

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Phát triển website tuyển dụng thông minh

CHU VĂN HIẾU

hieu.cv194046@sis.hust.edu.vn

Ngành Khoa học máy tính

Giảng viên hướng dẫn: ThS. Vũ Đức Vượng

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 12/2025

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Phát triển website tuyển dụng thông minh

CHU VĂN HIỂU

hieu.cv194046@sis.hust.edu.vn

Ngành Khoa học máy tính

Giảng viên hướng dẫn: ThS. Vũ Đức Vương

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 12/2025

LỜI CẢM ƠN

Trước khi bắt đầu viết đồ án, em xin phép gửi lời cảm ơn đến gia đình em - những người đã luôn luôn quan tâm trong quá trình em thực hiện đồ án tốt nghiệp. Em cũng cảm thấy rất may mắn khi có được sự đồng hành và giúp đỡ của các thầy cô đại học Bách Khoa trong suốt quá trình học tập của mình.

Ngoài ra, em cũng xin gửi lời cảm ơn đặc biệt tới thầy **Vũ Đức Vượng**, thầy đã cho em những lời khuyên, kinh nghiệm của quý báu để giúp em hoàn thành đồ án tốt nghiệp. Trong quá trình làm đồ án em đã gặp nhiều khó khăn và vất vả nên em biết mình cần phải trân trọng những kiến thức và kinh nghiệm mà mình đã học được từ thầy.

Lần nữa, em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong bối cảnh thị trường lao động ngày càng cạnh tranh và chịu tác động mạnh mẽ của quá trình chuyển đổi số, hoạt động tuyển dụng theo phương thức truyền thống đang bộc lộ nhiều hạn chế, đặc biệt là trong việc tìm kiếm và lựa chọn ứng viên phù hợp với các yêu cầu đã đặt ra. Hiện nay, nhiều website tuyển dụng vẫn chủ yếu hoạt động theo mô hình đăng tin và tìm kiếm thủ công, trong đó việc kết nối giữa nhà tuyển dụng và ứng viên phần lớn dựa vào việc hai bên tự chủ động tìm kiếm lẫn nhau, chưa khai thác hiệu quả dữ liệu để hỗ trợ quá trình sàng lọc và gợi ý.

Xuất phát từ thực trạng đó, em lựa chọn thực hiện đồ án "**Phát triển website tuyển dụng thông minh**" dựa trên việc kết hợp yêu cầu của nhà tuyển dụng, mong muốn của ứng viên và hành vi thực tế của ứng viên trong quá trình tìm kiếm việc làm, từ đó đưa ra các đề xuất mang tính cá nhân hóa, phù hợp cho cả hai phía. Hướng tiếp cận này giúp ứng viên và nhà tuyển dụng có cơ sở rõ ràng hơn trong việc đưa ra quyết định, đồng thời góp phần giảm thời gian sàng lọc hồ sơ và hạn chế các vòng phỏng vấn kéo dài nhưng kém hiệu quả.

Website này đóng vai trò như một nền tảng kết nối giữa nhà tuyển dụng và ứng viên thông qua các chức năng như đề xuất công việc cho ứng viên, đề xuất ứng viên cho nhà tuyển dụng, cơ chế thông báo chủ động và các chức năng hỗ trợ tuyển dụng khác. Kết quả thu được đáp ứng đầy đủ các yêu cầu chức năng đã đặt ra, cho phép quản lý hiệu quả hồ sơ ứng viên, nhu cầu người dùng và các dữ liệu đề xuất.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

ABSTRACT

In the context of an increasingly competitive labor market and the strong impact of digital transformation, traditional recruitment practices have revealed several limitations, particularly in identifying and selecting candidates who meet predefined requirements. At present, many recruitment websites mainly operate based on a job-posting and manual search model, in which the connection between employers and candidates largely relies on both parties proactively searching for each other, without effectively leveraging data to support screening and recommendation processes.

From this situation, this thesis undertakes the project entitled **Development of an Intelligent Recruitment Website**, which is based on the integration of employers' requirements, candidates' preferences, and candidates' actual behaviors during the job search process. On that basis, the system provides personalized recommendations that are suitable for both candidates and employers. This approach enables both parties to make more informed decisions, while also contributing to a reduction in resume screening time and limiting prolonged yet inefficient interview processes.

The website serves as a platform connecting employers and candidates through features such as job recommendations for candidates, candidate recommendations for employers, proactive notification mechanisms, and other recruitment support functionalities. The obtained results fully meet the predefined functional requirements, allowing effective management of candidate profiles, user needs, and recommendation-related data.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.2 Tổng quan chức năng	4
2.2.1 Biểu đồ use case tổng quát	5
2.2.2 Biểu đồ use case phân rã đăng nhập	6
2.2.3 Biểu đồ use case phân rã tìm kiếm	7
2.2.4 Biểu đồ use case phân rã yêu thích.....	7
2.2.5 Biểu đồ use case phân rã tạo hồ sơ cá nhân.....	8
2.2.6 Biểu đồ use case phân rã ứng tuyển.....	8
2.2.7 Biểu đồ use case phân rã tạo công ty	9
2.2.8 Biểu đồ use case phân rã quản lý tin tuyển dụng.....	9
2.2.9 Biểu đồ use case phân rã quản lý hồ sơ ứng tuyển	10
2.2.10 Biểu đồ use case phân rã phê duyệt	10
2.2.11 Biểu đồ use case phân rã quản lý người dùng	11
2.2.12 Biểu đồ use case phân rã tính điểm phù hợp.....	11
2.2.13 Biểu đồ use case phân rã tạo đề xuất	12
2.2.14 Biểu đồ use case phân rã nhật ký hành vi.....	12
2.2.15 Biểu đồ use case phân rã gửi thông báo	13
2.2.16 Quy trình nghiệp vụ.....	13

2.3 Đặc tả chức năng	17
2.3.1 Đặc tả use case đăng nhập.....	17
2.3.2 Đặc tả use case mong muốn công việc	19
2.3.3 Đặc tả use case mong muốn tuyển dụng	20
2.3.4 Đặc tả use case tạo công ty	21
2.3.5 Đặc tả use case tạo tin tuyển dụng	22
2.3.6 Đặc tả use case phê duyệt tin tuyển dụng.....	24
2.3.7 Đặc tả use case tạo đề xuất tin tuyển dụng	25
2.3.8 Đặc tả use case tạo đề xuất ứng viên	26
2.3.9 Đặc tả use case ứng tuyển	27
2.4 Yêu cầu phi chức năng	28
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	30
3.1 ReactTS.....	30
3.1.1 React	30
3.1.2 TypeScript.....	31
3.2 Tailwind CSS	31
3.3 NodeJS	32
3.4 MySQL	33
3.5 Phương pháp đề xuất dựa theo nội dung (Content-based Filtering – CBF)..	33
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	35
4.1 Thiết kế kiến trúc.....	35
4.1.1 Lựa chọn kiến trúc phần mềm	35
4.1.2 Thiết kế tổng quan.....	36
4.1.3 Thiết kế chi tiết gói	39
4.2 Thiết kế chi tiết.....	42
4.2.1 Thiết kế giao diện	42

4.2.2 Thiết kế lớp	47
4.2.3 Thiết kế cơ sở dữ liệu	53
4.3 Xây dựng ứng dụng.....	62
4.3.1 Thư viện và công cụ sử dụng.....	62
4.3.2 Kết quả đạt được	63
4.3.3 Minh họa các chức năng chính	63
4.4 Kiểm thử.....	69
4.4.1 Kiểm thử chức năng “Tạo tin tuyển dụng”	69
4.4.2 Kiểm thử chức năng “Đề xuất công việc”	69
4.4.3 Kiểm thử chức năng “Đề xuất ứng viên”.....	70
4.5 Triển khai	70
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	73
5.1 Chuẩn hóa dữ liệu.....	73
5.1.1 Đặt vấn đề	73
5.1.2 Giải pháp	73
5.1.3 Kết quả thực hiện	77
5.2 Tính hành vi người dùng	78
5.2.1 Đặt vấn đề	78
5.2.2 Giải pháp	78
5.2.3 Kết quả thực hiện	79
5.3 Xây dựng công thức tính điểm phù hợp.....	79
5.3.1 Đặt vấn đề	79
5.3.2 Giải pháp	80
5.3.3 Công thức tính điểm.....	81
5.3.4 Kết quả thực hiện	87

5.4 Gửi thông báo thông minh	87
5.4.1 Đặt vấn đề	87
5.4.2 Giải pháp	87
5.4.3 Kết quả thực hiện	88
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	89
6.1 Kết luận	89
6.2 Hướng phát triển.....	89
TÀI LIỆU THAM KHẢO.....	91

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quát	5
Hình 2.2	Biểu đồ use case phân rã đăng nhập	6
Hình 2.3	Biểu đồ use case phân rã tìm kiếm	7
Hình 2.4	Biểu đồ use case phân rã yêu thích	7
Hình 2.5	Biểu đồ use case phân rã tạo hồ sơ cá nhân	8
Hình 2.6	Biểu đồ use case phân rã ứng tuyển	8
Hình 2.7	Biểu đồ use case phân rã tạo công ty	9
Hình 2.8	Biểu đồ use case phân rã quản lý tin tuyển dụng	9
Hình 2.9	Biểu đồ use case phân rã quản lý hồ sơ ứng tuyển	10
Hình 2.10	Biểu đồ use case phân rã phê duyệt	10
Hình 2.11	Biểu đồ use case phân rã quản lý người dùng	11
Hình 2.12	Biểu đồ use case phân rã tính điểm phù hợp	11
Hình 2.13	Biểu đồ use case phân rã tạo đề xuất	12
Hình 2.14	Biểu đồ use case phân rã nhật ký hành vi	12
Hình 2.15	Biểu đồ use case phân rã gửi thông báo	13
Hình 2.16	Biểu đồ hoạt động quy trình đăng tin tuyển dụng	14
Hình 2.17	Biểu đồ hoạt động quy trình đề xuất tin tuyển dụng	15
Hình 2.18	Biểu đồ hoạt động quy trình đề xuất ứng viên	16
Hình 2.19	Biểu đồ hoạt động quy trình ứng tuyển	17
Hình 3.1	Minh họa thư viện React	30
Hình 4.1	Kiến trúc phân lớp của hệ thống [10]	35
Hình 4.2	Biểu đồ phụ thuộc gói	37
Hình 4.3	Thiết kế gói dịch vụ Frontend	39
Hình 4.4	Thiết kế gói các trang giao diện của tầng User Interface	40
Hình 4.5	Thiết kế gói Controllers và Services Backend	41
Hình 4.6	Thiết kế gói định tuyến API Backend	42
Hình 4.7	Thiết kế giao diện đăng nhập	43
Hình 4.8	Thiết kế giao diện màn hình chính	44
Hình 4.9	Thiết kế giao diện trang cá nhân	45
Hình 4.10	Thiết kế giao diện quản trị viên	46
Hình 4.11	Thiết kế chi tiết lớp User	47
Hình 4.12	Thiết kế chi tiết lớp Preference	47
Hình 4.13	Thiết kế chi tiết lớp Job	48
Hình 4.14	Thiết kế chi tiết lớp Recommendation	48

Hình 4.15	Thiết kế chi tiết lớp Application	49
Hình 4.16	Biểu đồ trình tự use case Tạo tin tuyển dụng	50
Hình 4.17	Biểu đồ trình tự use case Ứng tuyển	51
Hình 4.18	Biểu đồ trình tự use case Mong muốn công việc	51
Hình 4.19	Biểu đồ trình tự use case Mong muốn tuyển dụng	52
Hình 4.20	Biểu đồ trình tự use case Tạo đề xuất	52
Hình 4.21	Biểu đồ thực thể liên kết	53
Hình 4.22	Giao diện "Nộp đơn ứng tuyển"	64
Hình 4.23	Giao diện "Quản lý tin tuyển dụng"	65
Hình 4.24	Giao diện "Cập nhật tin tuyển dụng"	65
Hình 4.25	Giao diện "Quản lý đơn ứng tuyển"	66
Hình 4.26	Giao diện "Nhu cầu tuyển dụng"	66
Hình 4.27	Giao diện "Mong muốn nghề nghiệp"	67
Hình 4.28	Giao diện "Đề xuất ứng viên"	67
Hình 4.29	Giao diện "Trang chủ xem tin tuyển dụng"	68
Hình 4.30	Giao diện "Đề xuất tin tuyển dụng"	68
Hình 4.31	Kết quả đo bằng công cụ Sonarqube	71
Hình 4.32	Kết quả đo bằng công cụ Lighthouse	72
Hình 5.1	Điểm phù hợp và lý do đề xuất	87
Hình 5.2	Email thông báo cho nhà tuyển dụng	88

DANH MỤC BẢNG BIỂU

Bảng 2.1	Bảng đặc tả use case “Đăng nhập”	18
Bảng 2.2	Bảng đặc tả use case “Mong muốn công việc”	19
Bảng 2.3	Bảng đặc tả use case “Mong muốn tuyển dụng”	20
Bảng 2.4	Bảng đặc tả use case “Tạo công ty”	21
Bảng 2.5	Bảng đặc tả use case “Tạo tin tuyển dụng”	22
Bảng 2.6	Bảng đặc tả use case “Phê duyệt tin tuyển dụng”	24
Bảng 2.7	Bảng đặc tả use case “Tạo đề xuất tin tuyển dụng”	25
Bảng 2.8	Bảng đặc tả use case “Tạo đề xuất ứng viên”	26
Bảng 2.9	Bảng đặc tả use case “Ứng tuyển”	27
Bảng 4.1	Bảng Users	54
Bảng 4.2	Bảng Companies	55
Bảng 4.3	Bảng CompanyVerifications	55
Bảng 4.4	Bảng Jobs	55
Bảng 4.5	Bảng JobApprovals	56
Bảng 4.6	Bảng JobRequiredSkills	56
Bảng 4.7	Bảng Tags	56
Bảng 4.8	Bảng JobTags	57
Bảng 4.9	Bảng UserFavoriteJobs	57
Bảng 4.10	Bảng Applications	57
Bảng 4.11	Bảng UserInterestHistory	57
Bảng 4.12	Bảng CareerPreferences	58
Bảng 4.13	Bảng CareerPreferenceTags	58
Bảng 4.14	Bảng RecruiterPreferences	58
Bảng 4.15	Bảng RecruiterRequiredSkills	59
Bảng 4.16	Bảng RecruiterPreferenceTags	59
Bảng 4.17	Bảng UserBehaviorProfile	59
Bảng 4.18	Bảng JobRecommendations	59
Bảng 4.19	Bảng CandidateRecommendations	60
Bảng 4.20	Bảng Skills	60
Bảng 4.21	Bảng UserSkills	60
Bảng 4.22	Bảng UserVector	60
Bảng 4.23	Bảng RecruiterVector	61
Bảng 4.24	Bảng JobVector	61
Bảng 4.25	Bảng UserJobMatrix	61

Bảng 4.26	Danh sách thư viện và công cụ sử dụng	62
Bảng 4.27	Thống kê thông tin ứng dụng	63
Bảng 4.28	Test case kiểm thử chức năng tạo tin tuyển dụng	69
Bảng 4.29	Test case kiểm thử chức năng đề xuất công việc	69
Bảng 4.30	Bảng kiểm thử chức năng đề xuất ứng viên	70
Bảng 4.31	Cấu hình môi trường thực nghiệm	71

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
BE	Tầng xử lý phía máy chủ (Backend)
CBF	Gợi ý dựa trên nội dung (Content-based Filtering)
CRUD	Bốn thao tác dữ liệu: Create–Read–Update–Delete
CSDL	Cơ sở dữ liệu (Database)
event-driven	Mô hình hướng sự kiện (Event-driven architecture)
FE	Tầng giao diện phía người dùng (Frontend)
HTML/JSX	HTML (HyperText Markup Language) và JSX (JavaScript XML)
JSON	Định dạng dữ liệu JavaScript Object Notation (JSON)
JWT	Chuẩn token xác thực JSON Web Token (JWT)
non-blocking I/O	Cơ chế I/O không chặn (Non-blocking Input/Output)
Prisma ORM	Công cụ ORM Prisma (Prisma Object-Relational Mapping)
SEO	Tối ưu hóa công cụ tìm kiếm (Search Engine Optimization)
TF–IDF	Phương pháp biểu diễn văn bản dựa trên trọng số Term Frequency–Inverse Document Frequency
TS/JS	TypeScript/JavaScript
UI	Giao diện người dùng (User Interface)
utility-first	Hướng tiếp cận CSS dựa trên lớp tiện ích (Utility-first CSS)

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong những năm gần đây, thị trường lao động đã chịu sự tác động mạnh mẽ của quá trình chuyển đổi số và sự phát triển nhanh chóng của công nghệ thông tin. Số lượng doanh nghiệp có nhu cầu tuyển dụng ngày càng gia tăng, kéo theo sự đa dạng về vị trí việc làm và yêu cầu kỹ năng. Tuy nhiên, song song với đó, người lao động, đặc biệt là sinh viên mới tốt nghiệp và người tìm việc, cũng gặp không ít khó khăn trong việc tiếp cận các cơ hội việc làm phù hợp với năng lực, kinh nghiệm và định hướng nghề nghiệp của bản thân.

Trên thực tế, hoạt động tuyển dụng hiện nay vẫn chủ yếu dựa trên các nền tảng đăng tin việc làm và phương thức tìm kiếm thủ công. Nhà tuyển dụng phải xử lý một khối lượng lớn hồ sơ ứng tuyển, trong đó không ít hồ sơ chưa đáp ứng đúng yêu cầu của vị trí tuyển dụng, gây tốn kém thời gian và nguồn lực cho quá trình sàng lọc. Ở chiều ngược lại, ứng viên thường phải tự tìm kiếm, đọc và đánh giá từng tin tuyển dụng riêng lẻ, dẫn đến việc bỏ lỡ những cơ hội phù hợp hoặc ứng tuyển vào các vị trí không thực sự tương thích.

Những hạn chế nêu trên cho thấy hệ thống gợi ý hiệu quả giữa nhà tuyển dụng và ứng viên vẫn chưa được giải quyết một cách triệt để. Nếu bài toán này được xử lý tốt, quá trình tuyển dụng có thể trở nên nhanh chóng, chính xác và minh bạch hơn, mang lại lợi ích thiết thực cho cả nhà tuyển dụng lẫn người tìm việc. Bên cạnh lĩnh vực tuyển dụng, việc giải quyết bài toán đề xuất hiệu quả còn có tiềm năng được áp dụng trong nhiều lĩnh vực khác như giới thiệu sản phẩm, hiển thị thông tin và video giải trí, nơi nhu cầu cá nhân hóa và xử lý dữ liệu ngày càng trở nên quan trọng.

1.2 Mục tiêu và phạm vi đề tài

Các hệ thống tuyển dụng hiện nay chủ yếu cung cấp những chức năng cơ bản như đăng tin tuyển dụng, tìm kiếm việc làm, tiếp nhận và quản lý hồ sơ ứng viên. Một số nền tảng đã bước đầu bổ sung các chức năng gợi ý việc làm hoặc đề xuất ứng viên dựa trên thông tin hồ sơ, qua đó cải thiện phần nào trải nghiệm người dùng so với phương thức tuyển dụng truyền thống. Tuy nhiên, khi so sánh tổng quan các sản phẩm hiện có, có thể nhận thấy rằng phần lớn các hệ thống vẫn tập trung vào việc hiển thị thông tin và hỗ trợ tìm kiếm ở mức cơ bản, trong khi khả năng hỗ trợ đánh giá mức độ phù hợp giữa ứng viên và vị trí tuyển dụng còn hạn chế, chưa đáp ứng tốt nhu cầu thực tế của cả hai phía.

Từ những phân tích trên, có thể khái quát rằng các hệ thống tuyển dụng hiện tại còn gặp nhiều hạn chế trong việc khai thác dữ liệu người dùng và hỗ trợ quá trình ra quyết định. Trên cơ sở đó, đề tài hướng tới việc giải quyết bài toán nâng cao hiệu quả kết nối giữa nhà tuyển dụng và ứng viên, đồng thời khắc phục những hạn chế liên quan đến đánh giá mức độ phù hợp. Phần mềm được định hướng phát triển với các chức năng chính như quản lý thông tin tuyển dụng và hồ sơ ứng viên, hỗ trợ đề xuất việc làm và ứng viên, cũng như cung cấp các cơ chế hỗ trợ tương tác và thông báo trong quá trình tuyển dụng.

1.3 Định hướng giải pháp

Nhằm giải quyết các vấn đề đã được xác định ở phần trước, đồ án lựa chọn định hướng xây dựng một website tuyển dụng thông minh theo mô hình ứng dụng website. Hệ thống được phát triển dựa trên hướng tiếp cận gợi ý, kết hợp việc khai thác thông tin hồ sơ và dữ liệu hành vi của người dùng trong quá trình tuyển dụng. Định hướng này được lựa chọn nhằm phù hợp với đặc thù của bài toán tuyển dụng thực tế, đồng thời tạo điều kiện cho việc mở rộng và cải tiến hệ thống trong tương lai.

Theo định hướng trên, đồ án tiến hành xây dựng một website tuyển dụng đáp ứng các chức năng chính như quản lý thông tin nhà tuyển dụng và ứng viên, đăng tin tuyển dụng, tiếp nhận hồ sơ ứng tuyển và hỗ trợ đề xuất việc làm cho ứng viên cũng như đề xuất ứng viên cho nhà tuyển dụng. Hệ thống được thiết kế với giao diện thân thiện, dễ sử dụng và phù hợp với nhiều đối tượng người dùng, góp phần hỗ trợ quá trình tuyển dụng diễn ra thuận tiện và hiệu quả hơn.

Đóng góp chính của đồ án là việc đề xuất và triển khai một mô hình website tuyển dụng thông minh hỗ trợ kết nối và sàng lọc giữa nhà tuyển dụng và ứng viên. Kết quả đạt được cho thấy hệ thống đáp ứng các yêu cầu chức năng đã đề ra, vận hành ổn định và có khả năng hỗ trợ cải thiện hiệu quả tuyển dụng trong thực tế. Đồng thời, hệ thống tạo nền tảng cho việc tiếp tục phát triển và hoàn thiện các chức năng thông minh hơn trong các giai đoạn tiếp theo.

1.4 Bố cục đồ án

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

Chương 2 trình bày nhu cầu thực tế và hiện trạng của các hệ thống tuyển dụng hiện nay. Trên cơ sở đó, chương này phân tích các ưu điểm và hạn chế của những hệ thống hiện có nhằm xác định hướng phát triển phù hợp cho đồ án. Kết hợp với các vấn đề đã được nêu trong Mục 1.1, chương 2 tiến hành nghiên cứu, so sánh và phân tích một số hệ thống tuyển dụng tiêu biểu để làm rõ các chức năng cần xây dựng, tạo cơ sở cho việc xây dựng các tài liệu đặc tả yêu cầu, bao gồm yêu cầu chức

năng và yêu cầu phi chức năng của hệ thống.

Chương 3 được xây dựng dựa trên các yêu cầu và tính năng đã được xác định ở Chương 2, kết hợp với kiến thức chuyên môn và định hướng cá nhân của sinh viên nhằm lựa chọn và xác định các công nghệ sử dụng trong quá trình phát triển hệ thống. Chương này trình bày tổng quan về các công nghệ được lựa chọn, đồng thời phân tích vai trò, lợi ích và lý do sử dụng của từng công nghệ trong việc triển khai các thành phần của hệ thống.

Dựa trên các kết quả phân tích ở Chương 2, Chương 4 trình bày thiết kế tổng thể và thiết kế chi tiết của hệ thống, bao gồm thiết kế kiến trúc, thiết kế giao diện người dùng, thiết kế cơ sở dữ liệu và thiết kế các lớp chức năng. Chương này tập trung làm rõ cách thức hệ thống được xây dựng nhằm đảm bảo tính khả thi, khả năng mở rộng và trải nghiệm người dùng. Bên cạnh đó, Chương 4 cũng trình bày quá trình xây dựng ứng dụng, kiểm thử và triển khai hệ thống, đồng thời đánh giá mức độ đáp ứng của hệ thống so với các yêu cầu đã đặt ra ban đầu.

Chương 5 tập trung trình bày các đóng góp chính của hệ thống và các công nghệ cốt lõi được sử dụng để xây dựng một hệ thống tuyển dụng thông minh. Nội dung chương này phân tích các giải pháp và ý tưởng trọng tâm tạo nên sự khác biệt của hệ thống so với các hệ thống hiện có, đồng thời làm rõ những kiến thức và phương pháp đã được áp dụng nhằm đề xuất các giải pháp cho bài toán tuyển dụng.

Cuối cùng, Chương 6 trình bày kết quả đánh giá hệ thống thông qua việc so sánh với các hệ thống tuyển dụng tương tự. Chương này tổng hợp các nội dung mà đề án đã thực hiện, chỉ ra những hạn chế còn tồn tại và các đóng góp đạt được, từ đó đề xuất các định hướng và hướng phát triển trong tương lai nhằm tiếp tục hoàn thiện và nâng cao hiệu quả của hệ thống.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Việc khảo sát hiện trạng cho đề tài được thực hiện thông qua việc quan sát và trải nghiệm các nền tảng tuyển dụng trực tuyến hiện có, đồng thời xem xét nhu cầu sử dụng thực tế của người dùng. Quá trình khảo sát tập trung vào cách thức các hệ thống hiện nay hỗ trợ kết nối giữa nhà tuyển dụng và ứng viên, cũng như những chức năng được cung cấp trong quá trình tuyển dụng.

Qua khảo sát các nền tảng tuyển dụng phổ biến như ITviec[1] và TopCV [2], có thể nhận thấy các hệ thống này cung cấp những chức năng cơ bản như đăng tin tuyển dụng, tìm kiếm việc làm theo nhiều tiêu chí và cho phép ứng viên nộp hồ sơ trực tuyến. Một số tính năng hỗ trợ gợi ý việc làm cũng đã được tích hợp, góp phần cải thiện trải nghiệm người dùng so với hình thức tuyển dụng truyền thống. Tuy nhiên, việc đánh giá mức độ phù hợp giữa ứng viên và vị trí tuyển dụng vẫn chủ yếu dựa trên thông tin hồ sơ và tìm kiếm thủ công, trong khi khả năng cá nhân hóa và hỗ trợ ra quyết định còn hạn chế.

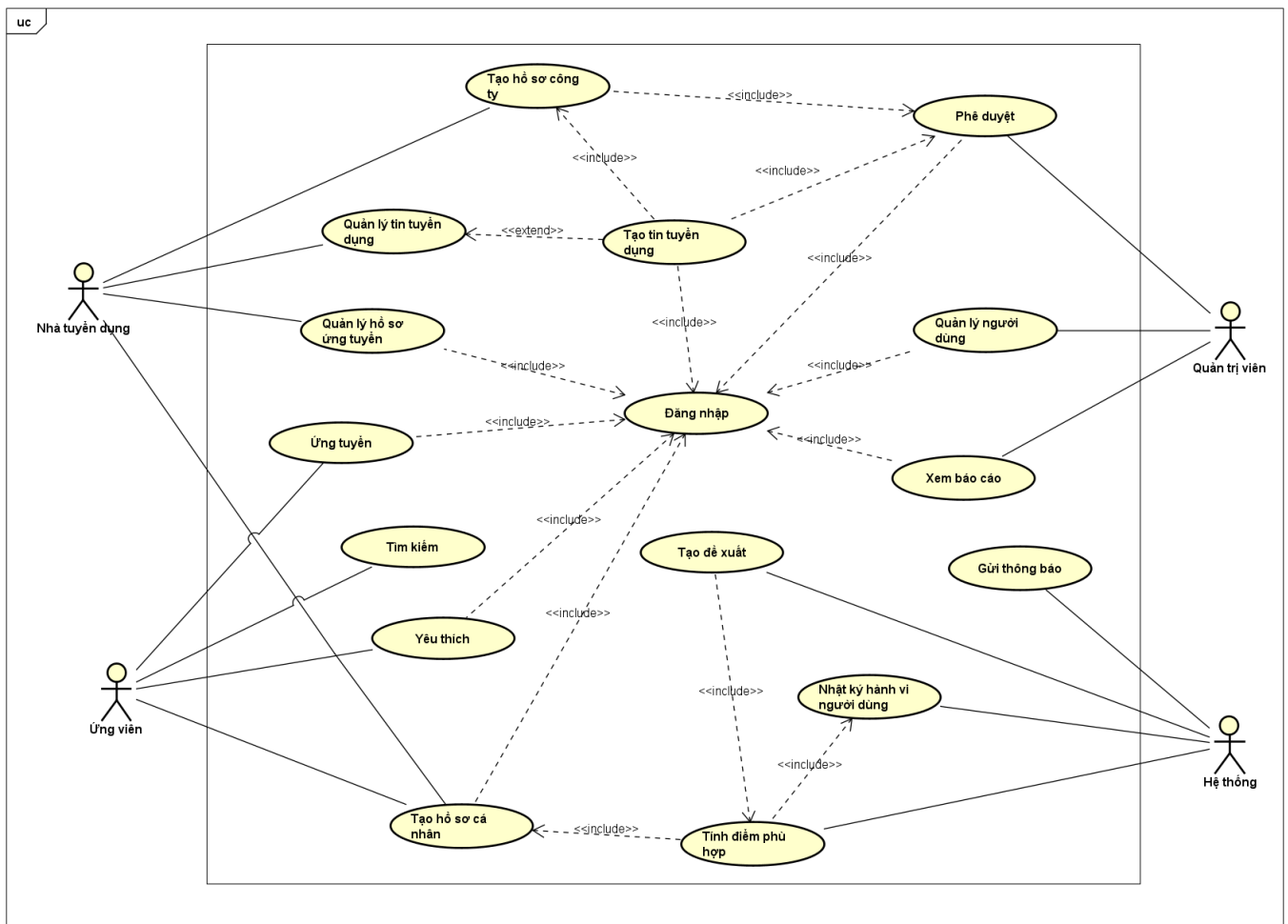
Từ thực tế khảo sát trên, có thể thấy rằng các hệ thống tuyển dụng hiện nay đã đáp ứng được nhu cầu cơ bản nhưng vẫn còn tồn tại những hạn chế nhất định trong việc hỗ trợ kết nối hiệu quả và khai thác dữ liệu người dùng. Trên cơ sở đó, đề tài xác định cần tập trung phát triển các tính năng hỗ trợ tuyển dụng ở mức hiệu quả hơn, làm nền tảng cho việc nâng cao trải nghiệm và hiệu quả sử dụng của hệ thống trong thực tế.

