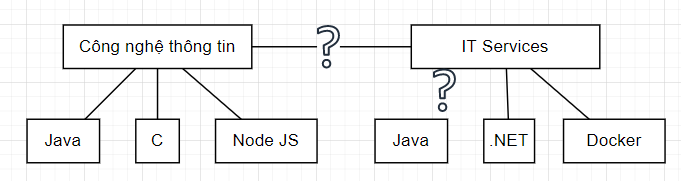
Ví dụ: Cong nghệ thông tin, công nghệ Thông tin, Công nghe thong tin… được chuyển đổi qua slug là: cong-nghe-thong-tin.

Khi nhà tuyển dụng của một công ty muốn tạo một bài viết liên quan đến công ty đó, hệ thống coi lĩnh vựa 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).



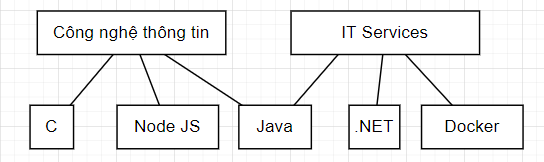
**Hình 44** Mối quan hệ kỹ năng khi tạo tin tuyển dụng

Ở mô hình này, 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, 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.



**Hình 45** 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 46 Thiết kế mối quan hệ giữa lĩnh vực tương đồng

Nhìn vào hình vẽ ta có thể dữ 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 27** Thiết kế Category Collection

## Thiết kế Notification Collection

### 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.

### 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/tử chối ngay trên chính thông báo mà không phải lọc rồi mới duyệt công ty).

Trường người nhận được thiết kế là một mảng giúp làm giảm đáng kể số lượng bản ghi được lưu trữ trong trường hợp người dùng tạo bài đăng hay công ty tạo công việc. Bảng –tham chiếu—thiết kế Notification Collection.

## Lưu trữ file

### Đăt 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.

### 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 diểm 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 rát nhanh vì file được đặt luôn tạo server. Tuy nhiên, phương pháp này chỉ áp dụng đối với hệ thống có số lượng file nhỏ vì khi file có sô 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 cloudinary giúp lưu các file: ảnh đại diện, CV, ảnh bài đăng, video…

# Kết luận và hướng phát triển

## Kết luận

Hệ thống xây dựng đã cung cấp được các chức năng cơ bản giúp nhà tuyển dụng và người lao động 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 thế 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, 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, 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.

Qua quá trình làm đồ án tốt nghiệp, em đã học được nhiều công nghệ mới, cũng như kỹ năng viết báo cáo. Việc làm đồ án tốt nghiệp giúp em 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 của em.

Do thời gian làm đồ án có hạn, cũng như hạn chế của năng lực bản thân nên hệ thống em xây dựng vẫn còn nhiều thiếu xót. 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…

## 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 đươc liên kết khóa ngoại với id của công việc mà người tim việc đánh giá.

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 nhiệm của người dùng.

# Tài liệu tham khảo

Trang đặc tả JWT của tổ chức auth0: <https://jwt.io/introduction>

@article{kudiabor2020state,

title={State management with React-Redux},

author={Kudiabor, Dominic Travis},

year={2020}

}

@article{li2020research,

title={Research on MVP design pattern modeling based on MDA},

author={Li, Dan Dan and Liu, Xiao Yan},

journal={Procedia Computer Science},

volume={166},

pages={51--56},

year={2020},

publisher={Elsevier}

}

# Phụ lục