2.2 Tổng quan chức năng

Phần này sẽ tập trung vào trình bày các chức năng cốt lõi của hệ thống như sau:

- (i) Tạo hồ sơ nhu cầu của ứng viên và nhà tuyển dụng.
- (ii) Tạo công ty, tạo tin tuyển dụng.
- (iii) Quản trị viên phê duyệt thông tin.
- (iv) Hệ thống sinh đề xuất cho ứng viên và nhà tuyển dụng.
- (v) Ứng tuyển.

2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát

Biểu đồ use case tổng quát (Hình 2.1) mô tả các chức năng mức cao của hệ thống tuyển dụng và vai trò của các tác nhân đối với hệ thống. Hệ thống bao gồm bốn tác nhân chính là Ứng viên, Nhà tuyển dụng, Quản trị viên và Hệ thống.

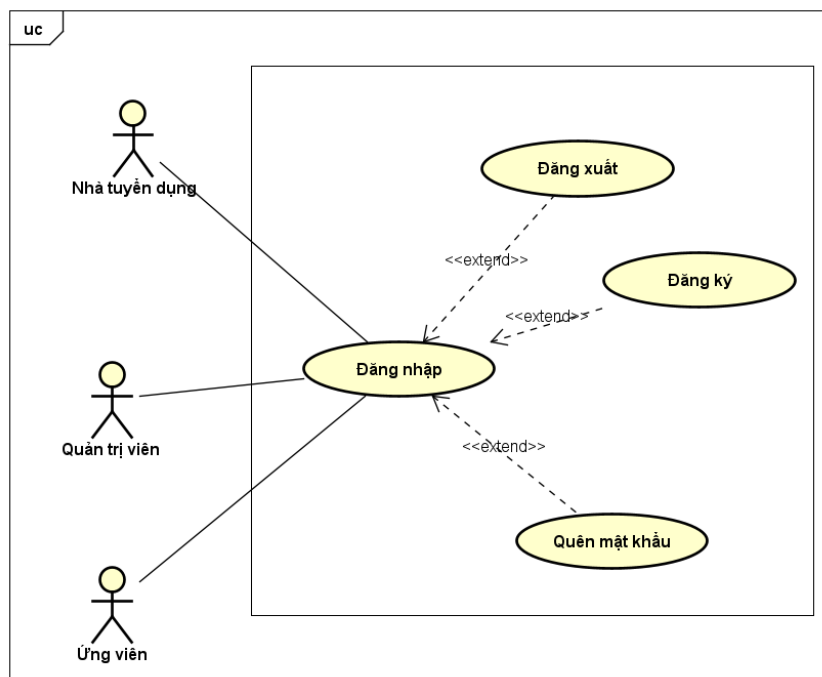
Ứng viên là tác nhân thực hiện các nghiệp vụ liên quan đến tìm kiếm và ứng tuyển công việc, bao gồm tạo hồ sơ cá nhân, tìm kiếm tin tuyển dụng, yêu thích tin tuyển dụng và gửi đơn ứng tuyển. Các chức năng này đều yêu cầu người dùng phải thực hiện đăng nhập trước khi tương tác với hệ thống. Trong tạo hồ sơ cá nhân, ứng viên sẽ cung cấp các thông tin về mong muốn nghề nghiệp cho chức năng đề xuất của hệ thống.

Nhà tuyển dụng thực hiện các nghiệp vụ quản lý hoạt động tuyển dụng, bao gồm tạo hồ sơ công ty, tạo và quản lý tin tuyển dụng, quản lý hồ sơ ứng tuyển của ứng viên. Hệ thống bắt buộc xác thực công ty trước khi có thể đăng tin tuyển dụng. Nhà tuyển dụng phải đăng nhập vào tài khoản có vai trò tuyển dụng và cũng cung cấp nhu cầu tuyển dụng cho hệ thống đề xuất.

Quản trị viên chịu trách nhiệm quản lý toàn bộ hệ thống, bao gồm quản lý người dùng, phê duyệt công ty, phê duyệt tin tuyển dụng và theo dõi các báo cáo tổng hợp. Quản trị viên cũng phải đăng nhập vào hệ thống bằng tài khoản có vai trò quản trị.

Hệ thống là tác nhân bên trong thực hiện các chức năng đề xuất bao gồm lưu nhật ký hành vi người dùng, tính toán điểm phù hợp hai chiều giữa ứng viên và nhà tuyển dụng, tạo các đề xuất phù hợp và gửi thông báo đến các tác nhân liên quan. Các chức năng tạo đề xuất, tính điểm phù hợp cần có dữ liệu từ ứng viên và nhà tuyển dụng.

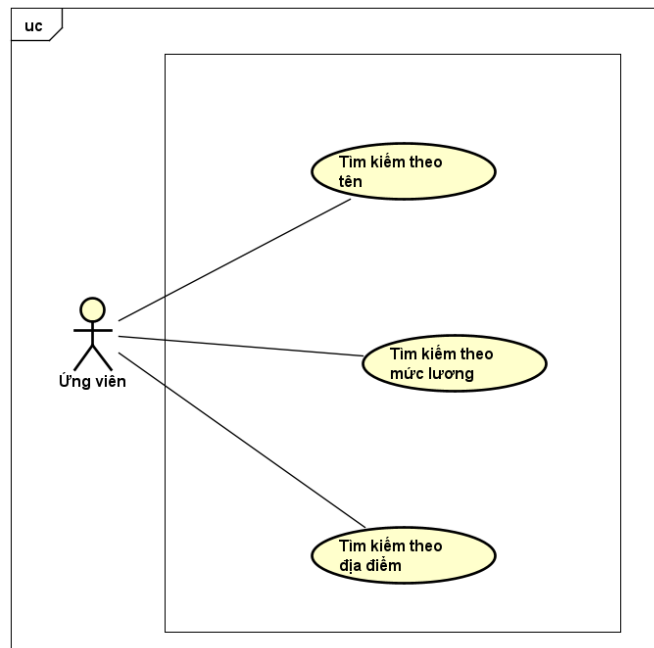
2.2.2 Biểu đồ use case phân rã đăng nhập



Hình 2.2: Biểu đồ use case phân rã đăng nhập

Đăng nhập là chức năng bắt buộc khi người dùng muốn sử dụng toàn bộ chức năng của hệ thống.

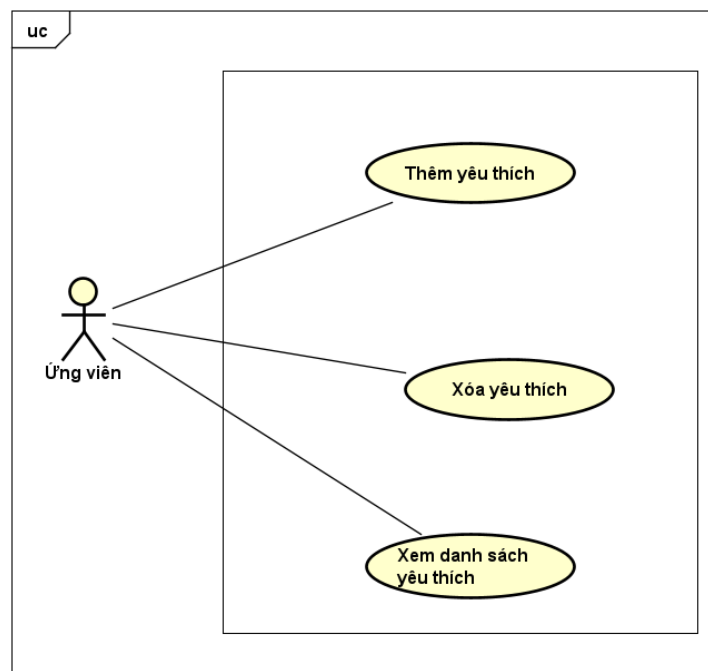
2.2.3 Biểu đồ use case phân rã tìm kiếm



Hình 2.3: Biểu đồ use case phân rã tìm kiếm

Chức năng tìm kiếm với các bộ lọc kèm theo khi tìm kiếm và có thể truy cập không cần đăng nhập.

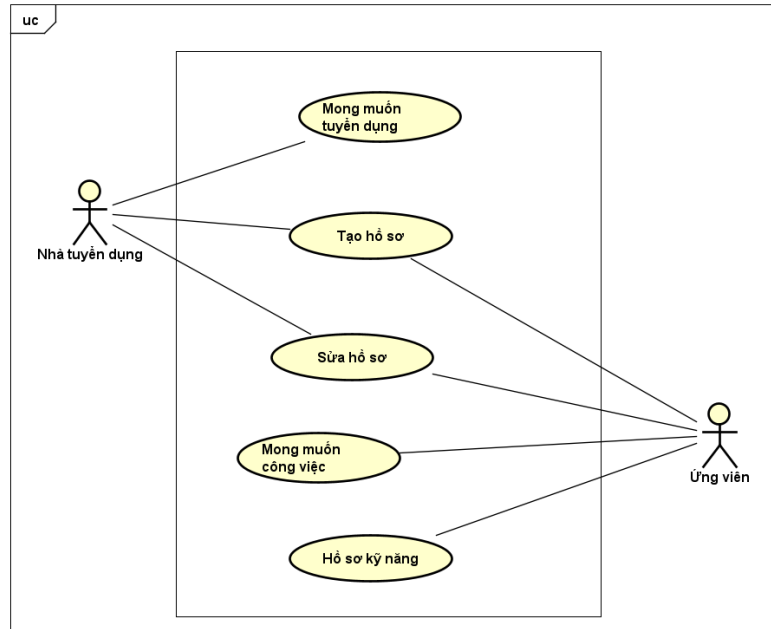
2.2.4 Biểu đồ use case phân rã yêu thích



Hình 2.4: Biểu đồ use case phân rã yêu thích

Người dùng yêu thích, quản lý yêu thích và hệ thống ghi nhận hành vi của người dùng.

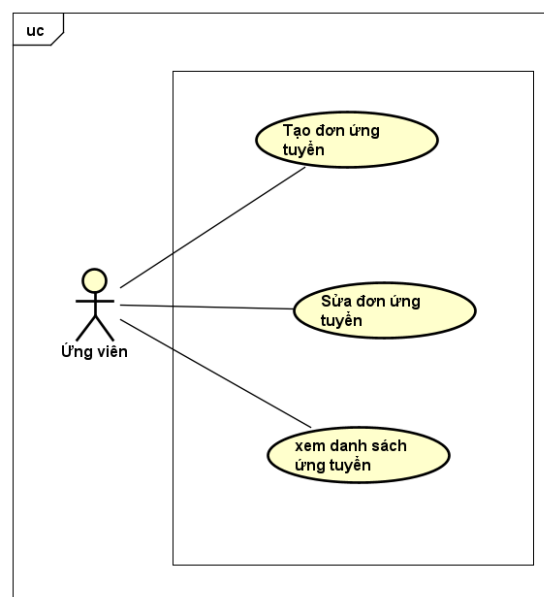
2.2.5 Biểu đồ use case phân rã tạo hồ sơ cá nhân



Hình 2.5: Biểu đồ use case phân rã tạo hồ sơ cá nhân

Hồ sơ cá nhân bao gồm các thông tin cơ bản và các mong muốn của người dùng tương ứng.

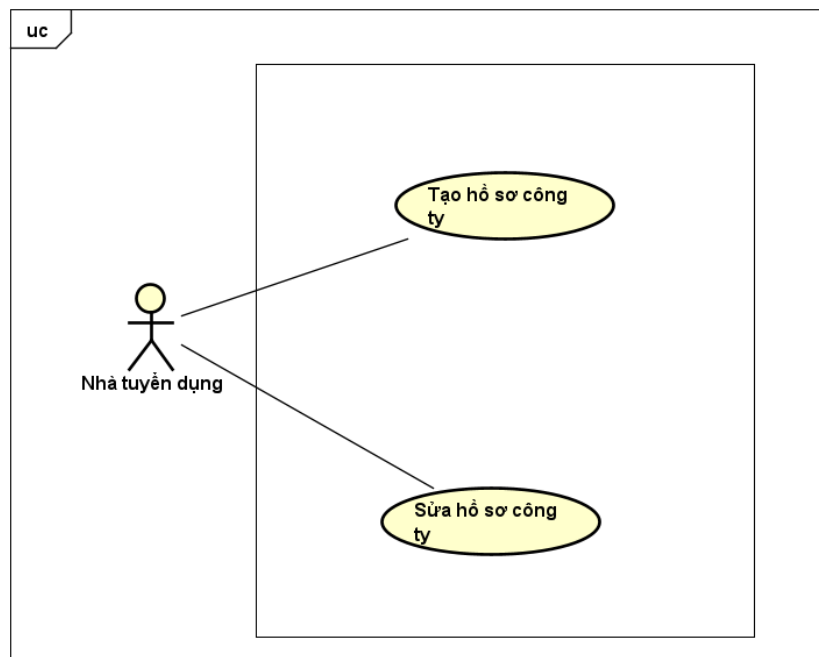
2.2.6 Biểu đồ use case phân rã ứng tuyển



Hình 2.6: Biểu đồ use case phân rã ứng tuyển

Ứng viên ứng tuyển và quản lý các đơn ứng tuyển của mình.

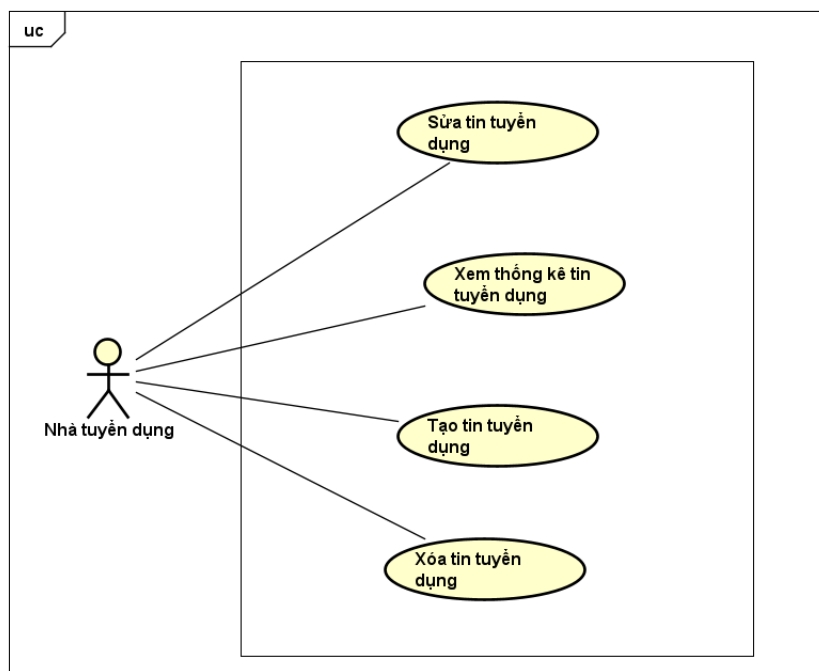
2.2.7 Biểu đồ use case phân rã tạo công ty



Hình 2.7: Biểu đồ use case phân rã tạo công ty

Nhà tuyển dụng tạo công ty trước khi được đăng tin tuyển dụng.

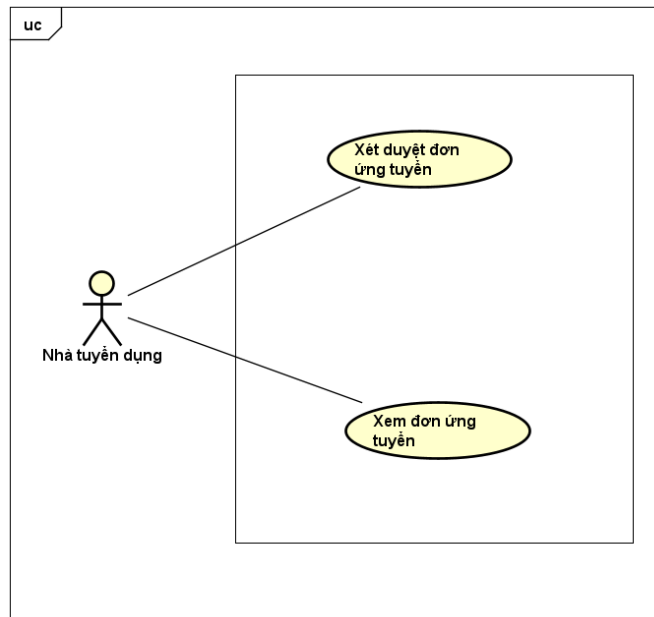
2.2.8 Biểu đồ use case phân rã quản lý tin tuyển dụng



Hình 2.8: Biểu đồ use case phân rã quản lý tin tuyển dụng

Nhà tuyển dụng tạo và quản lý tin tuyển dụng.

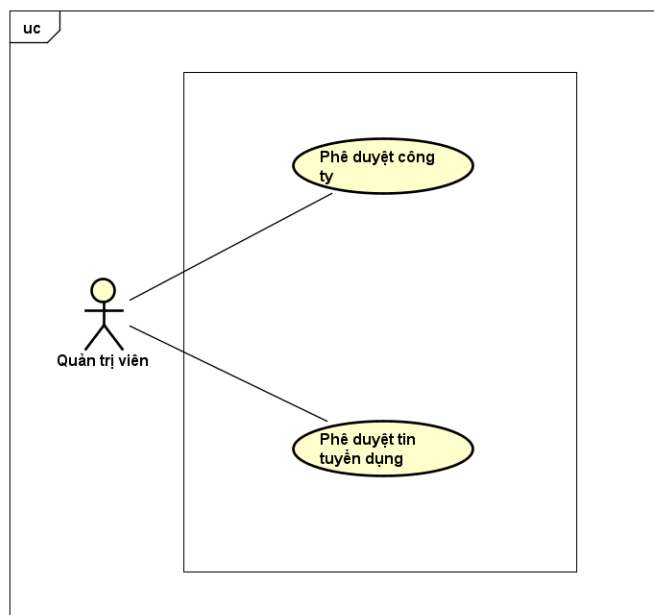
2.2.9 Biểu đồ use case phân rã quản lý hồ sơ ứng tuyển



Hình 2.9: Biểu đồ use case phân rã quản lý hồ sơ ứng tuyển

Nhà tuyển dụng xem xét và quản lý đơn ứng tuyển của ứng viên.

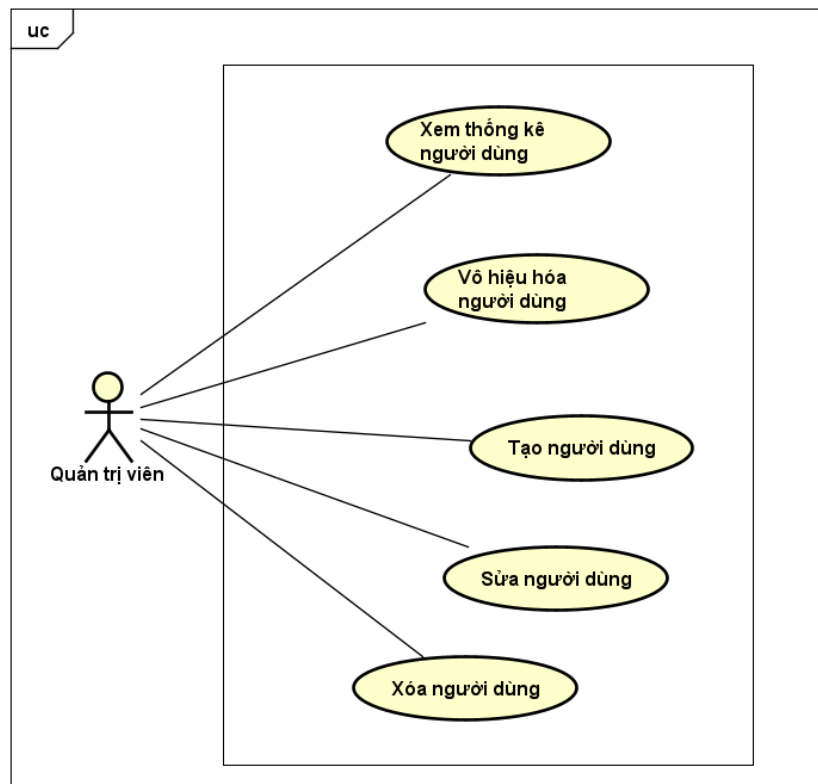
2.2.10 Biểu đồ use case phân rã phê duyệt



Hình 2.10: Biểu đồ use case phân rã phê duyệt

Quản trị viên xét duyệt công ty và các tin tuyển dụng.

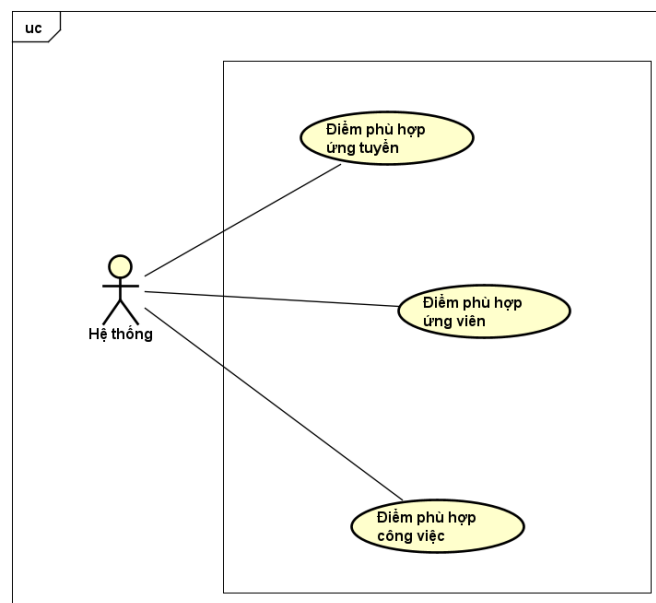
2.2.11 Biểu đồ use case phân rã quản lý người dùng



Hình 2.11: Biểu đồ use case phân rã quản lý người dùng

Quản trị viên quản lý tất cả người dùng của hệ thống.

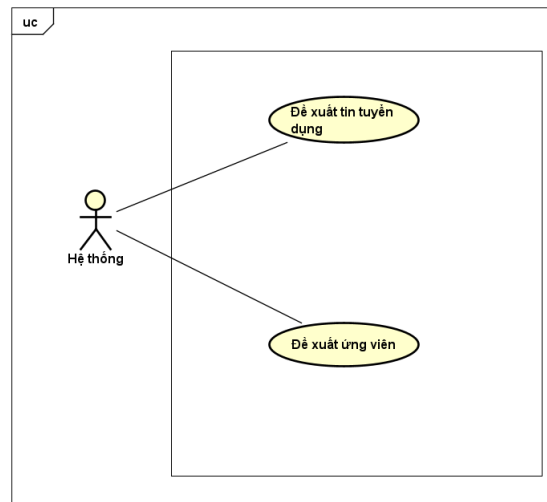
2.2.12 Biểu đồ use case phân rã tính điểm phù hợp



Hình 2.12: Biểu đồ use case phân rã tính điểm phù hợp

Hệ thống lấy dữ liệu nhu cầu, mong muốn để tính điểm phù hợp cho 3 nhóm chính: Điểm phù hợp của ứng viên cho 1 công việc đã ứng tuyển, điểm phù hợp của ứng viên với nhu cầu của nhà tuyển dụng, điểm phù hợp của công việc cho ứng viên.

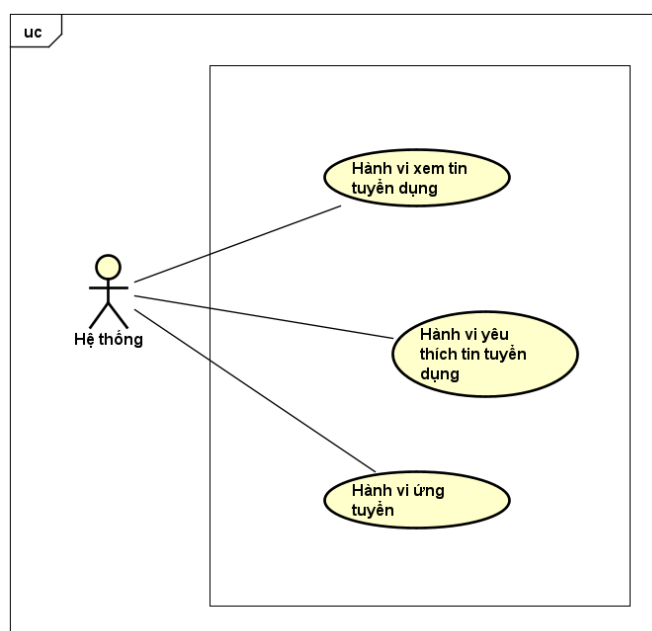
2.2.13 Biểu đồ use case phân rã tạo đề xuất



Hình 2.13: Biểu đồ use case phân rã tạo đề xuất

Hệ thống tạo đề xuất tin tuyển dụng cho ứng viên và đề xuất ứng viên cho nhà tuyển dụng.

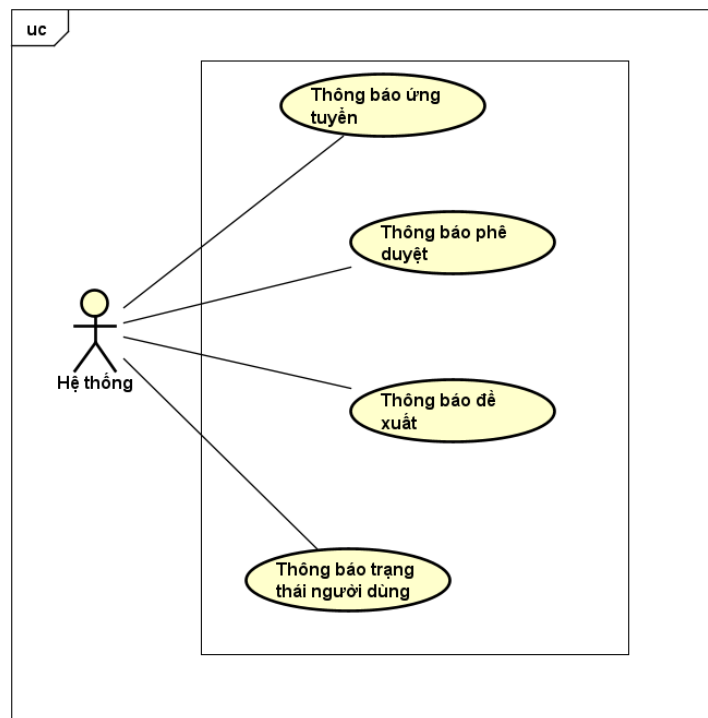
2.2.14 Biểu đồ use case phân rã nhật ký hành vi



Hình 2.14: Biểu đồ use case phân rã nhật ký hành vi

Hệ thống ghi nhận 3 hành vi chính của ứng viên: ứng viên xem một tin tuyển dụng, yêu thích một tin tuyển dụng và hành vi nộp đơn ứng tuyển.

2.2.15 Biểu đồ use case phân rã gửi thông báo



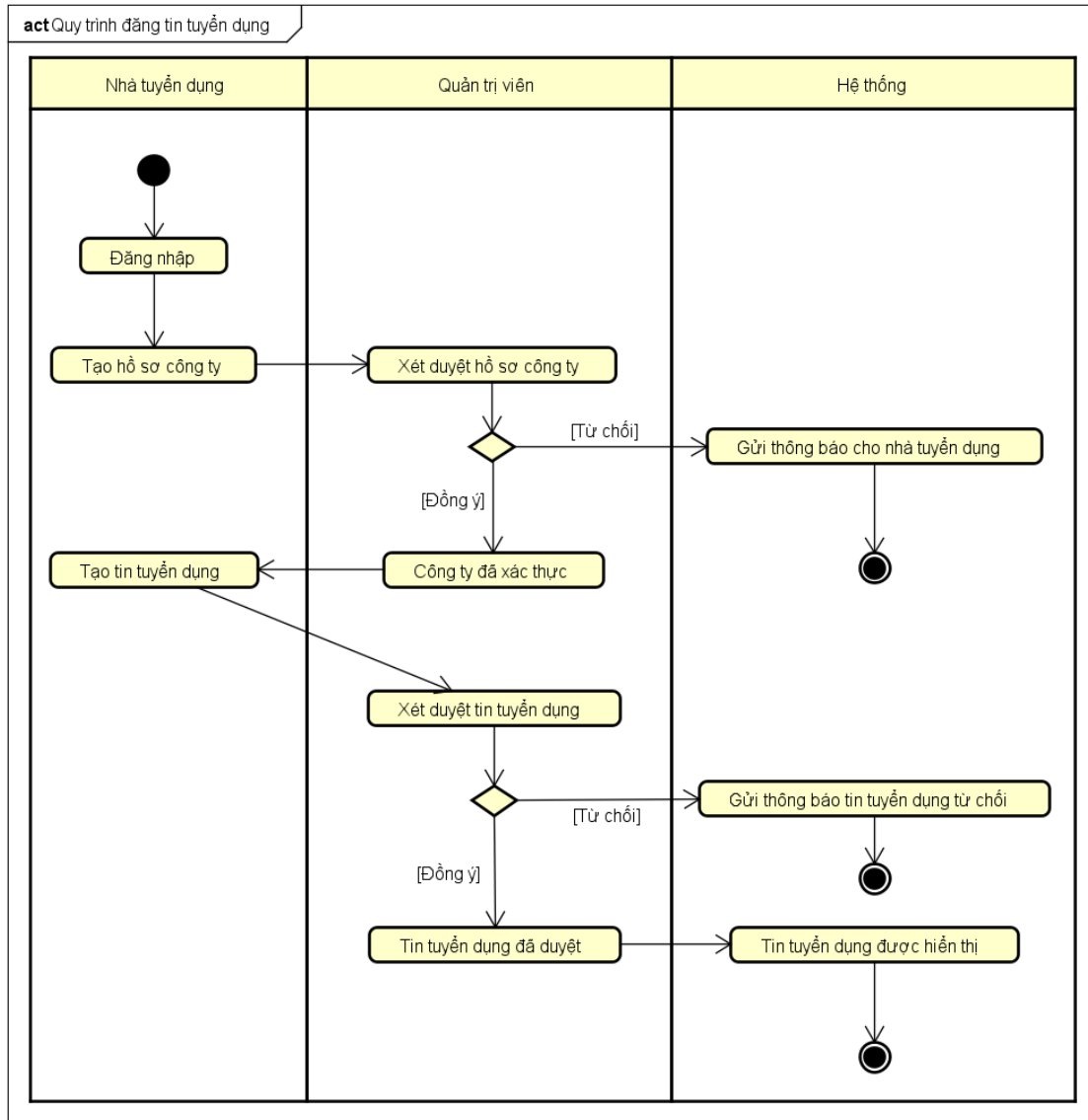
Hình 2.15: Biểu đồ use case phân rã gửi thông báo

Hệ thống gửi lấy dữ liệu đề xuất và gửi thông báo định kỳ cho người dùng tương ứng. Hệ thống gửi thông báo khi có sự kiện xảy ra.

2.2.16 Quy trình nghiệp vụ

Hệ thống tuyển dụng thông minh có các quy trình nghiệp vụ quan trọng liên quan đến tin tuyển dụng, người dùng và hệ thống đề xuất. Dưới đây là các biểu đồ hoạt động chính của các quy trình nghiệp vụ đó.

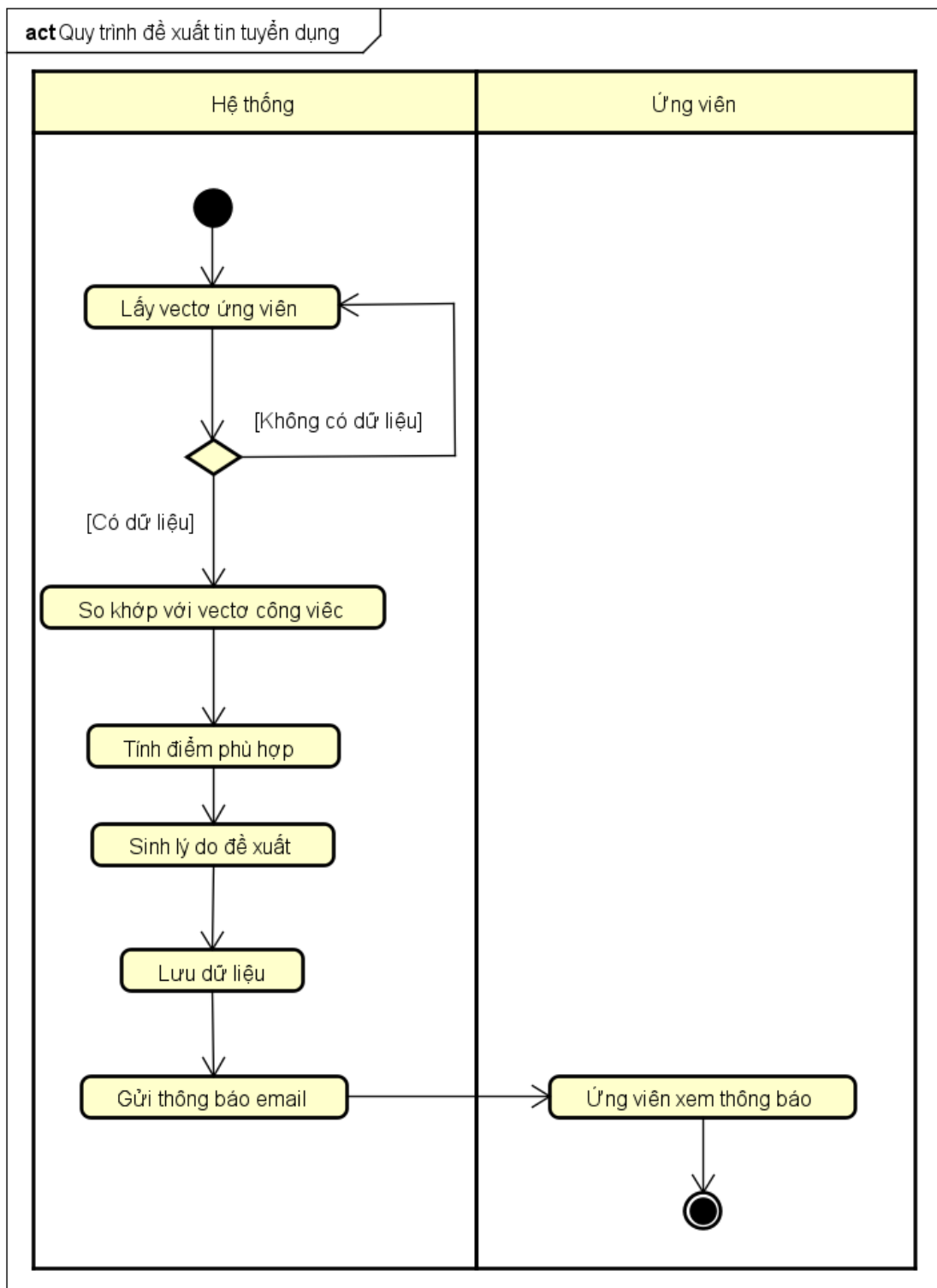
a, Quy trình đăng tin tuyển dụng



Hình 2.16: Biểu đồ hoạt động quy trình đăng tin tuyển dụng

Biểu đồ hoạt động (Hình 2.16) mô tả quy trình đăng tin tuyển dụng, nhà tuyển dụng đăng nhập, tạo công ty và đăng tin tuyển dụng. Các bước đều cần xét duyệt từ phía quản trị viên và hệ thống sẽ gửi thông báo tăng trải nghiệm người dùng.

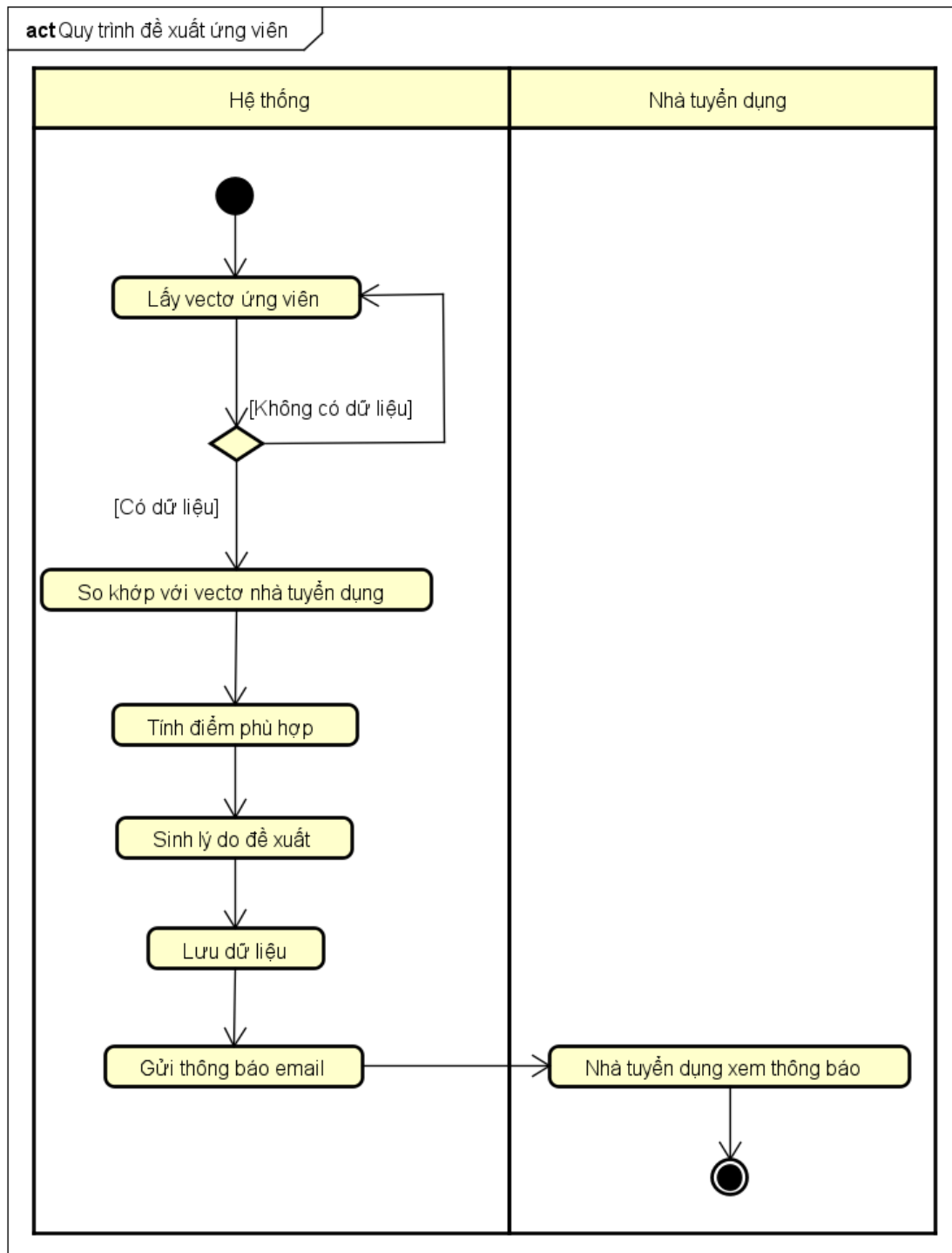
b, Quy trình đề xuất tin tuyển dụng



Hình 2.17: Biểu đồ hoạt động quy trình đề xuất tin tuyển dụng

Biểu đồ hoạt động (Hình 2.17) thể hiện quy trình hệ thống tự động đề xuất tin tuyển dụng cho ứng viên dựa trên việc so khớp vectơ đặc trưng, tính toán điểm phù hợp và gửi thông báo đến ứng viên.

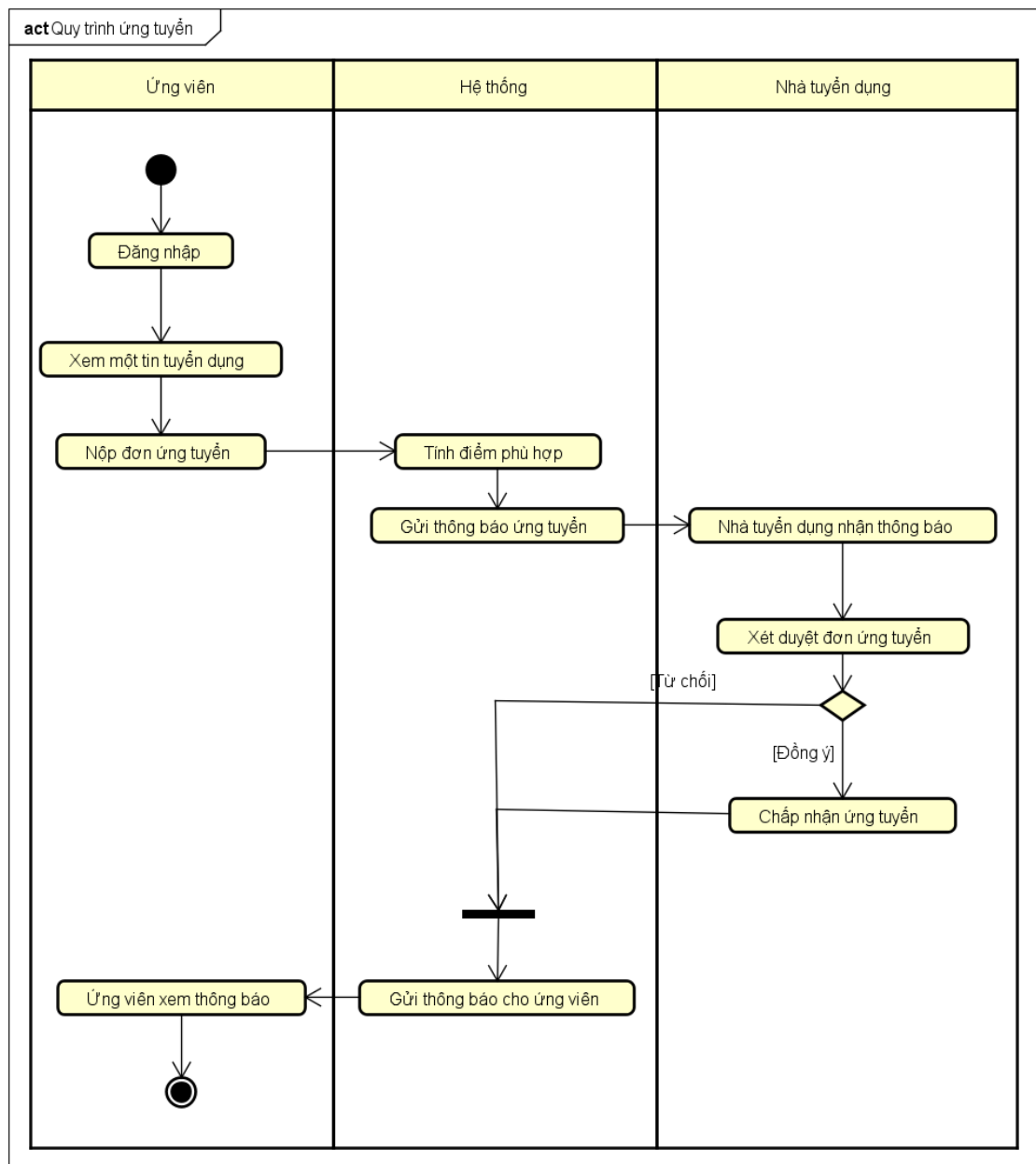
c, Quy trình đề xuất ứng viên



Hình 2.18: Biểu đồ hoạt động quy trình đề xuất ứng viên

Biểu đồ hoạt động (Hình 2.18) mô tả quy trình hệ thống đề xuất ứng viên phù hợp cho nhà tuyển dụng thông qua việc so khớp vectơ đặc trưng (giống quy trình đề xuất tin tuyển dụng), tính toán điểm phù hợp và gửi thông báo đến nhà tuyển dụng.

d, Quy trình ứng tuyển



Hình 2.19: Biểu đồ hoạt động quy trình ứng tuyển

Biểu đồ hoạt động (Hình 2.19) mô tả quy trình ứng tuyển từ phía ứng viên, bao gồm các bước đăng nhập, xem tin tuyển dụng, nộp đơn ứng tuyển, tính điểm phù hợp và xét duyệt đơn ứng tuyển từ phía nhà tuyển dụng.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case đăng nhập

Bảng 2.1: Bảng đặc tả use case “Đăng nhập”

Mã use case	UC01	Tên use case	Đăng nhập
Tác nhân	Người dùng		
Mô tả	Người dùng (Ứng viên/Nhà tuyển dụng/Quản trị viên) đăng nhập vào hệ thống		
Tiền điều kiện	Người dùng đã đăng ký tài khoản trên hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Người dùng xác thực tài khoản qua email
	2	Hệ thống	Hiển thị giao diện xác thực thành công
	3	Người dùng	Vào trang đăng nhập
	4	Người dùng	Nhập thông tin tài khoản và mật khẩu
	5	Hệ thống	Xác thực thông tin và kiểm tra vai trò người dùng
	6	Hệ thống	Truy cập giao diện theo vai trò người dùng
	STT	Thực hiện bởi	Hành động
Luồng sự kiện thay thế	1a	Người dùng	Không tìm thấy mail và chọn chức năng quên mật khẩu
	2a	Hệ thống	Thông báo lỗi: Token đã hết hạn!
	⁴ 5a	Hệ thống	Thông báo lỗi: Email hoặc mật khẩu không đúng!
	5b	Hệ thống	Thông báo lỗi: Tài khoản chưa được xác thực qua email!

Hậu điều kiện	Người dùng đăng nhập thành công và sử dụng các chức năng cá nhân hóa
----------------------	--

2.3.2 Đặc tả use case mong muốn công việc

Bảng 2.2: Bảng đặc tả use case “Mong muốn công việc”

Mã use case	UC02	Tên use case	Mong muốn công việc
Tác nhân	Ứng viên		
Mô tả	Ứng viên tạo và cập nhật hồ sơ mong muốn công việc nhằm phục vụ cho hệ thống đề xuất việc làm		
Tiền điều kiện	Ứng viên đã đăng nhập vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Ứng viên	Đăng nhập tài khoản có vai trò ứng viên
	2	Hệ thống	Hiển thị giao diện chính
	3	Ứng viên	Truy cập giao diện hồ sơ cá nhân (Profile)
	4	Ứng viên	Nhấn vào chức năng cập nhật mong muốn nghề nghiệp
	5	Hệ thống	Hiển thị cửa sổ (popup) nhập thông tin mong muốn nghề nghiệp
	6	Ứng viên	Nhập các thông tin cần thiết và thực hiện lưu dữ liệu
	7	Hệ thống	Hiển thị thông báo: Đã lưu mong muốn nghề nghiệp
	STT	Thực hiện bởi	Hành động

Luồng sự kiện thay thế	4a	Hệ thống	Tài khoản không có vai trò ứng viên, hệ thống không hiển thị chức năng nhập mong muốn nghề nghiệp
	7a	Hệ thống	Hiển thị thông báo lỗi: Không thể lưu mong muốn nghề nghiệp
Hậu điều kiện	Hệ thống lưu lại thông tin mong muốn nghề nghiệp của ứng viên để phục vụ chức năng đề xuất		

2.3.3 Đặc tả use case mong muốn tuyển dụng

Bảng 2.3: Bảng đặc tả use case “Mong muốn tuyển dụng”

Mã use case	UC03	Tên use case	Mong muốn tuyển dụng
Tác nhân	Nhà tuyển dụng		
Mô tả	Nhà tuyển dụng tạo và cập nhật hồ sơ nhu cầu tuyển dụng nhằm phục vụ cho hệ thống đề xuất ứng viên		
Tiền điều kiện	Nhà tuyển dụng đã đăng nhập vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Nhà tuyển dụng	Đăng nhập tài khoản có vai trò nhà tuyển dụng
	2	Hệ thống	Hiển thị giao diện nhà tuyển dụng
	3	Nhà tuyển dụng	Truy cập giao diện hồ sơ cá nhân (Profile)
	4	Nhà tuyển dụng	Nhấn vào chức năng cập nhật nhu cầu tuyển dụng
	5	Hệ thống	Hiển thị cửa sổ (popup) cập nhật nhu cầu tuyển dụng

	6	Nhà tuyển dụng	Nhập các thông tin cần thiết và thực hiện lưu dữ liệu
	7	Hệ thống	Hiển thị thông báo: Đã cập nhật nhu cầu tuyển dụng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Tài khoản không có vai trò nhà tuyển dụng, hệ thống không hiển thị chức năng nhập nhu cầu tuyển dụng
	7a	Hệ thống	Hiển thị thông báo lỗi: Không thể cập nhật nhu cầu tuyển dụng
Hậu điều kiện	Hệ thống lưu lại thông tin nhu cầu tuyển dụng của nhà tuyển dụng để phục vụ chức năng đề xuất ứng viên		

2.3.4 Đặc tả use case tạo công ty

Bảng 2.4: Bảng đặc tả use case “Tạo công ty”

Mã use case	UC04	Tên use case	Tạo công ty
Tác nhân	Nhà tuyển dụng		
Mô tả	Nhà tuyển dụng tạo công ty và gửi cho quản trị viên duyệt trước khi có thể đăng tin tuyển dụng		
Tiền điều kiện	Nhà tuyển dụng đã đăng nhập vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Nhà tuyển dụng	Đăng nhập tài khoản có vai trò nhà tuyển dụng
	2	Hệ thống	Hiển thị giao diện nhà tuyển dụng
	3	Nhà tuyển dụng	Truy cập tab <i>Thông tin công ty</i>

	4	Nhà tuyển dụng	Nhấn vào chức năng tạo công ty
	5	Hệ thống	Hiển thị cửa sổ (popup) tạo công ty
	6	Nhà tuyển dụng	Nhập các thông tin cần thiết và thực hiện lưu dữ liệu
	7	Hệ thống	Hiển thị thông báo: Tạo công ty thành công!
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Tài khoản không có vai trò nhà tuyển dụng, hệ thống không hiển thị giao diện nhà tuyển dụng
	4a	Hệ thống	Trường hợp công ty đã tồn tại, hệ thống hiển thị giao diện cập nhật thông tin công ty
	7a	Hệ thống	Hiển thị thông báo lỗi: Không thể tạo công ty!
Hậu điều kiện	Hệ thống lưu lại thông tin công ty và thiết lập trạng thái công ty là <i>submitted</i>		

2.3.5 Đặc tả use case tạo tin tuyển dụng

Bảng 2.5: Bảng đặc tả use case “Tạo tin tuyển dụng”

Mã use case	UC05	Tên use case	Tạo tin tuyển dụng
Tác nhân	Nhà tuyển dụng		
Mô tả	Nhà tuyển dụng tạo và đăng tin tuyển dụng để tìm kiếm ứng viên trên hệ thống		
Tiền điều kiện	Nhà tuyển dụng đã đăng nhập và công ty đã được xác thực		

Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Nhà tuyển dụng	Đăng nhập tài khoản có vai trò nhà tuyển dụng
	2	Hệ thống	Hiển thị giao diện nhà tuyển dụng
	3	Nhà tuyển dụng	Truy cập tab <i>Bài đăng của tôi</i>
	4	Nhà tuyển dụng	Tìm và chọn chức năng tạo mới tại mục bài đăng tuyển dụng
	5	Hệ thống	Hiển thị cửa sổ (popup) <i>Thêm tin tuyển dụng</i>
	6	Nhà tuyển dụng	Nhập các thông tin cần thiết và thực hiện lưu dữ liệu
	7	Hệ thống	Kiểm tra tính đầy đủ của các trường thông tin bắt buộc
	8	Hệ thống	Hiển thị thông báo: Tạo tin tuyển dụng thành công!
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Tài khoản không có vai trò nhà tuyển dụng, hệ thống không hiển thị giao diện nhà tuyển dụng
	4a	Hệ thống	Trường hợp công ty chưa được xác thực, hệ thống hiển thị giao diện cập nhật thông tin công ty
	7a	Hệ thống	Hiển thị thông báo lỗi: <i>Please fill out this field.</i>

Hậu điều kiện	Hệ thống lưu lại tin tuyển dụng và thiết lập trạng thái tin tuyển dụng là <i>pending</i>
----------------------	--

2.3.6 Đặc tả use case phê duyệt tin tuyển dụng

Bảng 2.6: Bảng đặc tả use case “Phê duyệt tin tuyển dụng”

Mã use case	UC06	Tên use case	Phê duyệt tin tuyển dụng
Tác nhân	Quản trị viên		
Mô tả	Quản trị viên xem xét và phê duyệt tin tuyển dụng trước khi hiển thị công khai trên hệ thống		
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Quản trị viên	Đăng nhập tài khoản có vai trò quản trị viên
	2	Hệ thống	Hiển thị giao diện quản trị viên
	3	Quản trị viên	Truy cập tab <i>Danh sách bài đăng</i>
	4	Quản trị viên	Chọn một tin tuyển dụng cần phê duyệt
	5	Hệ thống	Hiển thị cửa sổ (popup) phê duyệt tin tuyển dụng
	6	Quản trị viên	Chọn chức năng phê duyệt
	7	Hệ thống	Hiển thị cửa sổ xác nhận phê duyệt
	8	Quản trị viên	Xác nhận phê duyệt tin tuyển dụng
	9	Hệ thống	Gửi email thông báo kết quả phê duyệt cho nhà tuyển dụng

Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Tài khoản không có vai trò quản trị viên, hệ thống không hiển thị giao diện quản trị viên
	6a	Quản trị viên	Chọn chức năng từ chối phê duyệt
	7a	Hệ thống	Hiển thị cửa sổ (popup) từ chối phê duyệt
	8a	Quản trị viên	Nhập lý do từ chối và xác nhận thao tác
Hậu điều kiện	Tin tuyển dụng được chuyển sang trạng thái <i>approved</i> và sẵn sàng hiển thị trên hệ thống		

2.3.7 Đặc tả use case tạo đề xuất tin tuyển dụng

Bảng 2.7: Bảng đặc tả use case “Tạo đề xuất tin tuyển dụng”

Mã use case	UC07	Tên use case	Tạo đề xuất tin tuyển dụng
Tác nhân	Hệ thống		
Mô tả	Hệ thống tự động tạo danh sách đề xuất tin tuyển dụng phù hợp cho ứng viên dựa trên mô hình so khớp vectơ		
Tiền điều kiện	Hệ thống đã có vectơ đặc trưng của công việc và vectơ đặc trưng của ứng viên		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Hệ thống	Lấy thông tin vectơ đặc trưng của công việc và ứng viên
	2	Hệ thống	Thực hiện hàm tính toán độ phù hợp của công việc đối với từng ứng viên

	3	Hệ thống	Lưu kết quả đề xuất (điểm phù hợp và lý do) vào bảng <i>Job Recommendation</i>
	4	Hệ thống	Lấy danh sách đề xuất theo thứ tự điểm phù hợp giảm dần
	5	Hệ thống	Hiển thị danh sách đề xuất cho ứng viên và gửi email thông báo tự động lúc 8 giờ sáng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	1a	Hệ thống	Không tìm thấy <i>user_vector</i> , hệ thống hiển thị thông báo lỗi và dừng quá trình đề xuất
Hậu điều kiện	Danh sách đề xuất tin tuyển dụng được hiển thị thành công và email thông báo được gửi tới ứng viên		

2.3.8 Đặc tả use case tạo đề xuất ứng viên

Bảng 2.8: Bảng đặc tả use case “Tạo đề xuất ứng viên”

Mã use case	UC08	Tên use case	Tạo đề xuất ứng viên
Tác nhân	Hệ thống		
Mô tả	Hệ thống tự động tạo danh sách đề xuất ứng viên phù hợp cho nhà tuyển dụng dựa trên mô hình so khớp vectơ		
Tiền điều kiện	Hệ thống đã có vectơ đặc trưng của ứng viên (<i>user_vector</i>) và vectơ đặc trưng của nhà tuyển dụng (<i>recruiter_vector</i>)		
	STT	Thực hiện bởi	Hành động
	1	Hệ thống	Lấy thông tin vectơ đặc trưng của ứng viên và nhà tuyển dụng

Luồng sự kiện chính	2	Hệ thống	Thực hiện hàm tính toán độ phù hợp giữa ứng viên và nhà tuyển dụng
	3	Hệ thống	Lưu kết quả đề xuất (điểm phù hợp và lý do) vào bảng <i>CandidateRecommendation</i>
	4	Hệ thống	Lấy danh sách đề xuất theo thứ tự điểm phù hợp giảm dần
	5	Hệ thống	Hiển thị danh sách đề xuất cho nhà tuyển dụng và gửi email thông báo tự động lúc 9 giờ sáng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	1a	Hệ thống	Không tìm thấy <i>Recruiter-Preference</i> cho người dùng này, hệ thống hiển thị thông báo lỗi và dừng quá trình đề xuất
Hậu điều kiện	Danh sách đề xuất ứng viên được hiển thị thành công và email thông báo được gửi tới nhà tuyển dụng		

2.3.9 Đặc tả use case ứng tuyển

Bảng 2.9: Bảng đặc tả use case “Ứng tuyển”

Mã use case	UC09	Tên use case	Ứng tuyển
Tác nhân	Ứng viên		
Mô tả	Ứng viên tạo đơn ứng tuyển vào một tin tuyển dụng trên hệ thống		
Tiền điều kiện	Ứng viên đã đăng nhập vào hệ thống		
	STT	Thực hiện bởi	Hành động

Luồng sự kiện chính	1	Ứng viên	Đăng nhập tài khoản có vai trò ứng viên
	2	Hệ thống	Hiển thị giao diện chính
	3	Ứng viên	Chọn một tin tuyển dụng phù hợp
	4	Ứng viên	Chọn chức năng <i>Ứng tuyển ngay</i>
	5	Hệ thống	Hiển thị cửa sổ (popup) ứng tuyển công việc
	6	Ứng viên	Nhập các thông tin cần thiết và xác nhận ứng tuyển
	7	Hệ thống	Tính điểm phù hợp của ứng viên với công việc và lưu đơn ứng tuyển vào cơ sở dữ liệu
	8	Hệ thống	Hiển thị thông báo: Ứng tuyển thành công!
	9	Hệ thống	Gửi email thông báo có ứng viên ứng tuyển cho nhà tuyển dụng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	7a	Hệ thống	Hiển thị thông báo lỗi: Vui lòng nhập thư ứng tuyển
	7b	Hệ thống	Hiển thị thông báo lỗi: Vui lòng tải lên CV
Hậu điều kiện	Ứng viên có thể xem lại đơn ứng tuyển của mình tại trang cá nhân		

2.4 Yêu cầu phi chức năng

Đối với hệ thống tuyển dụng thông minh, hệ thống cần đảm bảo mức độ bảo mật cao, thực hiện phân quyền rõ ràng theo vai trò người dùng và hiển thị giao diện phù hợp với từng chức năng, quyền hạn tương ứng. Mã nguồn và thiết kế cấu trúc

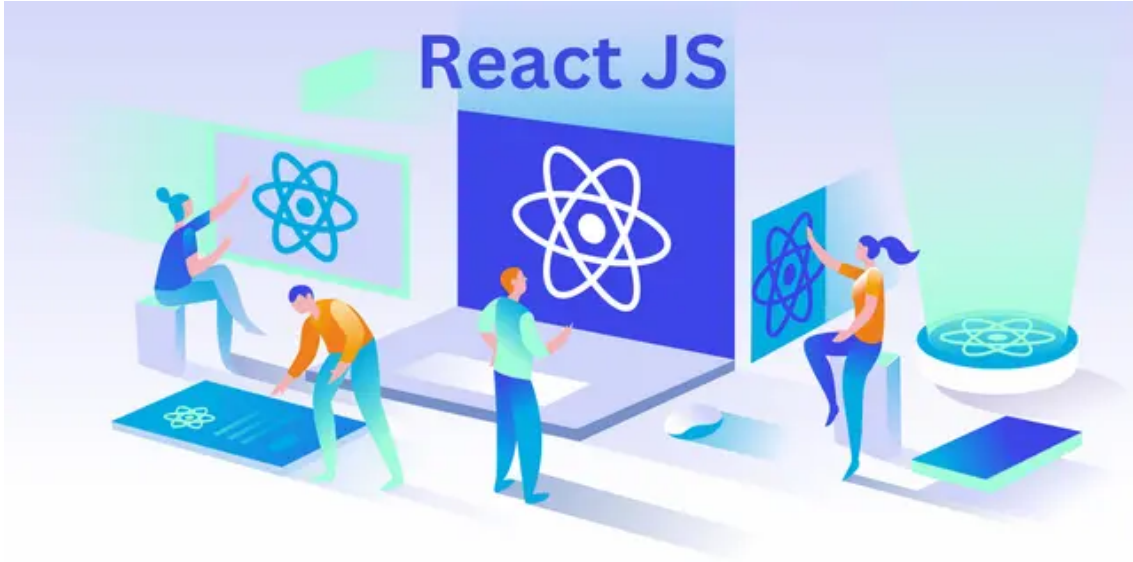
thư mục phải được tổ chức một cách khoa học, tuân theo các nguyên tắc thiết kế (design pattern) [3] ở mức tổng thể, nhằm nâng cao khả năng bảo trì, mở rộng và phát triển hệ thống trong tương lai.

Bên cạnh đó, giao diện hệ thống cần được thiết kế gọn gàng, trực quan, hạn chế thông tin dư thừa và tập trung vào các chức năng cốt lõi, qua đó nâng cao trải nghiệm người dùng. Cơ sở dữ liệu phải áp dụng các cơ chế mã hóa và bảo mật đối với những thông tin quan trọng, đặc biệt là dữ liệu nhạy cảm như mật khẩu người dùng.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 ReactTS

3.1.1 React



Hình 3.1: Minh họa thư viện React

React là một thư viện JavaScript mã nguồn mở do Meta phát triển, được thiết kế để xây dựng giao diện người dùng theo hướng khai báo và dựa trên các thành phần (component). React cho phép chia nhỏ giao diện thành các khối chức năng độc lập, có thể tái sử dụng và dễ kiểm soát luồng dữ liệu, từ đó phù hợp với các ứng dụng web có nhiều màn hình và tương tác thường xuyên. [4]

Trong đồ án, React được sử dụng để đáp ứng các yêu cầu ở Chương 4 liên quan đến tầng giao diện và trải nghiệm người dùng, bao gồm hiển thị danh sách tin tuyển dụng, trang chi tiết tin, trang quản lý hồ sơ ứng viên/nhà tuyển dụng, luồng ứng tuyển, và đặc biệt là màn hình hiển thị danh sách đề xuất dựa trên FitScore. Với đặc thù hệ thống có nhiều thao tác tìm kiếm, phân trang, lọc dữ liệu và cập nhật danh sách theo truy vấn, React tỏ ra phù hợp nhờ mô hình component giúp tách UI theo từng chức năng rõ ràng, đồng thời cơ chế quản lý trạng thái hỗ trợ cập nhật giao diện nhất quán theo hành vi người dùng. Cách tổ chức này cũng góp phần nâng cao khả năng mở rộng và bảo trì khi hệ thống phát triển thêm các nghiệp vụ và màn hình mới.

Một số lựa chọn thay thế gồm Vue (cú pháp đơn giản, dễ tiếp cận), Angular (framework đầy đủ, phù hợp hệ thống lớn với cấu trúc chặt chẽ), hoặc Svelte (nhẹ, tối ưu hiệu năng ở runtime). Tuy nhiên, React được lựa chọn vì có hệ sinh thái

phong phú, cộng đồng lớn, tài liệu và nguồn học tập đa dạng, đồng thời phổ biến trong thị trường phát triển phần mềm hiện nay, giúp thuận lợi hơn trong việc phát triển, bảo trì và mở rộng. Ngoài ra, React đồng bộ tốt với backend NodeJS do cùng hệ sinh thái JavaScript/TypeScript, từ đó giảm chi phí học công nghệ mới và giúp quá trình triển khai đồ án diễn ra hiệu quả hơn.

3.1.2 TypeScript

TypeScript là một ngôn ngữ được Microsoft phát triển dựa trên JavaScript, bổ sung hệ kiểu tĩnh (static typing) và các đặc tính hướng đối tượng nhằm tăng khả năng kiểm soát và bảo trì mã nguồn [5]. TypeScript được biên dịch về JavaScript để chạy trên trình duyệt và môi trường NodeJS, vì vậy vẫn tương thích với hệ sinh thái JavaScript hiện có nhưng cung cấp thêm cơ chế kiểm tra lỗi ngay trong quá trình phát triển. Nhờ các khái niệm như *type*, *interface*, *generic* và khả năng suy luận kiểu, TypeScript giúp mô tả rõ cấu trúc dữ liệu và ràng buộc hợp lệ của dữ liệu trong dự án, đặc biệt phù hợp với các hệ thống có nhiều thực thể và nhiều luồng nghiệp vụ.

TypeScript được sử dụng nhằm giải quyết tính ổn định và nhất quán dữ liệu giữa frontend và backend [6], đặc biệt với các cấu trúc dữ liệu có nhiều trường và dễ phát sinh lỗi như: hồ sơ ứng viên, tin tuyển dụng, vector đặc trưng (*skill/tag/location/salary*), và cấu trúc giải thích (*explanation*) trả về kèm *FitScore*. Khi mô hình dữ liệu thay đổi, TypeScript giúp phát hiện sai lệch kiểu dữ liệu ngay tại thời điểm biên dịch, giảm lỗi runtime và hỗ trợ refactor nhanh hơn khi hệ thống mở rộng.

Các lựa chọn thay thế có thể là JavaScript thuần (phát triển nhanh nhưng khó kiểm soát lỗi kiểu khi dự án lớn dần), hoặc kiểm tra dữ liệu ở runtime bằng schema validation phía client (ví dụ: validate JSON trước khi render). TypeScript được lựa chọn vì tạo ra ràng buộc kiểu rõ ràng xuyên suốt mã nguồn, làm giảm rủi ro sai khác giữa API và UI, đồng thời giúp sinh viên quản lý tốt hơn các hợp đồng dữ liệu (*data contract*) trong hệ thống có nhiều luồng nghiệp vụ.

3.2 Tailwind CSS

Tailwind CSS là một framework CSS theo hướng *utility-first*, cung cấp sẵn các lớp CSS nhỏ (*utility classes*) để xây dựng giao diện trực tiếp trong mã HTML/JSX mà không cần viết nhiều stylesheet thủ công [7]. Trong đồ án, Tailwind CSS được sử dụng để giải quyết yêu cầu ở Chương 4 liên quan đến thiết kế giao diện nhất quán, tối ưu tốc độ phát triển UI và đảm bảo khả năng mở rộng khi số lượng màn hình tăng lên. Cụ thể, Tailwind hỗ trợ xây dựng nhanh các thành phần giao diện như layout trang, lưới hiển thị danh sách, khối thẻ tin tuyển dụng, form tìm kiếm/lọc, cũng như các trạng thái hiển thị khác nhau (*hover, focus, responsive*) cho cả ứng

viên và nhà tuyển dụng. Việc áp dụng utility classes giúp giảm trùng lặp CSS, hạn chế xung đột style giữa các module, đồng thời tạo ra một quy ước thiết kế thống nhất trong toàn bộ hệ thống.

Xét về các lựa chọn thay thế, hệ thống có thể sử dụng CSS thuần kết hợp quy ước đặt tên, hoặc các thư viện UI/framework như Bootstrap, Material UI hay Ant Design. CSS thuần và SCSS cho phép tùy biến cao nhưng thường tốn thời gian xây dựng hệ thống style và dễ phát sinh thiếu nhất quán khi dự án mở rộng. Các thư viện UI như Bootstrap hoặc Material UI cung cấp sẵn component nhưng có thể làm giao diện bị phụ thuộc mạnh vào thư viện và khó đồng bộ thiết kế theo yêu cầu riêng của hệ thống. Tailwind CSS được sinh viên lựa chọn vì cân bằng tốt giữa tốc độ phát triển và mức độ tùy biến, dễ duy trì tính đồng nhất giao diện, hỗ trợ responsive đầy đủ và phù hợp với hướng phát triển theo component của React. Ngoài ra, Tailwind có thể cấu hình theme và tái sử dụng quy ước thiết kế xuyên suốt dự án, qua đó cải thiện hiệu quả triển khai và bảo trì giao diện trong phạm vi đồ án.

3.3 NodeJS

NodeJS là môi trường runtime cho JavaScript trên phía máy chủ, được xây dựng trên nền tảng V8 engine [8]. Khác với các mô hình xử lý đồng bộ truyền thống, NodeJS tổ chức xử lý theo cơ chế *event-driven* và *non-blocking I/O*, cho phép ứng dụng xử lý hiệu quả các tác vụ I/O như truy vấn cơ sở dữ liệu, đọc/ghi tệp, gọi API và gửi email. Nhờ đặc điểm này, NodeJS thường được sử dụng để phát triển các hệ thống web cung cấp API có tần suất truy cập cao, yêu cầu phản hồi nhanh và có nhiều thao tác bất đồng bộ.

Trong đồ án, NodeJS được sử dụng để xây dựng tầng dịch vụ và API cho hệ thống, bao gồm: xác thực/ủy quyền, quản lý tài khoản, quản lý hồ sơ ứng viên và nhà tuyển dụng, quản lý tin tuyển dụng, tính toán FitScore và trả về danh sách đề xuất, cũng như kích hoạt các tác vụ gửi thông báo email. Các chức năng này có đặc trưng I/O lớn (truy vấn CSDL, đọc/ghi dữ liệu, gửi email), vì vậy NodeJS với mô hình xử lý bất đồng bộ phù hợp để đáp ứng hiệu năng và khả năng phục vụ đồng thời nhiều request.

Các công nghệ backend thường gặp có thể thay thế NodeJS gồm Spring Boot (mạnh về cấu trúc và hệ sinh thái enterprise), Django/FastAPI (phát triển nhanh, thuận lợi cho bài toán dữ liệu), hoặc .NET (ổn định và hiệu năng cao). NodeJS được lựa chọn do phù hợp với mục tiêu triển khai một hệ thống web ứng dụng theo hướng dịch vụ trong phạm vi đồ án, đồng thời đồng nhất ngôn ngữ với frontend (JavaScript/TypeScript), giúp tăng tốc độ phát triển, giảm thời gian học công nghệ,

và dễ tích hợp các thư viện phục vụ nghiệp vụ như gửi email, xác thực JWT, và kết nối CSDL thông qua ORM.

3.4 MySQL

MySQL được sử dụng để lưu trữ dữ liệu nghiệp vụ có cấu trúc và có quan hệ rõ ràng, bao gồm: tài khoản, phân quyền, hồ sơ ứng viên/nhà tuyển dụng, tin tuyển dụng, kỹ năng/tag, và các quan hệ nhiều-nhiều giữa hồ sơ và danh mục (ví dụ: ứng viên-kỹ năng, tin tuyển dụng-kỹ năng, ứng viên-tag, tin tuyển dụng-tag). Đây là dạng dữ liệu phù hợp với mô hình cơ sở dữ liệu quan hệ và cần đảm bảo tính nhất quán khi cập nhật (đặc biệt ở các thao tác tạo/sửa hồ sơ, tạo/sửa tin, và ghi nhận lịch sử đề xuất hoặc trạng thái tương tác).

Về lựa chọn thay thế, hệ thống có thể dùng PostgreSQL (mạnh về truy vấn nâng cao và mở rộng dữ liệu kiểu JSON), MongoDB (linh hoạt dạng document, phù hợp dữ liệu ít ràng buộc), hoặc SQLite (đơn giản cho demo nhỏ). MySQL được lựa chọn vì phổ biến, dễ triển khai, đáp ứng tốt yêu cầu CRUD và quan hệ dữ liệu của bài toán tuyển dụng trong phạm vi đồ án, đồng thời phù hợp với mục tiêu xây dựng một hệ thống ổn định, dễ vận hành và dễ mở rộng theo hướng tối ưu chỉ mục hoặc tách lớp truy cập dữ liệu khi cần.

3.5 Phương pháp đề xuất dựa theo nội dung (Content-based Filtering – CBF)

Hệ gợi ý (Recommender Systems) là một dạng hệ thống lọc thông tin nhằm dự đoán mức độ quan tâm hoặc mức độ phù hợp giữa người dùng và một đối tượng (item) mà người dùng chưa trực tiếp xem xét trong quá khứ. Trong bối cảnh đồ án tuyển dụng, “người dùng” có thể là ứng viên hoặc nhà tuyển dụng, còn “item” có thể là tin tuyển dụng hoặc ứng viên. Do đó, bài toán đề xuất của hệ thống được hiểu là bài toán ước lượng mức độ phù hợp giữa các cặp đối tượng để từ đó xếp hạng và chọn ra Top-N kết quả gợi ý cho từng ngữ cảnh sử dụng [9].

Đồ án lựa chọn phương pháp gợi ý dựa theo nội dung (Content-based Filtering – CBF). Ý tưởng cốt lõi của CBF là sử dụng chính các thuộc tính mô tả của đối tượng để xây dựng “hồ sơ nội dung” và thực hiện so khớp dựa trên mức độ tương đồng giữa các hồ sơ đó [9]. Thay vì phụ thuộc mạnh vào lịch sử tương tác dày đặc, CBF tập trung vào dữ liệu mô tả như kỹ năng, lĩnh vực/tag, mức lương và địa điểm, từ đó vẫn có thể đưa ra gợi ý ngay cả khi hệ thống mới vận hành hoặc lượng phản hồi hành vi còn hạn chế.

Trong hệ thống của đồ án, CBF được triển khai theo hướng biểu diễn các thực thể dưới dạng vector đặc trưng (vector hồ sơ). Cụ thể, tin tuyển dụng/nhà tuyển dụng được mô hình hóa bởi các thuộc tính phản ánh nhu cầu tuyển dụng như bộ kỹ năng (kèm mức độ quan trọng), tập tag ngành nghề/lĩnh vực (kèm trọng số), mức

lượng đại diện và vị trí; ứng viên được mô hình hóa bởi các thuộc tính phản ánh năng lực và mong muốn như kỹ năng đã có, lĩnh vực quan tâm, mức lương kỳ vọng và vị trí ưu tiên. Từ hai vector hồ sơ, hệ thống tính các điểm thành phần tương ứng (kỹ năng, tag, lương, địa điểm) và tổ hợp chúng thành FitScore theo trọng số cho từng ngữ cảnh. Cách làm này phù hợp với yêu cầu đề xuất hai chiều trong tuyển dụng, đồng thời hỗ trợ khả năng giải thích kết quả gợi ý thông qua các thông tin trùng khớp/thiếu hụt (ví dụ: kỹ năng bắt buộc còn thiếu, tag trùng nhau, mức lương cao hơn/thấp hơn kỳ vọng, địa điểm phù hợp hay trung tính).

Xét về các hướng tiếp cận thay thế, hệ thống có thể sử dụng Collaborative Filtering (CF) dựa trên sự tương đồng hành vi giữa người dùng và/hoặc đối tượng, thường khai thác ma trận user-item (ví dụ: click, lưu tin, ứng tuyển, đánh giá) [9]. Ngoài ra, các mô hình lai (Hybrid) cũng là những hướng mạnh khi có đủ dữ liệu phản hồi để học tối ưu thứ hạng gợi ý theo thời gian. Tuy nhiên, trong phạm vi đề án, CBF được lựa chọn vì ba lý do chính: thứ nhất, giảm ảnh hưởng của bài toán cold-start do không bắt buộc phải có lịch sử tương tác dày; thứ hai, phù hợp với dữ liệu nghiệp vụ sẵn có của tuyển dụng (kỹ năng, tag, lương, địa điểm) và dễ tích hợp vào luồng nghiệp vụ hiện tại; thứ ba, kết quả có thể giải thích rõ ràng, là yếu tố quan trọng trong tuyển dụng vì người dùng thường cần hiểu lý do phù hợp trước khi quyết định ứng tuyển hoặc liên hệ.

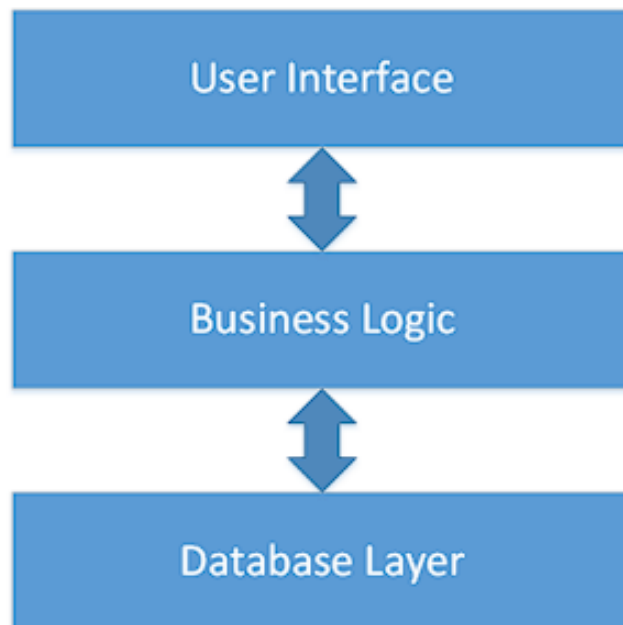
Tóm lại, phương pháp CBF trong đề án được hiện thực hóa bằng cơ chế vector hóa nội dung và công thức FitScore theo ngữ cảnh, qua đó đáp ứng yêu cầu đề xuất cốt lõi của hệ thống, đảm bảo tính khả thi triển khai và tính minh bạch khi trình bày kết quả gợi ý.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Hệ thống được xây dựng theo kiến trúc phân lớp (Layered Architecture), bao gồm ba tầng chính: User Interface, Business Logic và Database Layer, như minh họa trong Hình 4.1. Kiến trúc này cho phép phân tách rõ ràng trách nhiệm giữa các thành phần trong hệ thống, qua đó nâng cao khả năng bảo trì, mở rộng và kiểm soát luồng xử lý nghiệp vụ.



Hình 4.1: Kiến trúc phân lớp của hệ thống [10]

a, Tầng User Interface

Tầng User Interface (UI) chịu trách nhiệm tương tác trực tiếp với người dùng, bao gồm việc hiển thị giao diện và tiếp nhận các yêu cầu từ phía người dùng. Trong hệ thống, tầng này được triển khai chủ yếu trong thư mục client. Các chức năng chính của tầng User Interface bao gồm: hiển thị dữ liệu và trạng thái hệ thống cho người dùng; tiếp nhận các thao tác và yêu cầu (request); gửi các yêu cầu hợp lệ xuống tầng Business Logic để xử lý; và nhận kết quả xử lý để phản hồi lại cho người dùng. Tầng này không chứa logic nghiệp vụ và không thực hiện truy cập trực tiếp đến cơ sở dữ liệu.

b, Tầng Business Logic

Tầng Business Logic là tầng trung tâm của hệ thống, nơi xử lý toàn bộ các quy tắc nghiệp vụ, luồng xử lý và logic cốt lõi của ứng dụng. Tầng này tiếp nhận các yêu cầu hợp lệ từ tầng User Interface, thực hiện kiểm tra điều kiện, xử lý nghiệp vụ và quyết định cách thức truy cập dữ liệu.

Trong hệ thống, tầng Business Logic được triển khai chủ yếu trong thư mục server, bao gồm các Service chịu trách nhiệm xử lý logic nghiệp vụ và thực hiện truy cập cơ sở dữ liệu. Việc truy cập và thao tác dữ liệu được thực hiện thông qua ORM Prisma. Tập schema.prisma đóng vai trò định nghĩa trung tâm cho mô hình dữ liệu và cơ chế persistence, làm cơ sở cho việc sinh mã Prisma Client và đảm bảo tính nhất quán, an toàn về kiểu dữ liệu khi truy cập cơ sở dữ liệu.

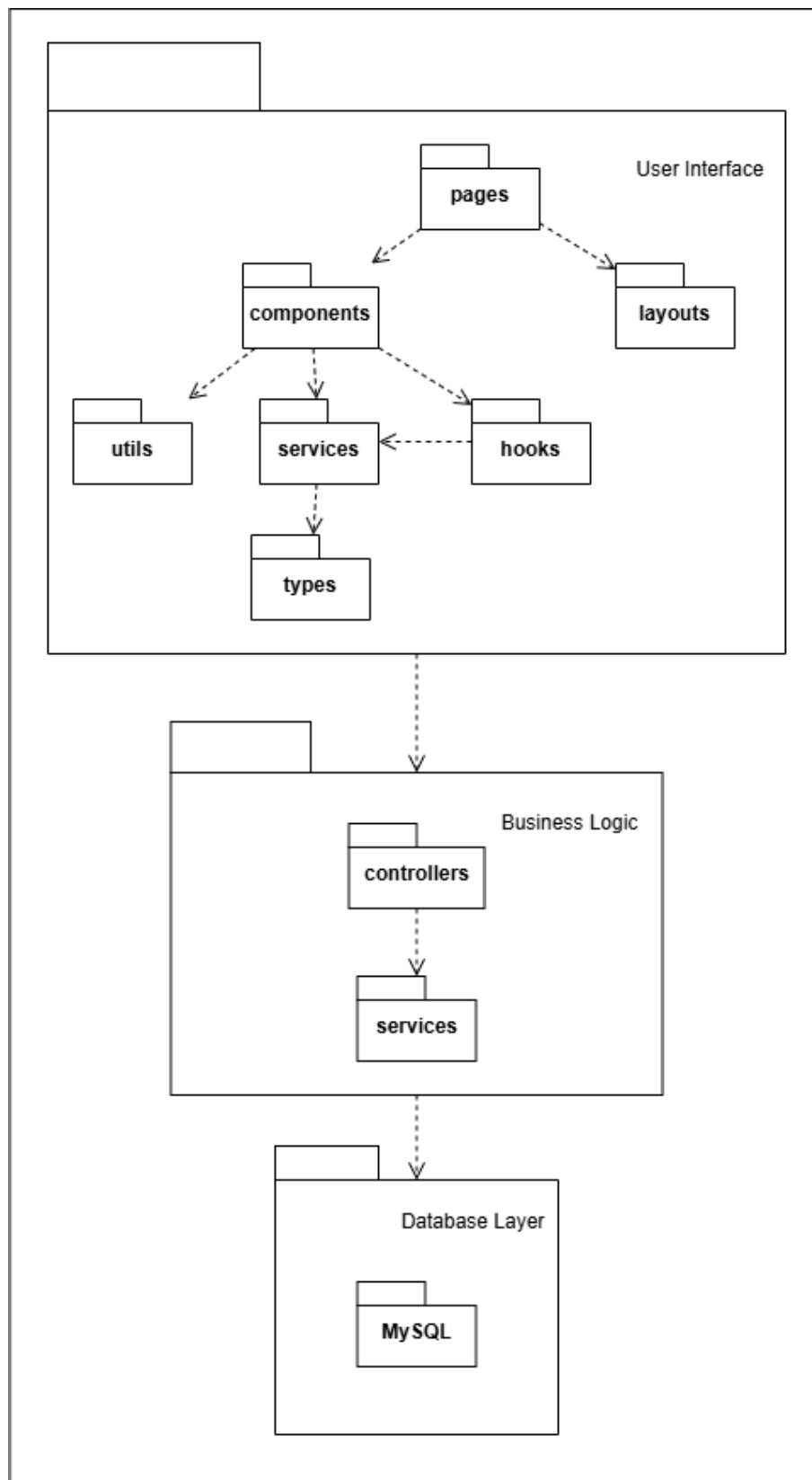
Chức năng persistence trong hệ thống được tích hợp trực tiếp trong tầng Business Logic thông qua các Service và ORM Prisma, thay vì được tách thành một Persistence Layer độc lập. Cách tiếp cận này giúp giảm độ phức tạp của hệ thống trong khi vẫn đảm bảo nguyên tắc phân tách trách nhiệm giữa các tầng.

c, Tầng Database

Tầng Database (Database Layer) đóng vai trò là lớp lưu trữ toàn bộ dữ liệu của hệ thống. Trong đề án này, sinh viên sử dụng hệ quản trị cơ sở dữ liệu quan hệ MySQL. Việc tương tác giữa tầng Business Logic và Database được thực hiện thông qua Prisma ORM. Cách tiếp cận này giúp tối ưu hóa các truy vấn SQL, đồng thời đảm bảo sự đồng nhất giữa mô hình dữ liệu (Schema) trong mã nguồn và cấu trúc bảng thực tế trong cơ sở dữ liệu.

4.1.2 Thiết kế tổng quan

Hệ thống gồm ba tầng chính: User Interface, Business Logic và Database Layer. Mỗi tầng đảm nhiệm một nhóm trách nhiệm riêng biệt và được tổ chức theo mối quan hệ phụ thuộc một chiều từ trên xuống dưới, phù hợp với nguyên tắc của kiến trúc phân lớp. Cấu trúc và mối quan hệ phụ thuộc giữa các gói trong từng tầng của hệ thống được minh họa trong Hình 4.2.



Hình 4.2: Biểu đồ phụ thuộc gói

a, Tầng User Interface

Tầng User Interface chịu trách nhiệm hiển thị giao diện và tương tác trực tiếp với người dùng. Tầng này được tổ chức thành các gói chính gồm: pages, đại diện cho

các trang giao diện của hệ thống; components, bao gồm các thành phần giao diện con có khả năng tái sử dụng trong các trang; layouts, xác định bố cục giao diện cố định cho từng nhóm trang; utils, chứa các hàm và tiện ích dùng chung nhằm giảm trùng lặp mã nguồn; services, đảm nhiệm việc giao tiếp với hệ thống backend thông qua các API; hooks, đóng vai trò là lớp trung gian giữa giao diện và tầng service, giúp đóng gói logic truy xuất và quản lý dữ liệu; và types, dùng để chuẩn hóa và định nghĩa các kiểu dữ liệu trao đổi với backend, đảm bảo tính nhất quán và an toàn kiểu dữ liệu trong quá trình phát triển.

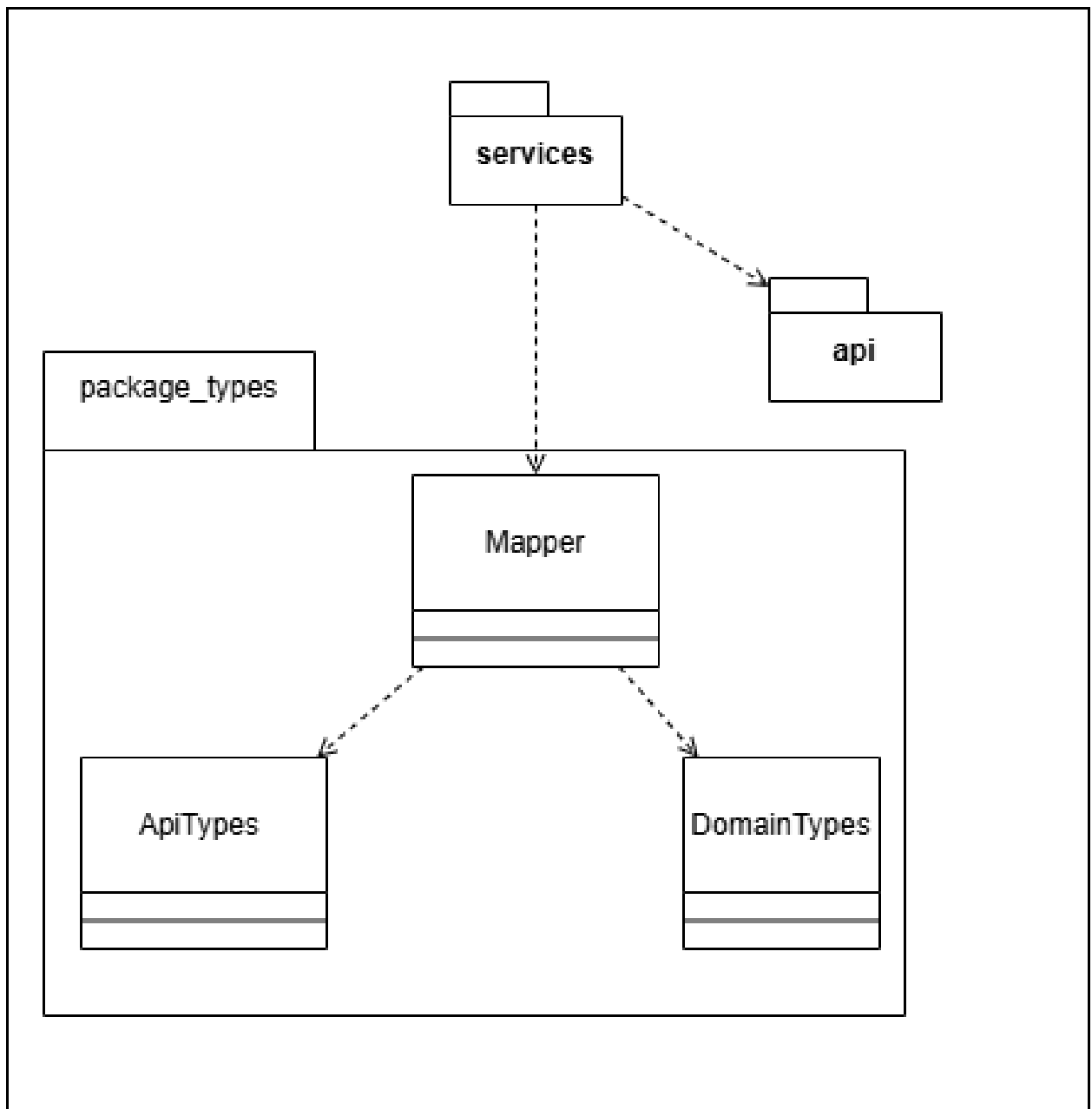
b, Tầng Business Logic

Tầng Business Logic là tầng trung tâm của hệ thống, chịu trách nhiệm xử lý toàn bộ các quy tắc nghiệp vụ và điều phối luồng xử lý chính của ứng dụng. Trong biểu đồ, tầng này được chia thành hai gói con chính là controllers và services. Gói controllers đảm nhiệm việc tiếp nhận các yêu cầu từ tầng User Interface, kiểm tra dữ liệu đầu vào và điều phối luồng xử lý tương ứng. Sau đó, các yêu cầu được chuyển tiếp tới gói services, nơi thực hiện các xử lý nghiệp vụ cốt lõi của hệ thống.

c, Tầng Database

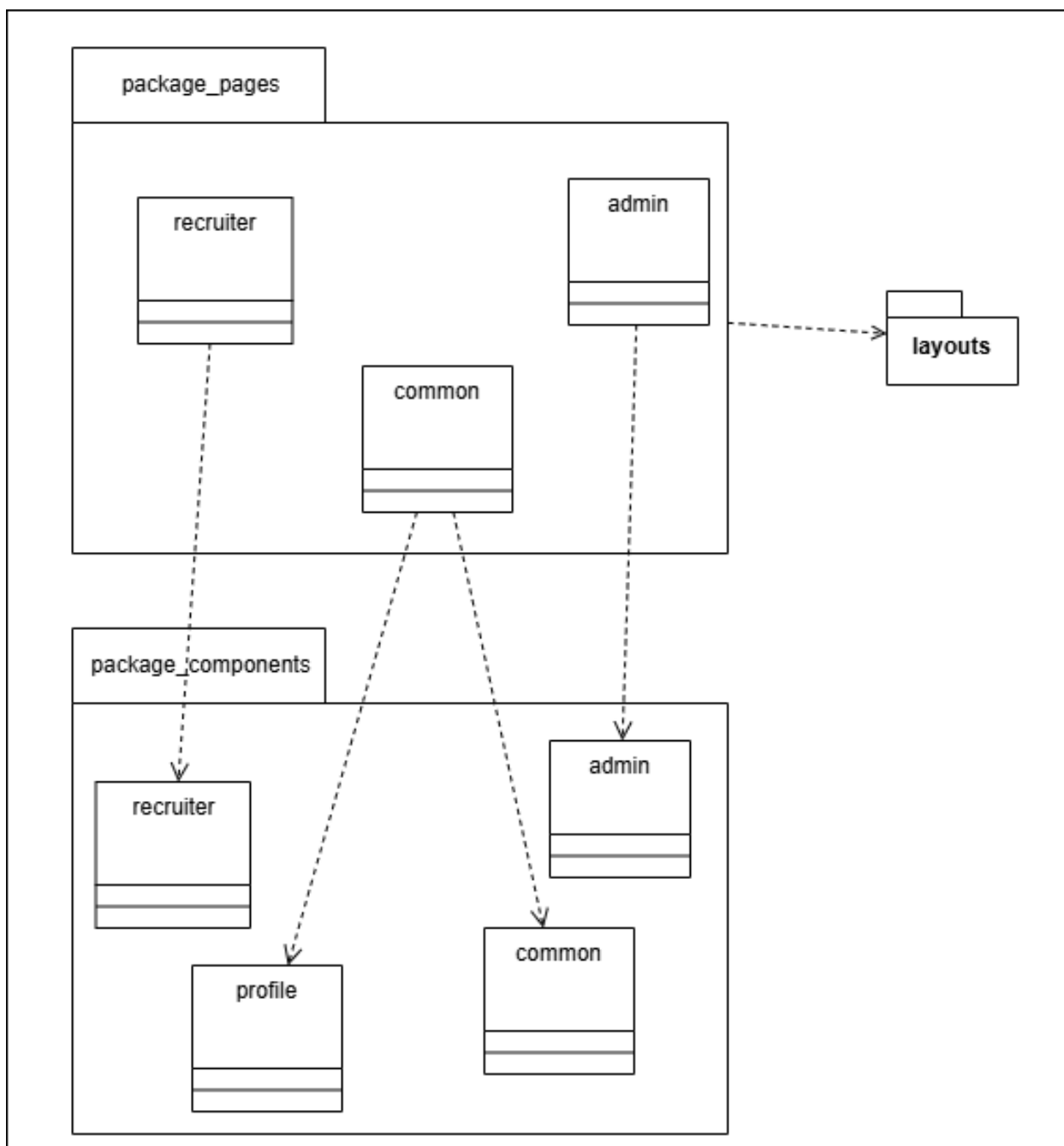
Tầng Database Layer đại diện cho hệ quản trị cơ sở dữ liệu của hệ thống. Trong biểu đồ, tầng này được biểu diễn bằng gói MySQL, phản ánh việc hệ thống sử dụng hệ quản trị cơ sở dữ liệu MySQL để lưu trữ dữ liệu.

4.1.3 Thiết kế chi tiết gói



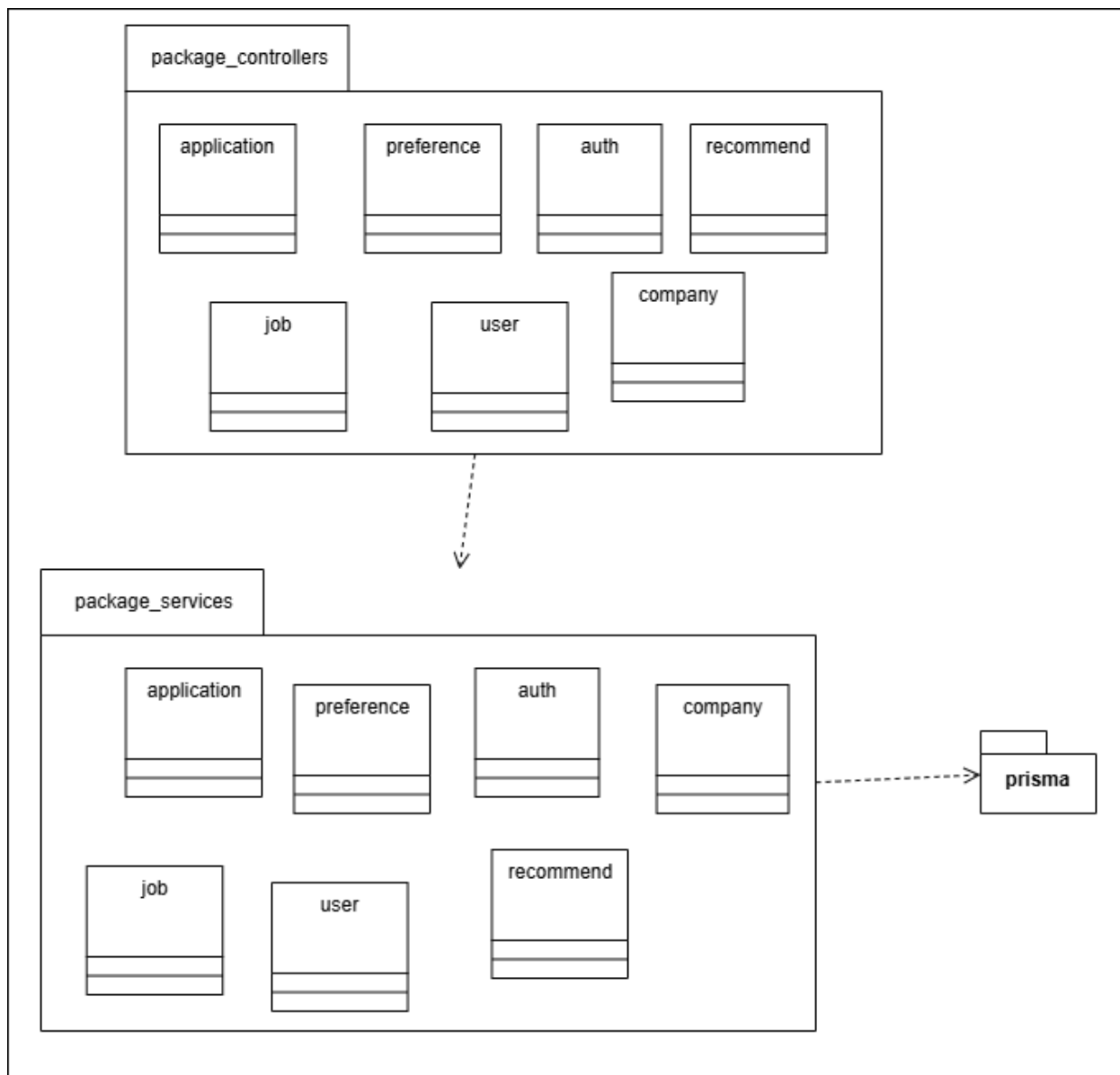
Hình 4.3: Thiết kế gói dịch vụ Frontend

Biểu đồ thiết kế gói ở Hình 4.3 mô tả tầng dịch vụ phía Frontend, bao gồm các gói chính: services, api và types. Trong đó, gói types định nghĩa các kiểu dữ liệu của hệ thống và, thông qua thành phần Mapper, thực hiện việc chuyển đổi dữ liệu nhận từ API sang mô hình dữ liệu nội bộ được sử dụng tại tầng giao diện.



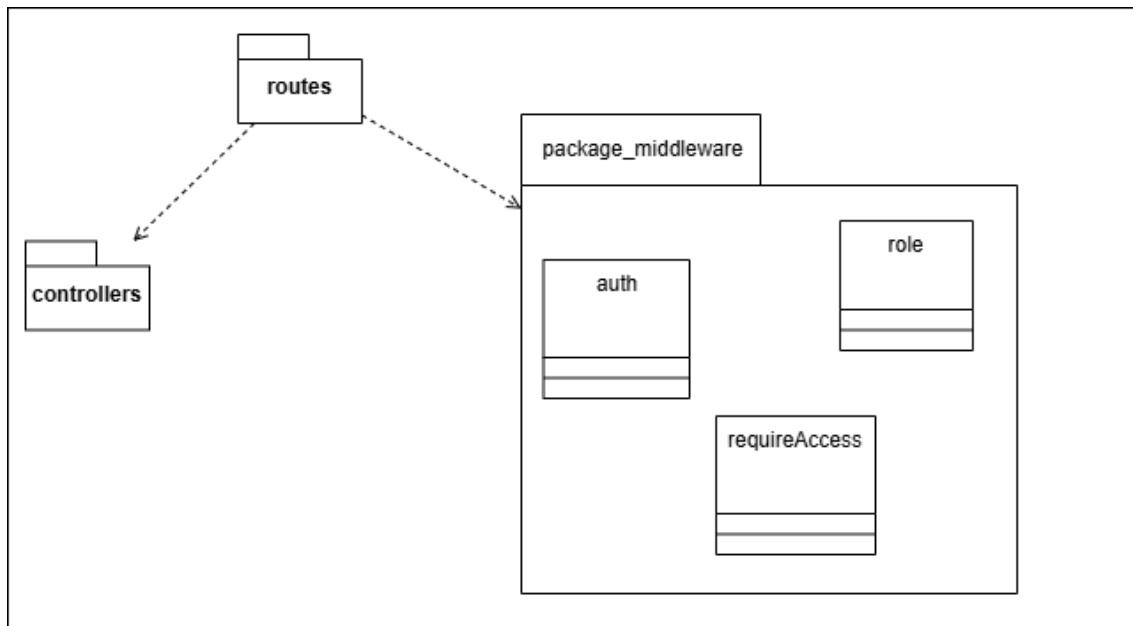
Hình 4.4: Thiết kế gói các trang giao diện của tầng User Interface

Biểu đồ thiết kế gói ở Hình 4.4 mô tả cấu trúc tổ chức các trang giao diện của hệ thống, bao gồm các gói chính: pages, components và layouts. Trong đó, mỗi trang giao diện (page) được xây dựng trên một bố cục chung (layout) và được cấu thành từ nhiều thành phần giao diện tái sử dụng (components).



Hình 4.5: Thiết kế gói Controllers và Services Backend

Biểu đồ thiết kế gói ở Hình 4.5 thể hiện cách tầng ứng dụng phía Backend được tổ chức nhằm xử lý các yêu cầu và thực thi nghiệp vụ của hệ thống. Trong đó, gói services đảm nhiệm việc triển khai logic nghiệp vụ và truy cập dữ liệu thông qua Prisma ORM, trong khi gói controllers đóng vai trò tiếp nhận yêu cầu từ các endpoint, gọi các dịch vụ tương ứng và trả kết quả xử lý về cho phía client.



Hình 4.6: Thiết kế gói định tuyến API Backend

Biểu đồ thiết kế gói ở Hình 4.6 mô tả cách tổ chức tầng định tuyến API của hệ thống. Các endpoint được khai báo trong gói routes và được liên kết với gói controllers để xử lý yêu cầu nghiệp vụ. Trước khi chuyển đến controller, các yêu cầu được kiểm tra thông qua gói middleware, bao gồm xác thực người dùng (auth – kiểm tra token), kiểm tra vai trò (role) và kiểm soát một số chức năng (requireAccess).

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Ứng dụng được thiết kế hướng tới đối tượng người dùng chính là người tìm việc và nhà tuyển dụng, những người chủ yếu truy cập hệ thống thông qua máy tính xách tay (laptop). Trong phạm vi đề án, sinh viên lựa chọn màn hình laptop có kích thước từ 14 inch trở lên làm thiết bị hiển thị mục tiêu để xây dựng và đánh giá giao diện người dùng.

Cụ thể, giao diện được thiết kế dựa trên các thông số sau: kích thước màn hình vật lý 14 inch, tỷ lệ màn hình 16:9 và độ phân giải Full HD (1920 × 1080).

Các thiết kế giao diện được sử dụng trong hệ thống:

(i) Các biểu tượng (icon) được sử dụng từ các thư viện có sẵn, theo phong cách tối giản và có màu sắc phù hợp với ngữ cảnh sử dụng. Các nút bấm và thẻ nội dung được thiết kế với các góc bo tròn nhằm tạo cảm giác thân thiện cho người dùng.

(ii) Bố cục giao diện của các trang bao gồm thanh điều hướng (navbar), vùng nội dung chính (main content) và chân trang (footer). Đối với các trang quản lý, giao

diện được bổ sung thanh điều hướng bên trái (sidebar) để hỗ trợ truy cập nhanh các chức năng.

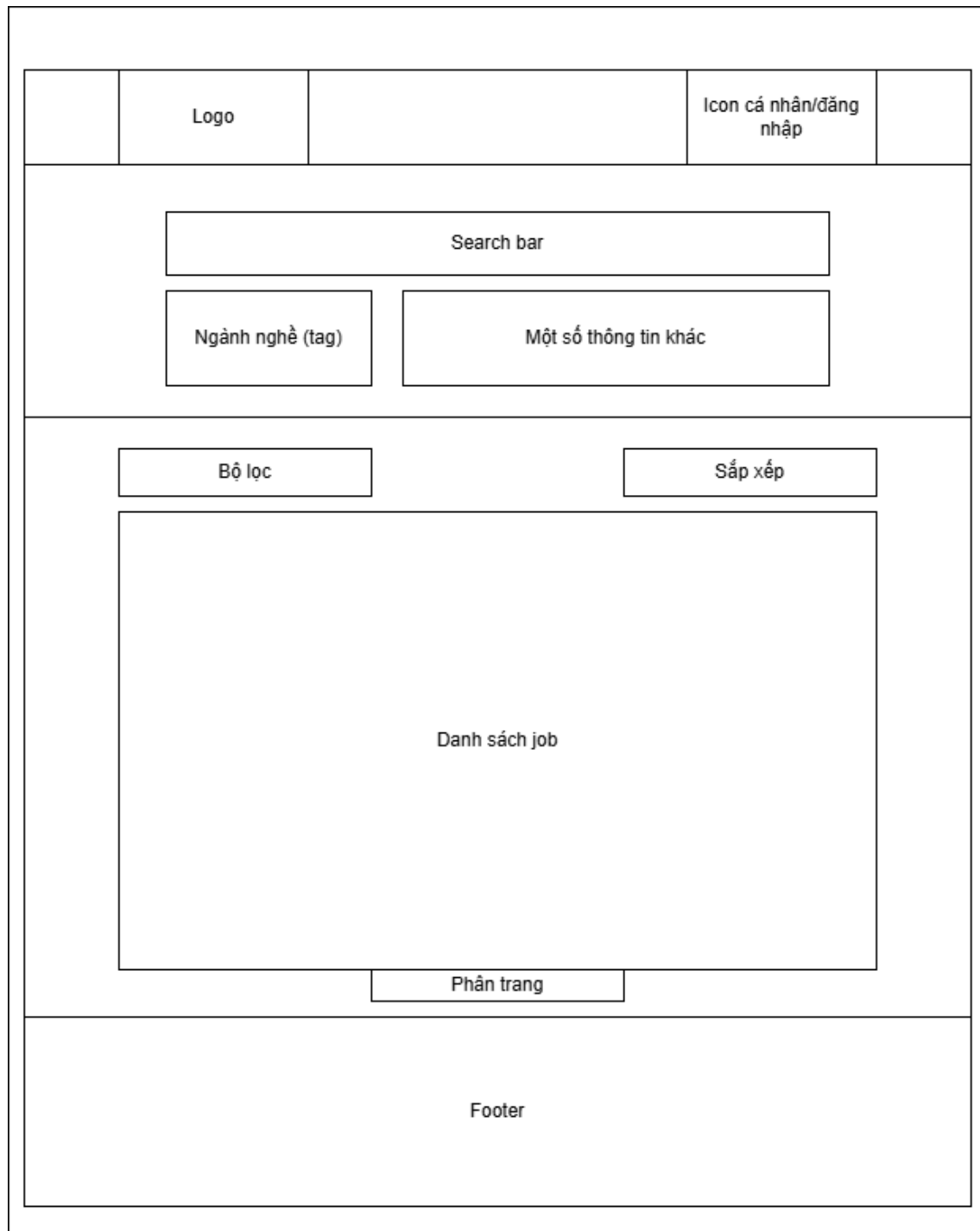
(iii) Logo của hệ thống được bố trí ở phía bên trái thanh điều hướng, trong khi biểu tượng tài khoản cá nhân được đặt ở phía bên phải nhằm thuận tiện cho thao tác của người dùng.

(iv) Vùng nội dung chính được chia thành nhiều khu vực (section) khác nhau, mỗi khu vực hiển thị một nhóm thông tin hoặc chức năng riêng biệt.

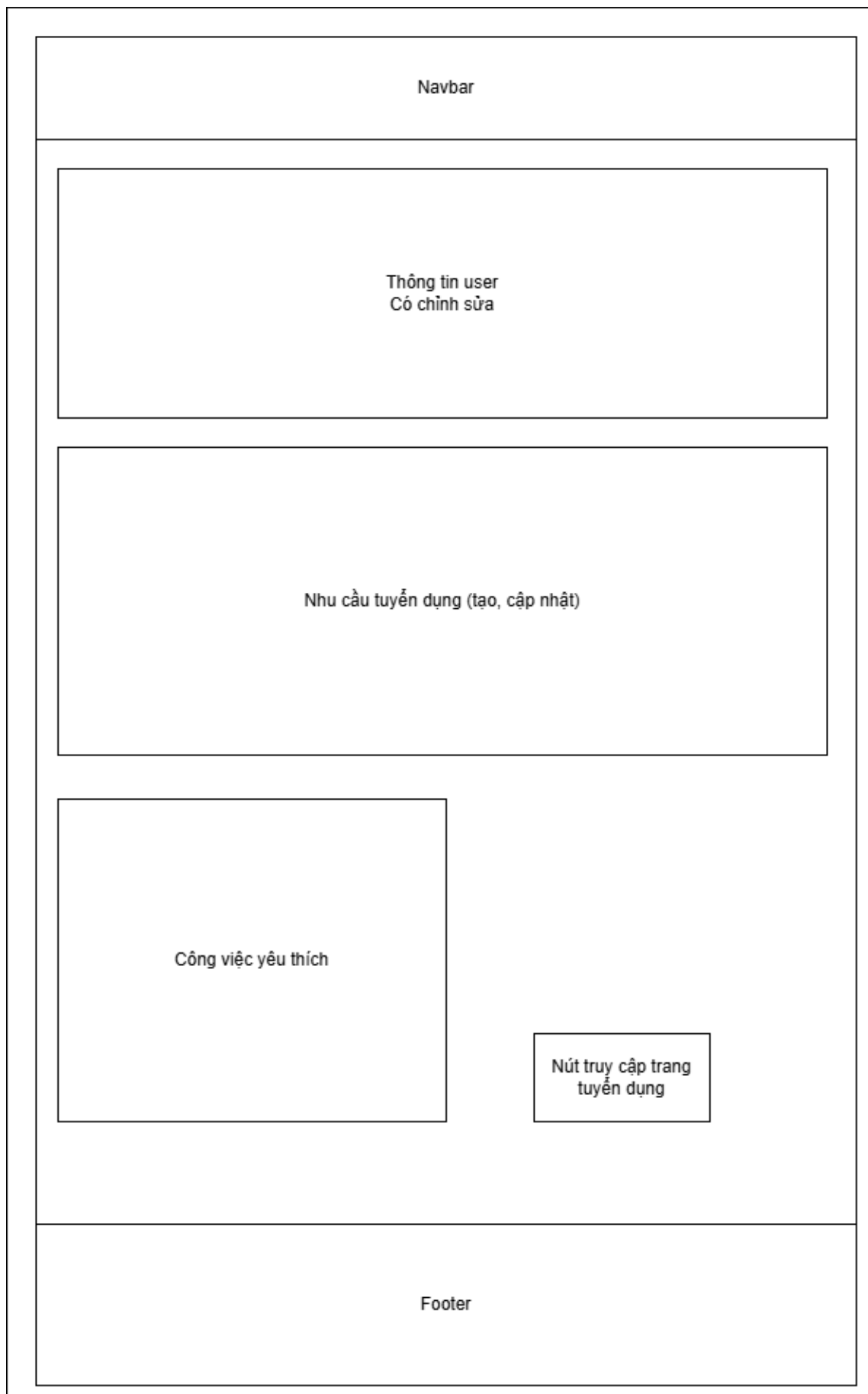
(v) Màu sắc chủ đạo của giao diện là màu xanh dương nhạt, kết hợp với hiệu ứng chuyển màu nhẹ từ trắng sang xanh dương nhằm tạo cảm giác hiện đại và dễ nhìn.



Hình 4.7: Thiết kế giao diện đăng nhập



Hình 4.8: Thiết kế giao diện màn hình chính



Hình 4.9: Thiết kế giao diện trang cá nhân

	Logo		Icon cá nhân/đăng nhập	
Bảng điều khiển	<div>Biểu đồ số công việc đã duyệt/công việc đã từ chối/chưa duyệt</div> <div>Biểu đồ công ty đang chờ/ đã duyệt/từ chối</div> <div> <div>Biểu đồ phân chia user theo role</div> <div>Biểu đồ thống kê user được thêm theo tháng</div> </div> <div>Danh sách user của hệ thống</div> <div>Phân trang</div>			
Danh sách bài đăng				
Danh sách công ty				
Footer				

Hình 4.10: Thiết kế giao diện quản trị viên

4.2.2 Thiết kế lớp

a, Thiết kế chi tiết lớp

User
-id: BigInt -name: String -avatar: String -email: String -role: UserRole -isVerified: boolean -receiveRecommendation: boolean -reset_token_expiry: DateTime -reset_password_hash: String -reset_token: String
+updateProfile() +changePassword() +toggleFavoriteJob() +setReceiveRecommendation() +adminDeleteUser() +adminSetUserActive() +adminCreateUser()

Hình 4.11: Thiết kế chi tiết lớp User

Preference
-user_id: BigInt -title: String -location: String -salary: Int -tags: Tag -skills: SKill
+getCareerPreference() +upsertCareerPreference() +getRecruiterPreference() +upsertRecruiterPreference()

Hình 4.12: Thiết kế chi tiết lớp Preference

Job
-id: BigInt -title: String -created_by_name: String -description: String -location: String -salaryMin: Int -salaryMax: Int -requirements: String -quality_score: Float -status: JobApprovalStatus -application_count: Int -createdAt: DateTime -updatedAt: DateTime
<hr/> +createJob() +updateJob() +getJobById() +approveJob() +rejectJob()

Hình 4.13: Thiết kế chi tiết lớp Job

Recommendation
+generateRecommendationsForUser() +generateCandidateRecommendations() +buildJobReasonFromExplanation() +getRecommendedJobsForUser() +getRecommendedCandidatesForRecruiter()

Hình 4.14: Thiết kế chi tiết lớp Recommendation

Application
-id: BigInt -status: ApplicationStatus -coverLetter: String -cv: String -phone: String -status: String -fitScore: Float -fitReason: String -review_note: String -reviewed_by: BigInt
+createApplication() +getApplicantsByJob() +reviewApplication() +updateApplication() +buildApplicationFitReason()

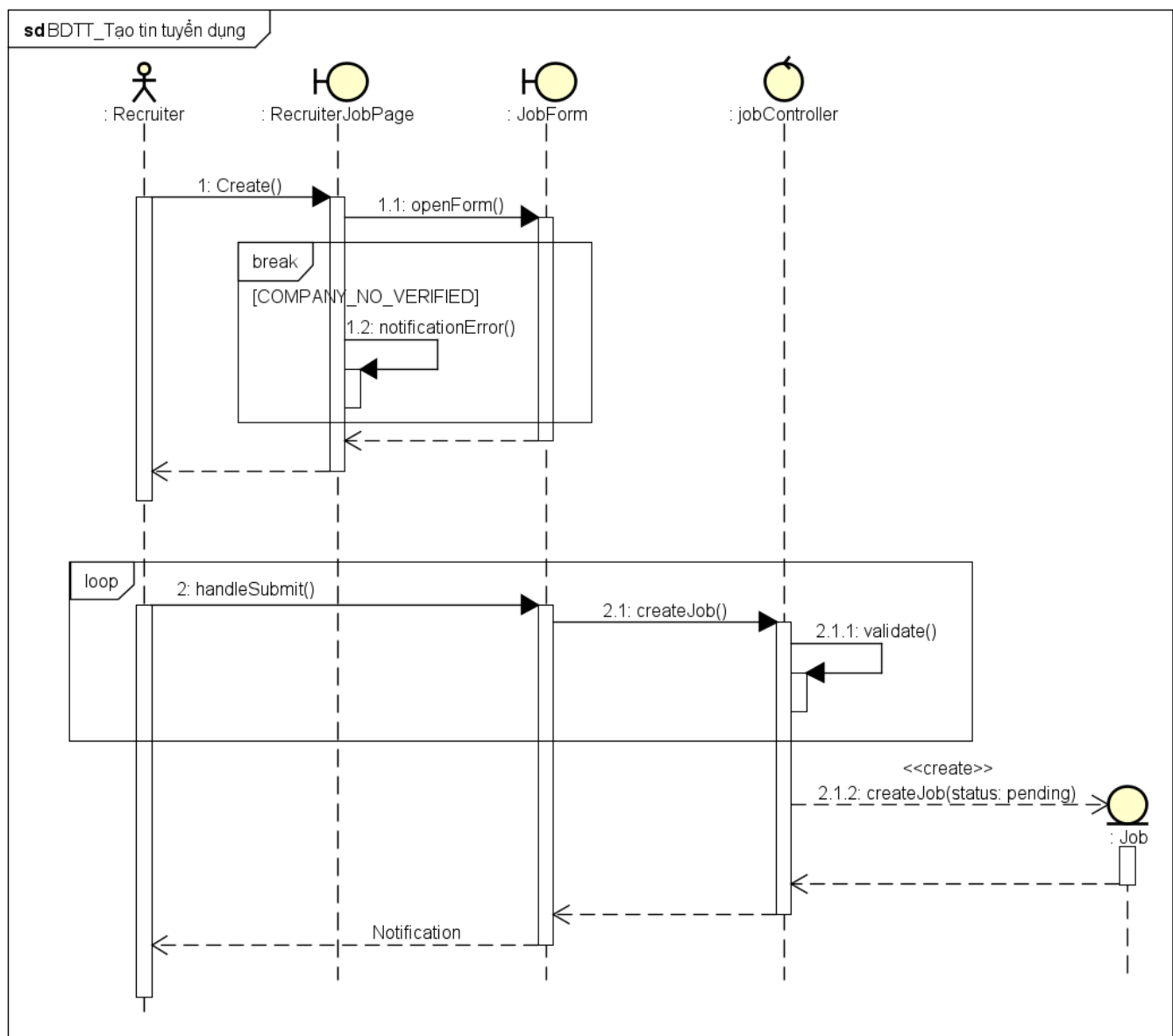
Hình 4.15: Thiết kế chi tiết lớp Application

Các lớp User, Preference, Job, Application và RecommendationService là các lớp chủ đạo của hệ thống. Các lớp này đóng vai trò trung tâm trong việc quản lý dữ liệu nghiệp vụ và triển khai chức năng đề xuất thông minh, bao gồm lưu trữ thông tin người dùng, công việc, hồ sơ ứng tuyển và thực hiện các thuật toán gợi ý.

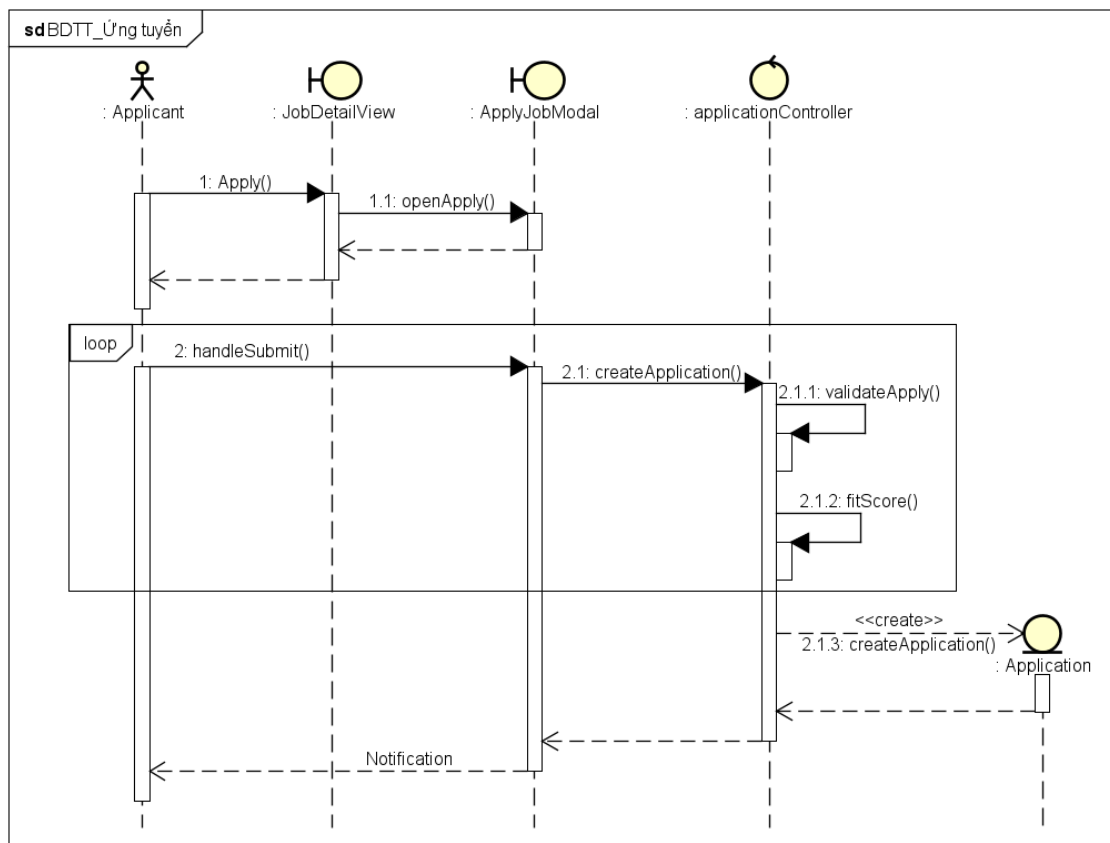
b, Biểu đồ trình tự

Các use case Tạo tin tuyển dụng, Ứng tuyển, Mong muốn công việc, Mong muốn tuyển dụng và Tạo đề xuất là các use case chính của hệ thống.

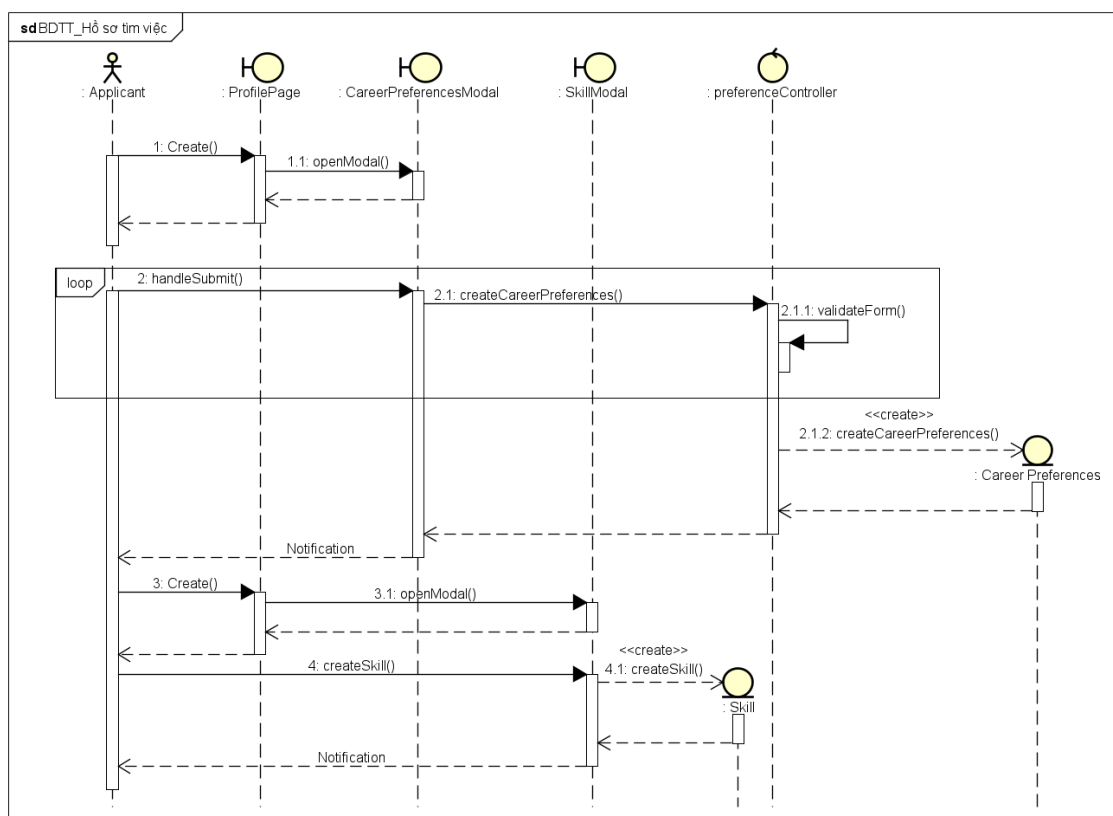
Các use case này phản ánh các nghiệp vụ cốt lõi và được mô tả chi tiết thông qua các biểu đồ trình tự ở các hình sau.



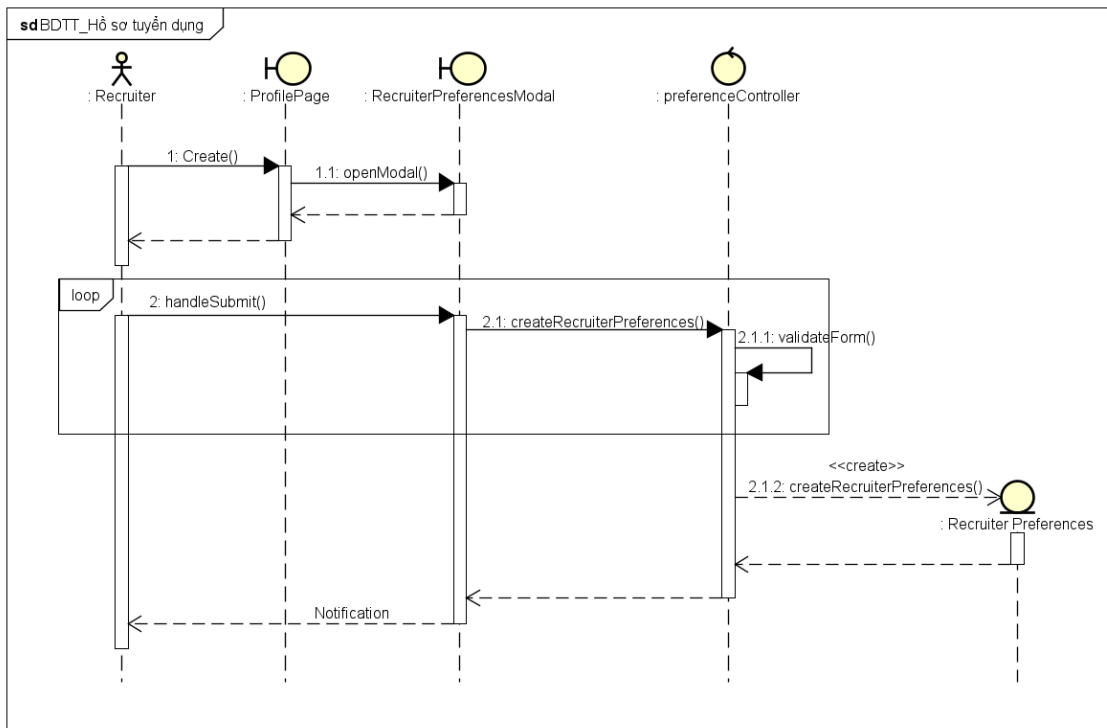
Hình 4.16: Biểu đồ trình tự use case Tạo tin tuyển dụng



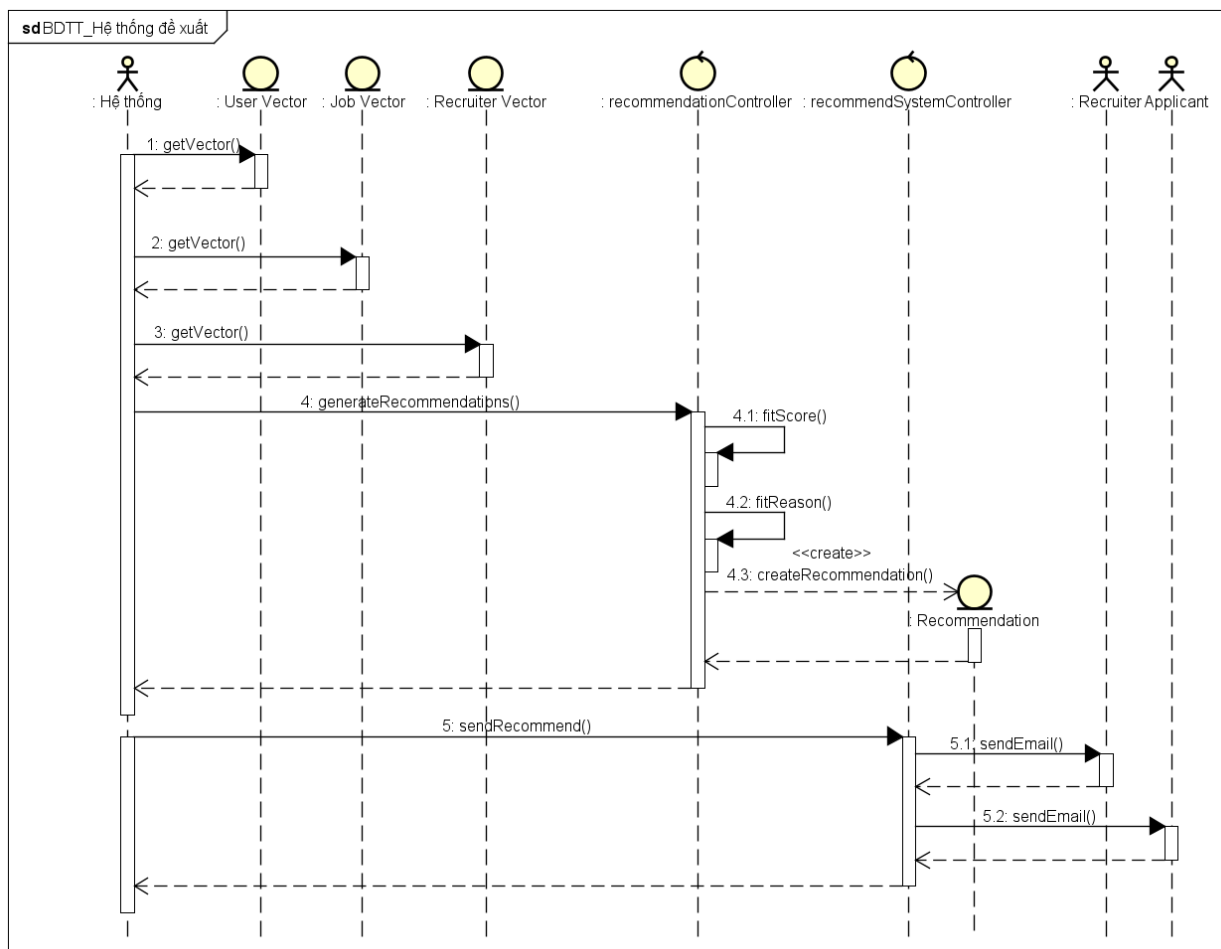
Hình 4.17: Biểu đồ trình tự use case Ứng tuyển



Hình 4.18: Biểu đồ trình tự use case Mong muốn công việc



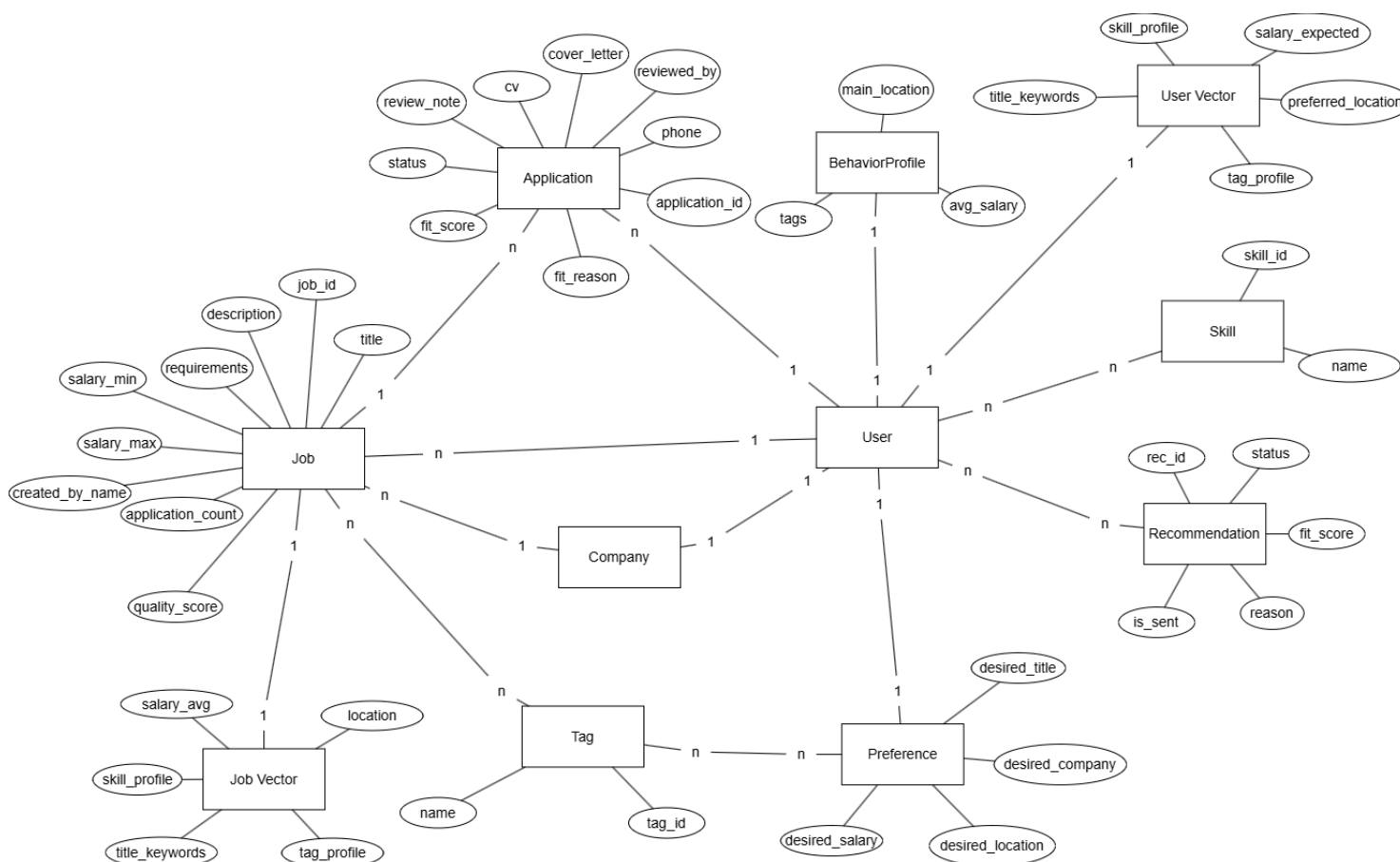
Hình 4.19: Biểu đồ trình tự use case Mong muốn tuyển dụng



Hình 4.20: Biểu đồ trình tự use case Tạo đề xuất

4.2.3 Thiết kế cơ sở dữ liệu

a, Biểu đồ thực thể liên kết



Hình 4.21: Biểu đồ thực thể liên kết

Biểu đồ thực thể–liên kết trong Hình 4.21 là phiên bản giản lược, tập trung vào các thực thể chính của hệ thống, bao gồm: Job (tin tuyển dụng), User (người dùng), Application (đơn ứng tuyển), Tag (ngành nghề), Skill (kỹ năng), Preference (mong muốn), JobVector, UserVector, BehaviorProfile (hành vi người dùng) và Recommendation (đề xuất).

Trong đó, User là thực thể trung tâm với ba vai trò chính: ứng viên, nhà tuyển dụng và quản trị viên. Tùy theo vai trò, người dùng sẽ có các quan hệ dữ liệu khác nhau. Cụ thể, nhà tuyển dụng liên quan đến các thực thể tin tuyển dụng và công ty; ứng viên có các liên kết tới hồ sơ ứng tuyển, hành vi người dùng và các thông tin mong muốn việc làm; trong khi đó, quản trị viên đảm nhiệm các chức năng quản lý và phê duyệt trong hệ thống.

Mỗi Job được tạo bởi một nhà tuyển dụng và thuộc về một công ty duy nhất. Đối với mỗi tin tuyển dụng, hệ thống xây dựng một JobVector tương ứng nhằm phục vụ cho việc tính toán mức độ phù hợp trong quá trình đề xuất.

Một ứng viên có thể tạo nhiều Application để ứng tuyển vào các tin tuyển dụng khác nhau. Đồng thời, mỗi tin tuyển dụng cũng có thể nhận được nhiều hồ sơ ứng tuyển từ các ứng viên khác nhau.

BehaviorProfile là tập dữ liệu được tổng hợp và tính toán dựa trên hành vi sử dụng của người dùng trong hệ thống. Dữ liệu này chỉ được ghi nhận đối với người dùng có vai trò là ứng viên.

Preference là các thông tin mong muốn của người dùng, bao gồm cả ứng viên và nhà tuyển dụng. Trong thực tế, hệ thống phân tách dữ liệu này thành hai bảng riêng biệt tương ứng với từng vai trò của người dùng.

Skill và Tag là các thông tin cơ bản, được sử dụng để mô tả yêu cầu của tin tuyển dụng, hồ sơ người dùng.

Đối với mỗi vai trò người dùng, hệ thống lưu trữ các dữ liệu Recommendation tương ứng, làm cơ sở cho việc triển khai hệ thống đề xuất phù hợp.

b, Danh sách các bảng dữ liệu

Bảng 4.1: Bảng Users

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	Khóa chính (PK), tự tăng, định danh người dùng
2	name	String	Tên hiển thị của người dùng
3	avatar	String	Đường dẫn ảnh đại diện
4	email	String	Email đăng nhập, duy nhất
5	password	String	Mật khẩu đã được mã hóa
6	role	UserRole	Vai trò người dùng: admin, recruiter, applicant
7	isVerified	Boolean	Trạng thái xác thực tài khoản
8	receive_recommendation	Boolean	Cho phép nhận đề xuất từ hệ thống
9	reset_token	String	Token đặt lại mật khẩu
10	reset_token_expiry	DateTime	Thời hạn hiệu lực của token
11	reset_password_hash	String	Hash xác nhận đặt lại mật khẩu
12	created_at	DateTime	Thời điểm tạo tài khoản
13	updated_at	DateTime	Thời điểm cập nhật gần nhất

Bảng 4.2: Bảng Companies

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	Khóa chính (PK), định danh công ty
2	legal_name	String	Tên pháp lý của công ty
3	registration_number	String	Mã đăng ký doanh nghiệp
4	tax_id	String	Mã số thuế
5	country_code	String	Mã quốc gia
6	registered_address	String	Địa chỉ đăng ký
7	incorporation_date	DateTime	Ngày thành lập
8	owner_id	BigInt	Khóa ngoại (FK), chủ công ty
9	logo	String	Đường dẫn logo công ty
10	created_at	DateTime	Thời điểm tạo
11	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.3: Bảng CompanyVerifications

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	PK, định danh bản ghi xác thực
2	reviewed_by	BigInt	FK, mã quản trị viên xử lý
3	company_id	BigInt	FK, mã công ty
4	status	CompanyVerificationStatus	Trạng thái xác minh (submitted, verified, rejected)
5	rejection_reason	Text	Lý do từ chối
6	submitted_at	DateTime	Thời điểm gửi yêu cầu xác minh
7	verified_at	DateTime	Thời điểm xác minh

Bảng 4.4: Bảng Jobs

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	Khóa chính (PK), định danh tin tuyển dụng
2	title	String	Tiêu đề tin tuyển dụng
3	company_id	BigInt	Khóa ngoại (FK) công ty sở hữu
4	created_by	BigInt	Khóa ngoại (FK) Người tạo
5	created_by_name	String	Tên nhà tuyển dụng tạo
6	location	String	Địa điểm làm việc

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
7	description	Text	Mô tả chi tiết công việc
8	salary_min	Int	Mức lương tối thiểu
9	salary_max	Int	Mức lương tối đa
10	requirements	Text	Yêu cầu công việc
11	quality_score	Float	Điểm chất lượng
12	application_count	Int	Số đơn đã ứng tuyển
13	created_at	DateTime	Thời điểm tạo
14	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.5: Bảng JobApprovals

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	PK, định danh bản ghi phê duyệt
2	job_id	BigInt	FK, mã tin tuyển dụng
3	status	JobApprovalStatus	Trạng thái phê duyệt (pending, approved, rejected)
4	reason	Text	Lý do từ chối
5	auditor_id	BigInt	FK, người phê duyệt
6	audited_at	DateTime	Thời điểm phê duyệt
7	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.6: Bảng JobRequiredSkills

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	job_id	BigInt	PK, FK, mã tin tuyển dụng
2	skill_id	Int	PK, FK, mã kỹ năng
3	level_required	Int	Mức độ kỹ năng yêu cầu (1-5)
4	years_required	Int	Số năm kinh nghiệm yêu cầu
5	must_have	Boolean	Kỹ năng bắt buộc
6	fit_weight	Float	Trọng số kỹ năng trong tính điểm phù hợp

Bảng 4.7: Bảng Tags

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Int	PK, tự tăng
2	name	String	Tên ngành nghề / lĩnh vực

Bảng 4.8: Bảng JobTags

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	job_id	BigInt	PK, FK, mã tin tuyển dụng
2	tag_id	Int	PK, FK, mã ngành nghề

Bảng 4.9: Bảng UserFavoriteJobs

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK. mã người dùng
2	job_id	BigInt	PK, FK, mã tin tuyển dụng

Bảng 4.10: Bảng Applications

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	Khóa chính (PK), định danh đơn ứng tuyển
2	job_id	BigInt	Khóa ngoại (FK) mã tin tuyển dụng
3	applicant_id	BigInt	Khóa ngoại (FK) mã ứng viên
4	cover_letter	Text	Nội dung thư ứng tuyển
5	cv	String	Đường dẫn CV
6	phone	String	Số điện thoại liên hệ
7	status	ApplicationStatus	Trạng thái hồ sơ
8	fit_score	Float	Điểm phù hợp
9	fit_reason	Text	Lý do phù hợp
10	review_note	Text	Ghi chú đánh giá
11	reviewed_by	BigInt	Người đánh giá
12	reviewed_at	DateTime	Thời điểm đánh giá
13	created_at	DateTime	Thời điểm nộp hồ sơ
14	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.11: Bảng UserInterestHistory

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	PK, định danh bản ghi hành vi

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
2	user_id	BigInt	FK, mã người dùng
3	job_id	BigInt	FK, mã tin tuyển dụng
4	job_title	String	Tiêu đề công việc
5	location	String	Địa điểm làm việc
6	avg_salary	Int	Mức lương trung bình
7	tags	Json	Danh sách ngành nghề
8	source	InterestSource	Nguồn phát sinh hành vi
9	event_type	String	Loại sự kiện (open_detail, apply, ...)
10	recorded_at	DateTime	Thời điểm ghi nhận hành vi
11	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.12: Bảng CareerPreferences

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	desired_title	String	Vị trí mong muốn
3	desired_company	String	Công ty mong muốn
4	desired_location	String	Địa điểm mong muốn
5	desired_salary	Int	Mức lương mong muốn
6	created_at	DateTime	Thời điểm tạo
7	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.13: Bảng CareerPreferenceTags

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	tag_id	Int	PK, FK, mã ngành nghề

Bảng 4.14: Bảng RecruiterPreferences

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	desired_location	String	Địa điểm mong muốn tuyển dụng
3	desired_salary_avg	Int	Mức lương trung bình mong muốn

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
4	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.15: Bảng RecruiterRequiredSkills

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	skill_id	Int	PK, FK, mã kỹ năng
3	years_required	Int	Số năm kinh nghiệm yêu cầu
4	must_have	Boolean	Kỹ năng bắt buộc

Bảng 4.16: Bảng RecruiterPreferenceTags

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	tag_id	Int	PK, FK, mã ngành nghề

Bảng 4.17: Bảng UserBehaviorProfile

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	avg_salary	Int	Mức lương trung bình
3	main_location	String	Địa điểm chính
4	tags	Json	Tập tag tổng hợp
5	keywords	Json	Từ khóa nổi bật
6	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.18: Bảng JobRecommendations

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	PK, định danh đề xuất
2	user_id	BigInt	FK, ứng viên
3	job_id	BigInt	FK, mã tin tuyển dụng
4	fit_score	Float	Điểm phù hợp
5	reason	Text	Giải thích đề xuất
6	is_sent	Boolean	Trạng thái đã gửi đề xuất
7	sent_at	DateTime	Thời điểm gửi
8	status	String	Trạng thái xử lý

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
9	recommended_at	DateTime	Thời điểm tạo đề xuất
10	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.19: Bảng CandidateRecommendations

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	BigInt	PK, định danh đề xuất
2	recruiter_id	BigInt	FK), nhà tuyển dụng
3	applicant_id	BigInt	FK, ứng viên
4	fit_score	Float	Điểm phù hợp
5	reason	Text	Giải thích đề xuất
6	status	String	Trạng thái đề xuất
7	is_sent	Boolean	Trạng thái đã gửi
8	sent_at	DateTime	Thời điểm gửi
9	recommended_at	DateTime	Thời điểm tạo
10	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.20: Bảng Skills

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	Int	PK, tự tăng
2	name	String	Tên kỹ năng

Bảng 4.21: Bảng UserSkills

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	skill_id	Int	PK, FK, mã kỹ năng
3	level	Int	Mức độ kỹ năng (1-5)
4	years	Int	Số năm kinh nghiệm
5	note	Text	Ghi chú thêm

Bảng 4.22: Bảng UserVector

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	skill_profile	Json	Tổng hợp kỹ năng

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
3	tag_profile	Json	Tổng hợp ngành nghề
4	title_keywords	Json	Từ khóa chính
5	preferred_location	String	Khu vực ưu tiên
6	salary_expected	Int	Mức lương kỳ vọng
7	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.23: Bảng RecruiterVector

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, nhà tuyển dụng
2	skill_profile	Json	Tổng hợp kỹ năng
3	tag_profile	Json	Tổng hợp ngành nghề
4	preferred_location	String	Khu vực tuyển dụng ưu tiên
5	salary_avg	Int	Mức lương trung bình
6	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.24: Bảng JobVector

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	job_id	BigInt	PK, FK, mã tin tuyển dụng
2	skill_profile	Json	Tổng hợp kỹ năng
3	tag_profile	Json	Tổng hợp ngành nghề
4	title_keywords	Json	Từ khóa chính
5	location	String	Địa điểm làm việc
6	salary_avg	Int	Mức lương trung bình
7	updated_at	DateTime	Thời điểm cập nhật

Bảng 4.25: Bảng UserJobMatrix

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	user_id	BigInt	PK, FK, mã người dùng
2	job_id	BigInt	PK, FK, mã tin tuyển dụng
3	score	Float	Điểm phù hợp
4	updated_at	DateTime	Thời điểm cập nhật

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 4.26 liệt kê các công cụ chính; danh sách đã được giản lược, chỉ giữ các thư viện tối thiểu cần thiết để hệ thống có thể vận hành.

Mục đích	Công cụ	Phiên bản	Địa chỉ URL
IDE lập trình	Visual Studio Code	1.108.0	https://code.visualstudio.com/
Ngôn ngữ lập trình Backend	Javascript		https://www.javascript.com/
Ngôn ngữ lập trình Fontend	Typescript	5.8.3	https://www.typescriptlang.org/
Web framework (REST API)	Express	5.1.0	https://expressjs.com/
Nền tảng lập trình	NodeJS	22.18.0	https://nodejs.org/
Framework Fontend	React	19.1.1	https://react.dev/
ORM	Prisma	6.15.0	https://www.prisma.io/
Cơ sở dữ liệu	MySQL	9.4.0	https://www.mysql.com/
Quản lý server-state	@tanstack/react-query	5.90.11	https://tanstack.com/query/
Xác thực JWT	jsonwebtoken	9.0.2	https://www.jwt.io/
Gọi API HTTP	Axios	1.11.0	https://axios-http.com/
CSS giao diện	Tailwind CSS	4.1.11	https://tailwindcss.com/
Quản lý state	Zustand	5.0.8	https://zustand-demo.pmnd.rs/
Công cụ build mã nguồn	Vite	7.1.0	https://vitejs.dev/
Thư viện gửi email	Nodemailer	7.0.5	https://nodemailer.com/
Tài liệu API (Swagger)	swagger-jsdoc	6.2.8	https://swagger.io/
Tìm và sửa lỗi code	ESLint	9.34.0	https://eslint.org/
Định dạng mã nguồn	Prettier	3.6.2	https://prettier.io/
Kiểm thử API	Postman	11.76.9	https://www.postman.com/
Quản lý phiên bản	Git	2.45.1	https://git-scm.com/
Kiểm tra và bảo mật	SonarQube	2.45.1	https://www.sonarsource.com/

Bảng 4.26: Danh sách thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

Hệ thống được đóng gói dưới dạng một ứng dụng website hoàn chỉnh, bao gồm backend kết nối và thao tác với cơ sở dữ liệu, cùng frontend triển khai giao diện phục vụ người dùng. Mã nguồn được tổ chức tách biệt theo từng thành phần, giúp việc triển khai trên môi trường/thiết bị khác thực hiện đơn giản thông qua các bước cơ bản như: cài đặt các thư viện phụ thuộc và cấu hình thông tin kết nối cơ sở dữ liệu; sau đó hệ thống có thể vận hành ổn định.

Sản phẩm đóng gói gồm hai thành phần chính tương ứng với hai thư mục server và client. Thư mục server chứa các thành phần cấu thành backend, thực hiện các chức năng như: kết nối cơ sở dữ liệu, xử lý nghiệp vụ, và trả kết quả về cho phía client thông qua API. Thư mục client chứa mã nguồn frontend, bao gồm các giao diện tương tác với người dùng, tiếp nhận sự kiện thao tác, gửi yêu cầu đến backend và hiển thị kết quả phản hồi trên giao diện.

Các thông tin chi tiết của ứng dụng được thống kê trong Bảng 4.27. Trong đó, số dòng code được tính không bao gồm dòng chú thích (comment); dung lượng mã nguồn được tính không bao gồm thư mục chứa ảnh/tài nguyên tĩnh và không bao gồm các gói thư viện cài đặt.

STT	Mô tả	Số lượng	Đơn vị
1	Số dòng code trong thư mục server	13,713	dòng
2	Số dòng code trong thư mục client	7,373	dòng
3	Dung lượng toàn bộ mã nguồn	1424	Kb
4	Số module	14	module
5	Số file	253	file

Bảng 4.27: Thống kê thông tin ứng dụng

4.3.3 Minh họa các chức năng chính

Các chức năng nổi bật của hệ thống xoay quanh nghiệp vụ tin tuyển dụng, mô hình hoá nhu cầu người dùng, và chức năng đề xuất. Dựa trên bố cục giao diện đã trình bày tại Mục 2, Chương 4, sinh viên đã xây dựng các màn hình giao diện chính, đáp ứng đầy đủ các chức năng cốt lõi của hệ thống.

Ứng tuyển công việc

Solutions Architect



Thư ứng tuyển *

Ứng tuyển vị trí Front-end Developer, tôi có kinh nghiệm phát triển giao diện web với React/Next.js và JavaScript/TypeScript, tập trung vào việc xây dựng UI đúng thiết kế, component hoá, tối ưu hiệu năng và đảm bảo trải nghiệm người dùng mượt trên đa thiết bị. Tôi đã từng triển khai các luồng nghiệp vụ phổ biến như authentication, form nhiều bước, trang danh sách/chi tiết, filter/search, phân quyền hiển thị theo role; v.v..

Số điện thoại

0966532140

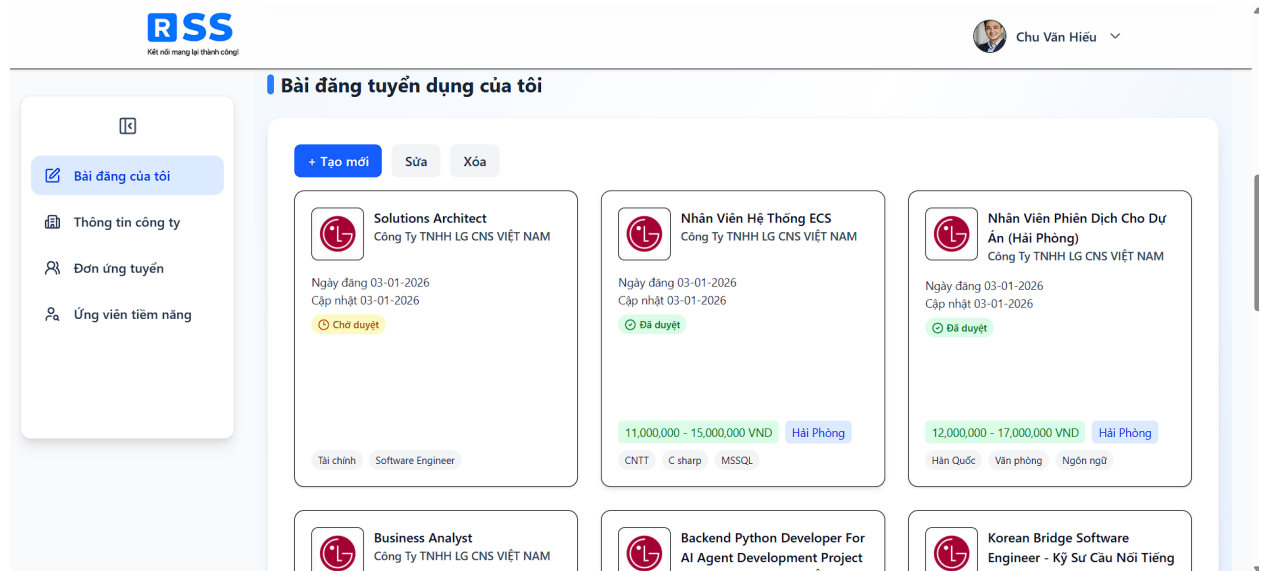
CV *

 CV ứng tuyển FE 2024.pdf

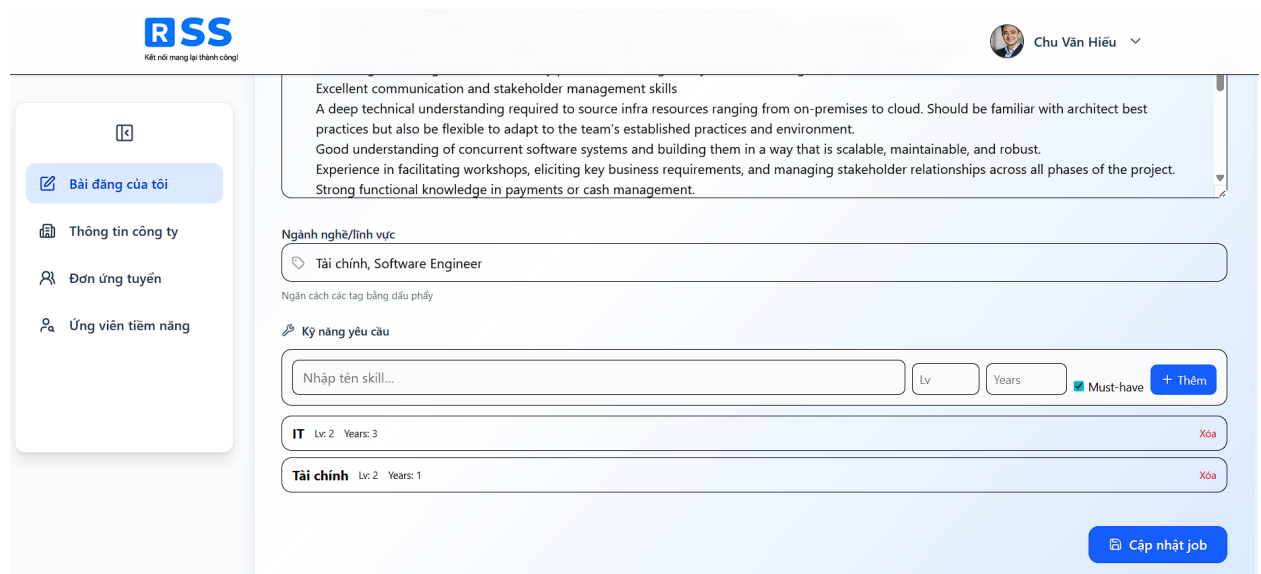
Ứng tuyển ngay

Hình 4.22: Giao diện "Nộp đơn ứng tuyển"

Giao diện trong Hình 4.22 là biểu mẫu mà ứng viên cần điền khi ứng tuyển vào một vị trí. Các trường dữ liệu như CV, số điện thoại và thư ứng tuyển là những thông tin bắt buộc, phục vụ nhà tuyển dụng trong quá trình tiếp nhận và đánh giá hồ sơ ứng tuyển.



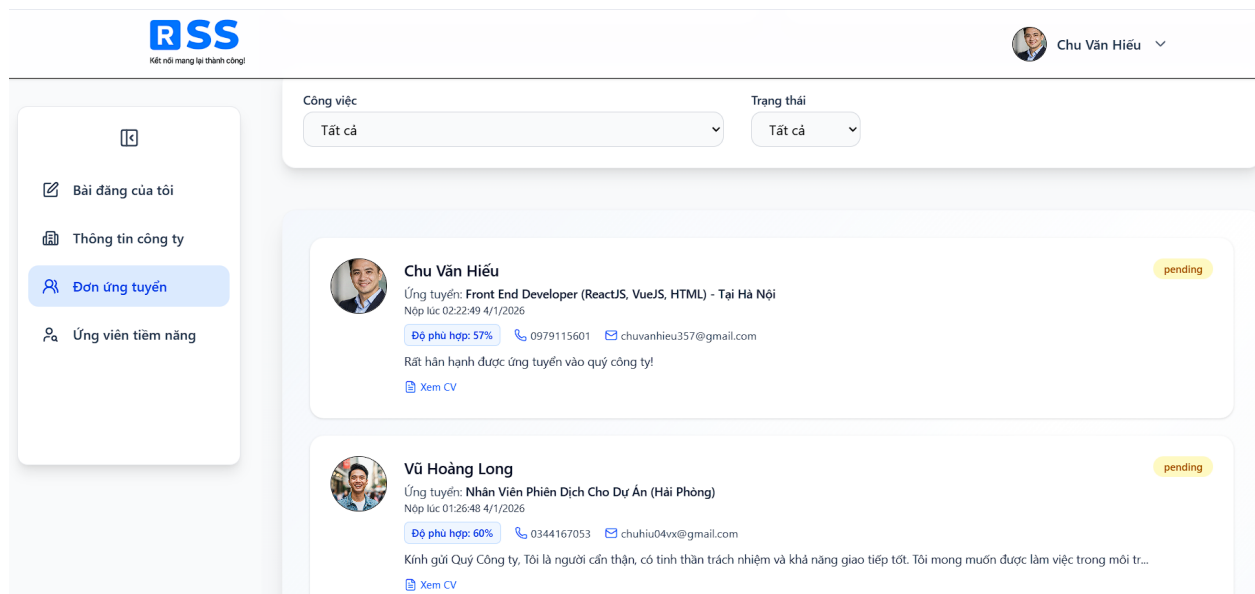
Hình 4.23: Giao diện "Quản lý tin tuyển dụng"



Hình 4.24: Giao diện "Cập nhật tin tuyển dụng"

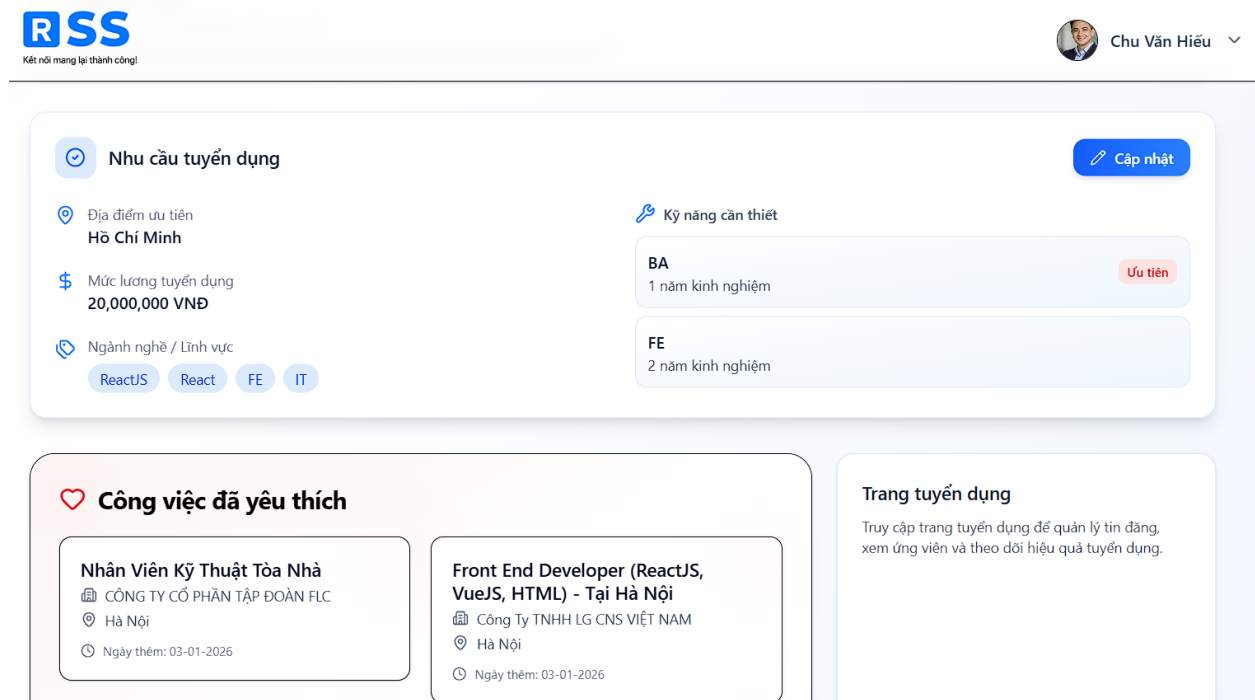
Hình 4.23 minh họa trang chính sau khi nhà tuyển dụng đăng nhập vào hệ thống. Tại đây, nhà tuyển dụng có thể tạo mới, chỉnh sửa và xóa các tin tuyển dụng. Hình 4.24 là biểu mẫu dùng để nhập thông tin khi tạo hoặc cập nhật tin tuyển dụng.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



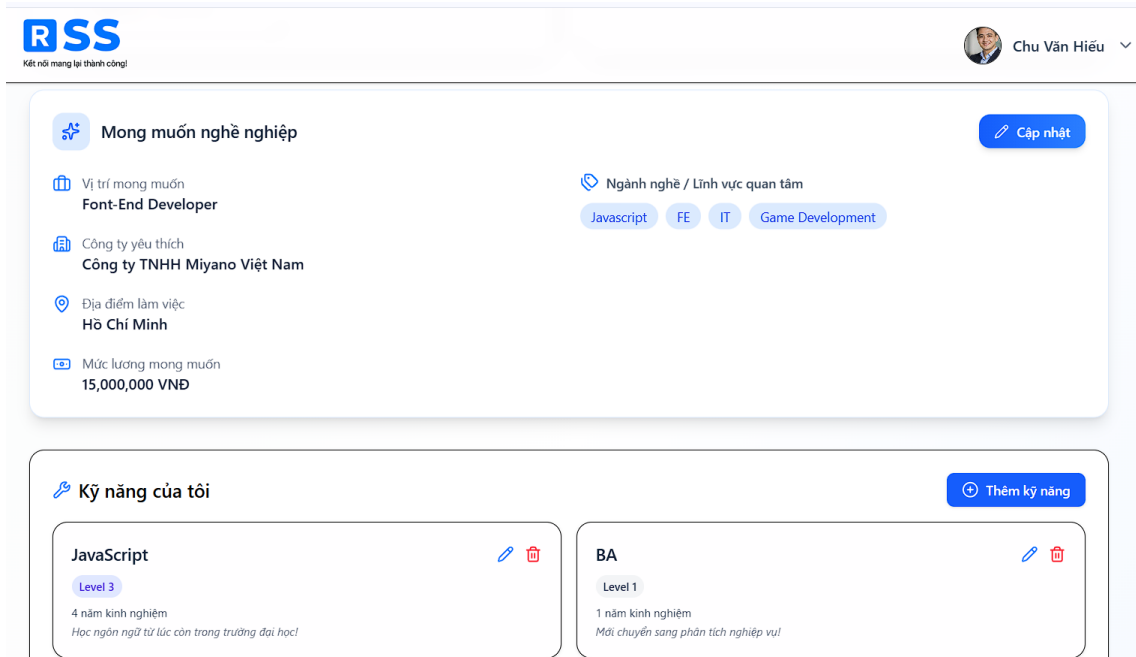
Hình 4.25: Giao diện "Quản lý đơn ứng tuyển"

Giao diện quản lý đơn ứng tuyển cho phép nhà tuyển dụng theo dõi tổng quan danh sách hồ sơ, phê duyệt/từ chối đơn ứng tuyển, đồng thời hiển thị mức độ phù hợp của từng ứng viên đối với vị trí tuyển dụng.



Hình 4.26: Giao diện "Nhu cầu tuyển dụng"

Nhà tuyển dụng cần khai báo nhu cầu tuyển dụng như trong Hình 4.26 để hệ thống có đủ dữ liệu làm cơ sở tạo các đề xuất phù hợp.



RSS
Kết nối mạng lại thành công!

Chu Văn Hiếu ▾

Mong muốn nghề nghiệp [Cập nhật](#)

Vị trí mong muốn
Font-End Developer

Công ty yêu thích
Công ty TNHH Miyano Việt Nam

Địa điểm làm việc
Hồ Chí Minh

Mức lương mong muốn
15,000,000 VND

Ngành nghề / Lĩnh vực quan tâm
Javascript FE IT Game Development

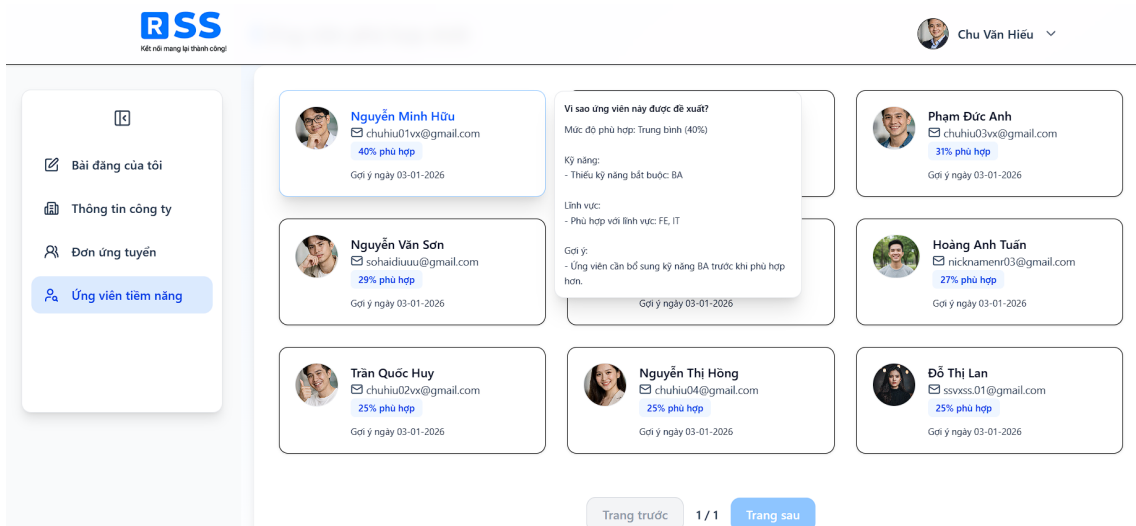
Kỹ năng của tôi [Thêm kỹ năng](#)

JavaScript
Level 3
4 năm kinh nghiệm
Học ngôn ngữ từ lúc còn trong trường đại học!

BA
Level 1
1 năm kinh nghiệm
Mới chuyển sang phân tích nghiệp vụ!

Hình 4.27: Giao diện "Mong muốn nghề nghiệp"

Tương tự nhà tuyển dụng, ứng viên cũng cần khai báo mong muốn nghề nghiệp và kỹ năng của bản thân như trong Hình 4.27 để hệ thống có đủ dữ liệu nhằm đưa ra các đề xuất phù hợp hơn.



RSS
Kết nối mạng lại thành công!

Chu Văn Hiếu ▾

Bài đăng của tôi
Thông tin công ty
Đơn ứng tuyển
Ứng viên tiềm năng

Nguyễn Minh Hữu
chuhuu01vx@gmail.com
40% phù hợp
Gợi ý ngày 03-01-2026

Phạm Đức Anh
chuhuu03vx@gmail.com
31% phù hợp
Gợi ý ngày 03-01-2026

Nguyễn Văn Sơn
sohaiduuu@gmail.com
29% phù hợp
Gợi ý ngày 03-01-2026

Hoàng Anh Tuấn
nicknamet03@gmail.com
27% phù hợp
Gợi ý ngày 03-01-2026

Trần Quốc Huy
chuhuu02vx@gmail.com
25% phù hợp
Gợi ý ngày 03-01-2026

Nguyễn Thị Hồng
chuhuu04@gmail.com
25% phù hợp
Gợi ý ngày 03-01-2026

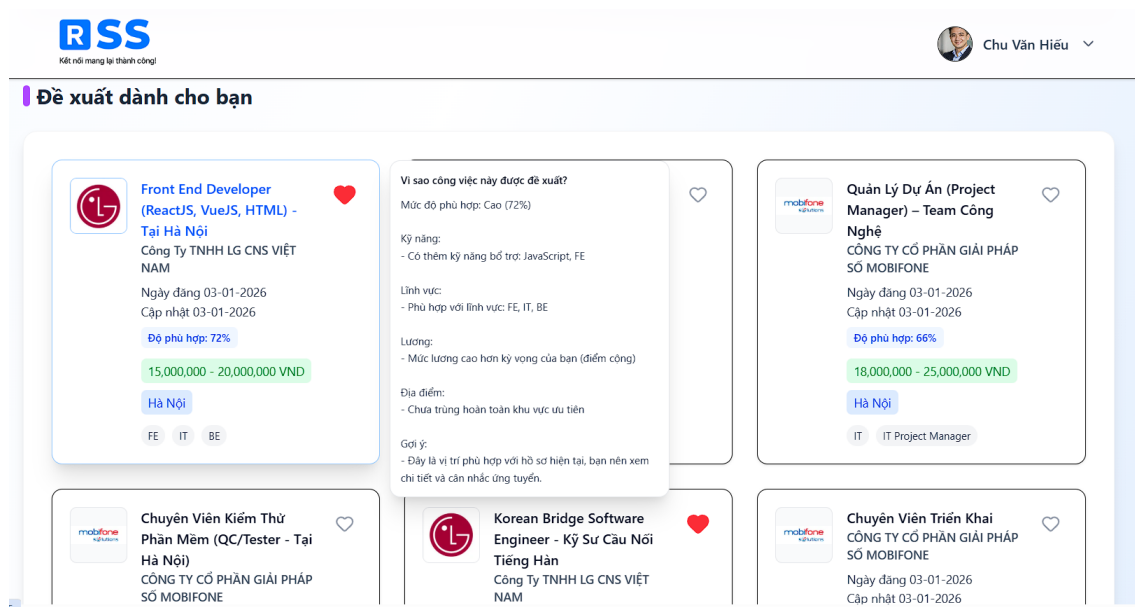
Đỗ Thị Lan
svxss.01@gmail.com
25% phù hợp
Gợi ý ngày 03-01-2026

Vi sao ứng viên này được đề xuất?
Mức độ phù hợp: Trung bình (40%)
Kỹ năng:
- Thiếu kỹ năng bắt buộc: BA
Lĩnh vực:
- Phù hợp với lĩnh vực: FE, IT
Gợi ý:
- Ứng viên cần bổ sung kỹ năng BA trước khi phù hợp hơn.
Gợi ý ngày 03-01-2026

Trang trước 1/1 Trang sau

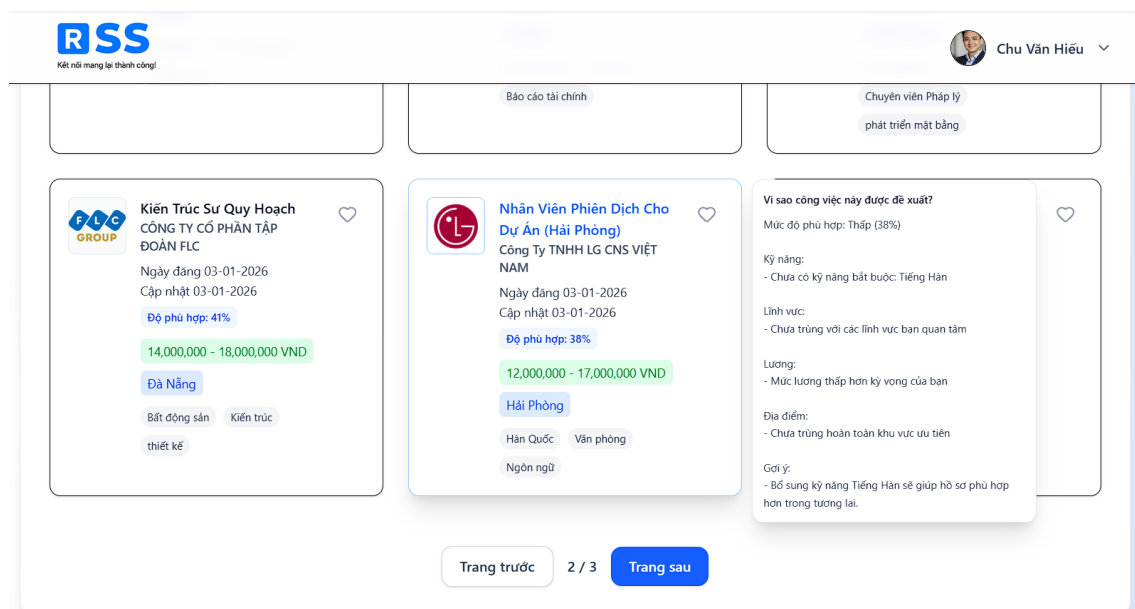
Hình 4.28: Giao diện "Đề xuất ứng viên"

Nhà tuyển dụng có thể xem danh sách ứng viên được đề xuất thông qua giao diện như Hình 4.28. Danh sách hiển thị các thông tin gồm: hồ sơ ứng viên, mức độ phù hợp và lý do đề xuất.



Hình 4.29: Giao diện "Trang chủ xem tin tuyển dụng"

Hình 4.29 minh họa một phần trên trang chủ, hiển thị danh sách các công việc được đề xuất và sắp xếp theo mức độ phù hợp giảm dần (từ cao đến thấp). Bên cạnh các thông tin cơ bản của công việc, giao diện còn hiển thị lý do đề xuất nhằm giải thích tính phù hợp và tăng mức độ thuyết phục ứng viên khi quyết định ứng tuyển.



Hình 4.30: Giao diện "Đề xuất tin tuyển dụng"

Tương tự Hình 4.29, giao diện đề xuất tin tuyển dụng hiển thị đầy đủ thông tin liên quan và trình bày danh sách công việc theo cơ chế phân trang, giúp người dùng theo dõi và tra cứu một cách rõ ràng, thuận tiện.

4.4 Kiểm thử

Hệ thống sử dụng kỹ thuật kiểm thử hộp đen để thiết kế các trường hợp kiểm thử. Trong quá trình kiểm thử, các trường hợp kiểm thử được xây dựng dựa trên dữ liệu đầu vào và kết quả đầu ra của hệ thống, sau đó so sánh kết quả thực tế với kết quả kỳ vọng nhằm đánh giá mức độ đúng đắn của từng trường hợp kiểm thử.

4.4.1 Kiểm thử chức năng “Tạo tin tuyển dụng”

Bảng 4.28: Test case kiểm thử chức năng tạo tin tuyển dụng

Mã test case	Test case	Dữ liệu đầu vào	Kết quả mong muốn	Kết quả
TC001	Chưa có công ty	Thực hiện tạo tin tuyển dụng khi người dùng chưa tạo công ty	Hệ thống hiển thị thông báo lỗi yêu cầu tạo công ty trước khi đăng tin	Đạt
TC002	Công ty chưa xác thực	Thực hiện tạo tin tuyển dụng khi công ty chưa được xác thực	Hệ thống hiển thị thông báo lỗi công ty chưa được xác thực	Đạt
TC003	Nhập thiếu thông tin	Nhập thiếu trường tiêu đề khi tạo tin tuyển dụng	Hệ thống hiển thị cảnh báo yêu cầu điền đầy đủ các trường bắt buộc	Đạt
TC004	Tạo tin tuyển dụng	Nhập đầy đủ các thông tin hợp lệ để tạo tin tuyển dụng	Hệ thống tạo tin tuyển dụng thành công và hiển thị tin với trạng thái “Chờ duyệt”	Đạt

4.4.2 Kiểm thử chức năng “Đề xuất công việc”

Bảng 4.29: Test case kiểm thử chức năng đề xuất công việc

Mã test case	Test case	Dữ liệu đầu vào	Kết quả mong muốn	Kết quả
TC001	Cập nhật đề xuất	Thay đổi thông tin mong muốn của ứng viên	Danh sách công việc đề xuất được cập nhật tương ứng	Đạt
TC002	Sinh đề xuất công việc	Dữ liệu của ứng viên và công việc	Hiển thị các công việc được đề xuất tại trang chủ	Đạt

Mã test case	Test case	Dữ liệu đầu vào	Kết quả mong muốn	Kết quả
TC003	Danh sách đề xuất	Dữ liệu đề xuất công việc	Hiển thị danh sách công việc theo điểm phù hợp giảm dần	Đạt
TC004	Sinh lý do đề xuất	Dữ liệu của ứng viên và công việc	Hiển thị lý do đề xuất cho từng công việc	Đạt
TC005	Tính điểm phù hợp	Ứng viên tạo các sự kiện như ứng tuyển, yêu thích, xem tin tuyển dụng	Điểm phù hợp của các đề xuất thay đổi tương ứng	Đạt
TC006	Thông báo đề xuất	Dữ liệu đề xuất công việc	Ứng viên nhận được email thông báo đề xuất	Đạt

4.4.3 Kiểm thử chức năng “Đề xuất ứng viên”

Bảng 4.30: Bảng kiểm thử chức năng đề xuất ứng viên

Mã test case	Test case	Dữ liệu đầu vào	Kết quả mong muốn	Kết quả
TC001	Cập nhật đề xuất	Thay đổi mong muốn tuyển dụng của nhà tuyển dụng	Danh sách ứng viên được đề xuất thay đổi tương ứng	Đạt
TC002	Sinh đề xuất ứng viên	Dữ liệu của ứng viên và nhà tuyển dụng	Hiển thị các ứng viên được đề xuất tại trang nhà tuyển dụng	Đạt
TC003	Danh sách đề xuất	Dữ liệu đề xuất ứng viên	Hiển thị danh sách ứng viên theo điểm phù hợp giảm dần	Đạt
TC004	Sinh lý do đề xuất	Dữ liệu của ứng viên và nhà tuyển dụng	Hiển thị lý do đề xuất cho từng ứng viên	Đạt
TC005	Thông báo đề xuất	Dữ liệu đề xuất ứng viên	Nhà tuyển dụng nhận được email thông báo đề xuất	Đạt

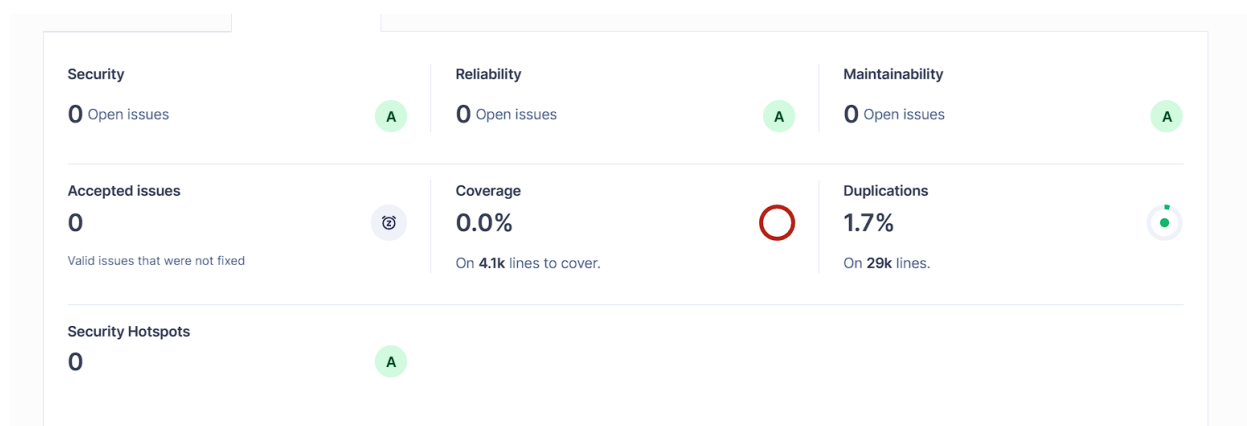
4.5 Triển khai

Hệ thống được triển khai trên máy tính có cấu hình.

Bảng 4.31: Cấu hình môi trường thực nghiệm

Thông số	Mô tả
CPU	AMD Ryzen 5 5600H
Hệ điều hành	Windows 11
RAM	20 GB
Bộ nhớ	512 GB SSD

Ngoài ra, Hệ thống sử dụng công cụ sonarqube để đánh giá mã nguồn và Light-house để đo kết quả thử nghiệm.

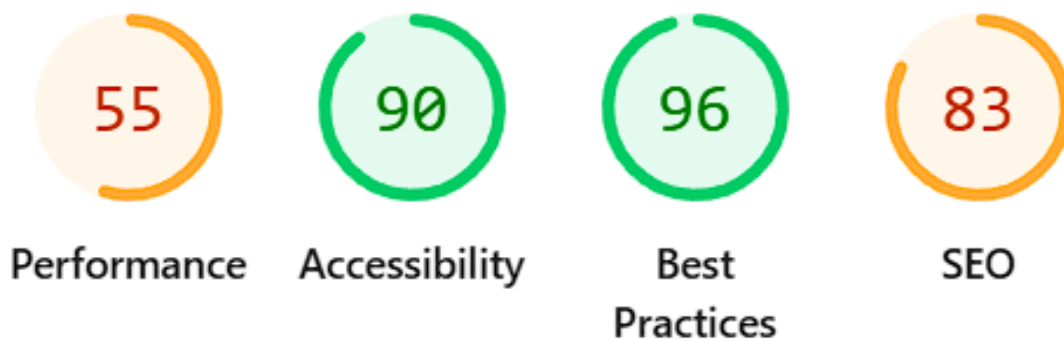


Hình 4.31: Kết quả đo bằng công cụ Sonarqube

SonarQube đánh giá chất lượng mã nguồn dựa trên các tiêu chí chính bao gồm: *Security* (bảo mật), *Reliability* (độ tin cậy), *Maintainability* (khả năng bảo trì), *Coverage* (độ bao phủ kiểm thử) và *Duplications* (mức độ trùng lặp mã nguồn).

Kết quả đánh giá cho thấy hệ thống không phát hiện các vấn đề nghiêm trọng liên quan đến bảo mật, độ tin cậy và khả năng bảo trì, với số lượng lỗi mở (Open Issues) bằng 0 và mức đánh giá đạt mức A ở các tiêu chí này. Điều này cho thấy mã nguồn được tổ chức tốt, tuân thủ các nguyên tắc lập trình an toàn và dễ dàng mở rộng, bảo trì trong tương lai.

Hệ thống tự triển khai các bài test thử công nên độ bao phủ kiểm thử (Coverage) của hệ thống hiện tại ở mức 0%. Tỷ lệ trùng lặp mã nguồn (Duplications) ở mức 1.7% trên tổng số 29.000 dòng mã cho thấy mức độ trùng lặp thấp và chấp nhận được.



Hình 4.32: Kết quả đo bằng công cụ Lighthouse

Lighthouse đánh giá ứng dụng web dựa trên bốn tiêu chí chính gồm: *Performance* (hiệu năng), *Accessibility* (khả năng truy cập), *Best Practices* (thực hành tốt) và *SEO* (tối ưu hóa công cụ tìm kiếm).

Kết quả đánh giá cho thấy hệ thống đạt điểm *Accessibility* là 90 và *Best Practices* là 96, phản ánh giao diện người dùng được xây dựng tương đối thân thiện, tuân thủ các nguyên tắc truy cập cơ bản và các thực hành phát triển web hiện đại.

Chỉ số *SEO* đạt 83 cho thấy hệ thống đã đáp ứng tốt các tiêu chí cơ bản về tối ưu hóa công cụ tìm kiếm, giúp cải thiện khả năng hiển thị của website trên các nền tảng tìm kiếm. Chỉ số *Performance* đạt 55 điểm cho thấy hiệu năng tải trang của hệ thống chưa được tối ưu, chủ yếu do các tệp hình ảnh chưa được xử lý và tối ưu hiệu quả, ảnh hưởng đến tốc độ tải trang.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Đối với hệ thống đề xuất, phương pháp và cách thức đề xuất chính là yếu tố cốt lõi, thể hiện tính “thông minh” của hệ thống. Website tuyển dụng thông minh trong đồ án sử dụng phương pháp lọc dựa trên nội dung (Content-based Filtering – CBF) để đánh giá và so sánh mức độ phù hợp giữa các đối tượng, bao gồm ứng viên và tin tuyển dụng. Trong quá trình triển khai, một trong những khó khăn lớn nhất là việc chuẩn hóa và biểu diễn dữ liệu của người dùng và tin tuyển dụng nhằm đảm bảo quá trình so khớp và tính toán độ phù hợp được thực hiện chính xác và hiệu quả.

5.1 Chuẩn hóa dữ liệu

5.1.1 Đặt vấn đề

Đối với hệ thống tuyển dụng thông minh, dữ liệu đầu vào đóng vai trò đặc biệt quan trọng trong quá trình xây dựng và vận hành hệ thống đề xuất. Các dữ liệu này đến từ nhiều nguồn khác nhau như hồ sơ ứng viên, mong muốn nghề nghiệp, kỹ năng, hành vi tương tác của người dùng và thông tin từ các tin tuyển dụng. Tuy nhiên, các dữ liệu thu thập được thường không đồng nhất về định dạng, cấu trúc và mức độ đầy đủ, gây khó khăn cho quá trình so sánh và đánh giá mức độ phù hợp giữa ứng viên và tin tuyển dụng.

Trong hệ thống đề xuất sử dụng phương pháp lọc dựa trên nội dung (Content-based Filtering), việc so khớp các đặc trưng giữa các đối tượng là yếu tố cốt lõi. Do đó, nếu dữ liệu không được chuẩn hóa một cách thống nhất, kết quả tính toán độ phù hợp có thể thiếu chính xác hoặc không phản ánh đúng năng lực của ứng viên cũng như yêu cầu của nhà tuyển dụng. Vì vậy, bài toán chuẩn hóa dữ liệu được đặt ra như một bước tiền xử lý bắt buộc nhằm đảm bảo chất lượng và hiệu quả của hệ thống đề xuất.

5.1.2 Giải pháp

Dựa trên phương pháp lọc dựa trên nội dung (Content-based Filtering), hệ thống đề xuất trong đồ án biểu diễn cả ứng viên và tin tuyển dụng dưới dạng các vector đặc trưng có cùng cấu trúc. Việc sử dụng mô hình vector giúp hệ thống dễ dàng so sánh và tính toán mức độ tương đồng giữa các đối tượng, từ đó đưa ra kết quả đề xuất phù hợp.

Các vector đặc trưng được xây dựng từ những thuộc tính quan trọng nhất trong bài toán tuyển dụng, bao gồm: kỹ năng (skills), lĩnh vực hoặc ngành nghề quan tâm (tags), mức lương và địa điểm làm việc. Mỗi đặc trưng được gán một trọng số thể

hiện mức độ quan trọng hoặc mức độ phù hợp của đặc trưng đó đối với từng đối tượng.

a, Ứng viên

Trước khi thực hiện quá trình chuẩn hóa, dữ liệu của ứng viên được thu thập từ hai nguồn chính là hành vi cá nhân và thông tin mong muốn nghề nghiệp. Trong đó, hành vi cá nhân bao gồm các sự kiện được hệ thống ghi nhận trong quá trình ứng viên tương tác với nền tảng, chẳng hạn như ứng tuyển vào một công việc, thêm hoặc loại bỏ một công việc khỏi danh sách yêu thích, cũng như xem chi tiết thông tin của một tin tuyển dụng. Các sự kiện này lưu lại thông tin về những tin tuyển dụng mà ứng viên đã tương tác, từ đó được tổng hợp và chuyển đổi thành các đặc trưng tương ứng trong vector biểu diễn ứng viên.

Hành vi người dùng trong hệ thống được xây dựng dựa trên việc tổng hợp thông tin từ các tin tuyển dụng mà ứng viên đã tương tác, kết hợp với loại sự kiện phát sinh trong quá trình sử dụng hệ thống. Các dữ liệu này được hệ thống ghi nhận và lưu trữ trong bảng UserInterestHistory. Mỗi bản ghi hành vi phản ánh một lần tương tác cụ thể giữa ứng viên và một tin tuyển dụng, bao gồm các thông tin như mã ứng viên, mã tin tuyển dụng, tiêu đề công việc, địa điểm, mức lương trung bình, các lĩnh vực liên quan và loại sự kiện tương tác.

Các loại sự kiện hành vi phổ biến bao gồm: ứng viên xem chi tiết một tin tuyển dụng (viewed), ứng viên ứng tuyển vào công việc (applied), hoặc thêm và loại bỏ tin tuyển dụng khỏi danh sách yêu thích (favorite). Thông qua việc ghi nhận các sự kiện này, hệ thống có thể xác định mức độ quan tâm của ứng viên đối với từng tin tuyển dụng cụ thể.

Ví dụ, một bản ghi trong bảng UserInterestHistory của ứng viên được biểu diễn như sau:

```
{
  "id": 349,
  "user_id": 2,
  "job_id": 37,
  "job_title": "Chuyên Viên Kiểm Thử Phần Mềm (QC/Tester -
  Tại Hà Nội)",
  "location": "Hà Nội",
  "avg_salary": 25000000,
  "tags": [
    { "name": "IT" },
    { "name": "Tester" },
```



```
{ "name": "QC" }  
],  
"source": "viewed",  
"event_type": "open_detail"  
}
```

Ngoài hành vi, ứng viên còn có một bảng dữ liệu ghi nhận mong muốn người dùng:

```
{  
  "user_id": 1,  
  "desired_title": "Front-End Developer",  
  "desired_company": "Công ty TNHH Miyano Việt Nam",  
  "desired_location": "Hồ Chí Minh",  
  "desired_salary": 15000000,  
  "tags": [  
    { "id": 27, "name": "Javascript" },  
    { "id": 28, "name": "FE" },  
    { "id": 40, "name": "IT" },  
    { "id": 41, "name": "Game Development" }  
  ]  
}
```

Từ hai nguồn dữ liệu trên kết hợp với dữ liệu về kỹ năng của người dùng, hệ thống tiến hành tổng hợp và sinh ra dữ liệu đặc trưng cho từng ứng viên dựa trên công thức tính toán cụ thể được trình bày tại Mục 2, Chương 5. Tập dữ liệu này nhằm phản ánh một cách toàn diện cả mong muốn nghề nghiệp khai báo và hành vi tương tác thực tế của ứng viên.

b, Nhà tuyển dụng

Nhà tuyển dụng cũng là một người dùng trong hệ thống, tuy nhiên có vai trò khác với ứng viên, do đó được biểu diễn bằng một vector đặc trưng riêng. Vector này được xây dựng dựa trên dữ liệu về nhu cầu tuyển dụng do nhà tuyển dụng khai báo, nhằm phản ánh các yêu cầu và tiêu chí tuyển chọn ứng viên cho từng vị trí. Dữ liệu nhu cầu tuyển dụng sau khi được chuẩn hóa sẽ được sử dụng để xây dựng vector biểu diễn nhà tuyển dụng. Một ví dụ về dữ liệu nhu cầu tuyển dụng được trình bày như sau:

```
{
```

```

"user_id": 1,
"desired_location": "Hồ Chí Minh",
"desired_salary_avg": 20000000,
"desired_tags": [
  { "id": 2, "name": "ReactJS" },
  { "id": 14, "name": "React" },
  { "id": 28, "name": "FE" },
  { "id": 40, "name": "IT" }
],
"required_skills": [
  { "id": 18, "name": "BA", "years_required": 1,
    "must_have": true },
  { "id": 21, "name": "FE", "years_required": 2,
    "must_have": false }
]
}

```

c, Tin tuyển dụng

Tin tuyển dụng được biểu diễn dựa trên các thông tin tuyển dụng do nhà tuyển dụng khai báo cho từng vị trí công việc. Hệ thống sử dụng các trường dữ liệu này để xây dựng vector đặc trưng tương ứng cho mỗi tin tuyển dụng, bao gồm các yêu cầu về kỹ năng, lĩnh vực, địa điểm làm việc và mức lương.

Các vector của ứng viên và nhà tuyển dụng được chuẩn hóa theo cùng cấu trúc với vector tin tuyển dụng. Trên cơ sở đó, hệ thống áp dụng phương pháp lọc dựa trên nội dung (Content-based Filtering – CBF) để tính toán mức độ phù hợp và sinh ra lý do phù hợp giữa các đối tượng được trình bày tại Mục 3, Chương 5.

```

{
  "vector": {
    "job_id": 33,
    "skill_profile": [
      { "id": 1, "must": true, "weight": 0.44 },
      { "id": 21, "must": true, "weight": 0.48 }
    ],
    "tag_profile": [
      { "id": 28, "weight": 1 }
    ],
  },
}

```

```
"title_keywords": null,
"location": "HN",
"salary_avg": 14000000,
"updated_at": "2026-01-18T05:47:16.740Z"
}
}
```

Trường *title_keywords* trong vector đặc trưng được tính toán bằng phương pháp TF-IDF nhưng chưa được sử dụng trong đồ án do chưa hoàn thiện, và được định hướng phát triển trong các phiên bản tiếp theo của hệ thống.

5.1.3 Kết quả thực hiện

Các vector đặc trưng trong hệ thống được xây dựng theo cùng một cấu trúc nhằm đảm bảo tính nhất quán và thuận tiện cho việc so sánh giữa các đối tượng. Các trường dữ liệu chính trong mỗi vector bao gồm: mã định danh (*id*), kỹ năng (*skills*), lĩnh vực hoặc ngành nghề (*tags*), từ khóa chức danh (*title_keywords*), địa điểm làm việc (*location*) và mức lương trung bình (*salary_avg*).

```
{
  "user_id": 1,
  "skill_profile": [
    { "id": 1, "w": 0.64 },
    { "id": 18, "w": 0.36 },
    { "id": 21, "w": 0.52 }
  ],
  "tag_profile": [
    { "id": 95, "weight": 0.4476 },
    { "id": 28, "weight": 0.7344 },
    { "id": 40, "weight": 1 }
  ],
  "title_keywords": [
    { "keyword": "reactjs", "weight": 1 },
    { "keyword": "vuejs", "weight": 1 }
  ],
  "preferred_location": "HCM",
  "salary_expected": 17037818,
  "updated_at": "2026-01-18T06:18:00.907Z"
}
```

5.2 Tính hành vi người dùng

5.2.1 Đặt vấn đề

Như đã trình bày tại Mục 1, Chương 5, dữ liệu của ứng viên bao gồm nhiều thành phần khác nhau nhằm phản ánh đầy đủ mong muốn và đặc điểm của ứng viên, bao gồm dữ liệu hành vi và dữ liệu mong muốn nghề nghiệp. Do đó, cần có một phương pháp tính toán và tổng hợp các nguồn dữ liệu này thành một vector chuẩn cho mỗi ứng viên, làm cơ sở cho quá trình so khớp và đề xuất trong hệ thống.

5.2.2 Giải pháp

Dữ liệu hành vi của người dùng được thu thập từ nhiều nguồn khác nhau, trong đó mỗi loại hành vi phản ánh mức độ quan tâm của ứng viên ở các mức độ khác nhau. Do đó, hệ thống gán các trọng số khác nhau cho từng loại hành vi nhằm phản ánh đúng mức độ ảnh hưởng của chúng trong quá trình xây dựng vector đặc trưng của ứng viên.

Các trọng số này được xác định dựa trên kết quả khảo sát một số người dùng kết hợp với nhu cầu và phạm vi triển khai của đề án. Cụ thể, hệ thống sử dụng các trọng số như sau: ứng viên ứng tuyển vào công việc có trọng số là 5, ứng viên đánh dấu yêu thích công việc có trọng số là 3, ứng viên xem tin tuyển dụng có trọng số là 1, và dữ liệu mong muốn nghề nghiệp do người dùng khai báo có trọng số là 4.

Từ các trọng số hành vi đã xác định, hệ thống tiến hành tính toán trọng số cho các trường dữ liệu tương ứng trong vector đặc trưng của ứng viên. Cách tính được thực hiện cho từng trường dữ liệu như sau:

- (i) **Từ khóa (*title_keywords*)**: Các từ khóa được tổng hợp từ tiêu đề công việc và tên công ty trong bảng dữ liệu hành vi. Phương pháp TF-IDF được sử dụng để xác định trọng số của các từ khóa này. Tuy nhiên, trong phạm vi đề án, trường dữ liệu này chưa được áp dụng để tính toán mức độ phù hợp và được định hướng phát triển trong các phiên bản tiếp theo của hệ thống.
- (ii) **Địa điểm (*location*)**: Địa điểm làm việc được xác định dựa trên tổng trọng số từ các hành vi liên quan. Ví dụ, ứng viên có các hành vi như: yêu thích công việc tại TP. Hồ Chí Minh, xem công việc tại Hà Nội và khai báo mong muốn làm việc tại Hà Nội. Khi đó, Hà Nội có tổng trọng số là 6, lớn hơn tổng trọng số 5 của TP. Hồ Chí Minh, do đó Hà Nội được chọn làm địa điểm đặc trưng của ứng viên.
- (iii) **Mức lương (*salary*)**: Mức lương được tính toán dựa trên trọng số hành vi, tuy nhiên có sự điều chỉnh so với cách cộng trọng số thông thường. Ví dụ (đơn vị:

triệu đồng), với các hành vi gồm: yêu thích mức lương 20, mong muốn mức lương 10 và ứng tuyển mức lương 12, giá trị mức lương đặc trưng được tính theo công thức:

$$0.7 \times \frac{20 \times 3 + 12 \times 5}{3 + 5} + 0.3 \times 10 = 13.5$$

Giá trị này phản ánh sự kết hợp giữa hành vi thực tế và mong muốn khai báo của ứng viên.

- (iv) **Ngành nghề (tags):** Tương tự như cách xác định địa điểm làm việc, các ngành nghề được tính toán dựa trên tổng trọng số từ các hành vi tương tác của ứng viên. Hệ thống lựa chọn ba ngành nghề có tổng trọng số cao nhất làm các ngành nghề đặc trưng trong vector biểu diễn ứng viên.

5.2.3 Kết quả thực hiện

Dựa trên các phương pháp tính toán và trọng số đã trình bày, hệ thống tiến hành tổng hợp và xây dựng dữ liệu hành vi của người dùng dưới dạng cấu trúc chuẩn hóa như sau:

```
{
  "user_id": 1,
  "avg_salary": 17037818,
  "main_location": "Hà Nội",
  "tags": [
    { "name": "IT", "weight": 1 },
    { "name": "FE", "weight": 0.7343 },
    { "name": "BE", "weight": 0.4478 }
  ],
  "keywords": [
    { "name": "reactjs", "weight": 1 },
    { "name": "vuejs", "weight": 1 }
  ]
}
```

5.3 Xây dựng công thức tính điểm phù hợp

5.3.1 Đặt vấn đề

Trong hệ thống tuyển dụng thông minh, việc xác định mức độ phù hợp giữa các đối tượng đóng vai trò then chốt, quyết định trực tiếp đến chất lượng của chức năng đề xuất. Điểm phù hợp (*Fit Score*) được sử dụng như một thước đo định lượng

nhằm đánh giá mức độ đáp ứng giữa hồ sơ ứng viên và yêu cầu của tin tuyển dụng, cũng như giữa ứng viên và nhu cầu tuyển dụng của nhà tuyển dụng. Bên cạnh việc tính toán điểm phù hợp, hệ thống còn sinh ra lý do đề xuất tương ứng với từng vai trò người dùng, qua đó giúp nâng cao trải nghiệm người dùng và tăng thuyết phục của các kết quả đề xuất.

Trong phần này, đồ án tập trung trình bày phương pháp và công thức tính toán điểm phù hợp dựa trên phương pháp lọc dựa trên nội dung (Content-based Filtering), làm cơ sở cho việc sắp xếp và sinh lý do đề xuất trong hệ thống.

5.3.2 Giải pháp

Để triển khai hai hướng đề xuất trên, hệ thống xét đến ba đối tượng liên quan gồm: ứng viên, nhà tuyển dụng và tin tuyển dụng. Mỗi đối tượng có các tiêu chí quan tâm khác nhau, do đó công thức tính điểm phù hợp được thiết kế linh hoạt theo từng ngữ cảnh so khớp.

Trên cơ sở đó, điểm phù hợp được tổng hợp từ các yếu tố quan trọng như mức độ đáp ứng về kỹ năng (bao gồm kỹ năng bắt buộc và kỹ năng bổ trợ), mức độ phù hợp về lĩnh vực và định hướng nghề nghiệp, cùng với các điều kiện ưu tiên như địa điểm làm việc và mức lương (trong trường hợp dữ liệu cho phép). Đồng thời, hệ thống sinh ra các lý do đề xuất tương ứng nhằm giúp người dùng hiểu rõ các điểm mạnh, các yếu tố còn thiếu và các gợi ý cải thiện, qua đó làm cho kết quả đề xuất trở nên minh bạch và thuyết phục hơn.

a, Dữ liệu đầu vào

Đối với ứng viên, ngoài dữ liệu hành vi đã trình bày tại Mục 2, Chương 5, vector ứng viên (*User Vector*) cần được bổ sung thêm hồ sơ kỹ năng (*skill profile*) của ứng viên, được lấy từ bảng lưu trữ kỹ năng trong hệ thống để hoàn thiện vector ứng viên.

Mỗi kỹ năng của ứng viên được gán một trọng số $w \in [0, 1]$, phản ánh mức độ phù hợp của kỹ năng đó. Trọng số này được xác định dựa trên mức trình độ (*level*) và số năm kinh nghiệm (*years*) mà ứng viên khai báo, qua đó giúp vector người dùng phản ánh chính xác hơn năng lực thực tế của ứng viên.

Trước hết, mỗi mức độ thành thạo được ánh xạ với một số năm kinh nghiệm tối đa tương ứng, dùng làm ngưỡng chuẩn. Trọng số kỹ năng được tính theo công thức:

$$w = 0.8 \times \frac{level}{5} + 0.2 \times \left(\min \left(\frac{years}{maxYears}, 1 \right) + bonus \right)$$

Trong đó:

- w : trọng số của kỹ năng, có giá trị trong khoảng $[0, 1]$, tối đa là 1 nếu vượt quá 1;
- $level$: cấp độ kỹ năng do ứng viên khai báo, với giá trị từ 1 đến 5;
- $years$: số năm kinh nghiệm thực tế của ứng viên đối với kỹ năng tương ứng;
- $maxYears$: số năm kinh nghiệm tối đa tương ứng với từng mức độ thành thạo, được sử dụng làm ngưỡng chuẩn để đánh giá kinh nghiệm;
- $bonus$: hệ số thưởng khi số năm kinh nghiệm vượt quá ngưỡng $maxYears$, có giá trị tối đa là 0.2 nhằm tránh làm mất cân bằng trọng số tổng thể.

Ví dụ: giả sử một ứng viên có kỹ năng với mức độ thành thạo $level = 3$ và số năm kinh nghiệm $years = 6$. Với mức $level = 3$, số năm kinh nghiệm tối đa tương ứng là $maxYears = 5$. Khi đó:

$$levelScore = \frac{3}{5} = 0.6, \quad experienceRatio = 1, \quad bonus = 0.05$$

Suy ra trọng số kỹ năng:

$$w = 0.8 \times 0.6 + 0.2 \times (1 + 0.05) = 0.69$$

Giá trị này phản ánh việc ứng viên có mức độ thành thạo tốt và kinh nghiệm vượt ngưỡng yêu cầu đối với kỹ năng tương ứng.

Đối với nhà tuyển dụng và tin tuyển dụng, dữ liệu sử dụng để xây dựng vector đặc trưng chỉ xuất phát từ một nguồn dữ liệu khai báo. Do đó, quá trình chuẩn hóa và xây dựng vector cho hai đối tượng này tương đối đơn giản, và cấu trúc vector thu được tương tự như kết quả đã trình bày tại Mục 1, Chương 5.

5.3.3 Công thức tính điểm

Điểm phù hợp (*Fit Score*) được tính toán dựa trên bốn yếu tố chính, bao gồm:

- S : kỹ năng (*SkillScore*);
- T : ngành nghề (*TagScore*);
- L : địa điểm làm việc (*LocationScore*);
- P : mức lương (*SalaryScore*).

a, So khớp kỹ năng

Xét hai tập kỹ năng cần so khớp, trong đó K_J là tập kỹ năng yêu cầu của đối tượng tham chiếu (tin tuyển dụng hoặc nhà tuyển dụng) và K_U là tập kỹ năng của đối tượng được so khớp (ứng viên). Mỗi kỹ năng $k \in K_J$ được gán một trọng số

yêu cầu $weight_k$ và được phân loại thành kỹ năng bắt buộc (*must*) hoặc kỹ năng bổ trợ (*optional*).

Trọng số tối đa của mỗi kỹ năng k được xác định như sau:

$$W_k = \begin{cases} 5 \cdot weight_k, & \text{nếu } k \text{ là kỹ năng bắt buộc,} \\ 1 \cdot weight_k, & \text{nếu } k \text{ là kỹ năng bổ trợ.} \end{cases}$$

Nếu đối tượng U không có kỹ năng bắt buộc k , điểm kỹ năng được gán bằng 0:

$$P_k = 0$$

Nếu đối tượng U có kỹ năng bắt buộc k với trọng số w_k^U , điểm kỹ năng được tính theo:

$$base_k = \min \left(\frac{w_k^U}{weight_k}, 1 \right)$$

$$bonus_k = \begin{cases} \min \left((w_k^U - weight_k) \cdot 0.2, 0.2 \right), & w_k^U > weight_k, \\ 0, & \text{ngược lại} \end{cases}$$

$$P_k = (base_k + bonus_k) \cdot W_k$$

Trong đó, thành phần $base_k$ phản ánh mức độ đáp ứng yêu cầu của kỹ năng, còn $bonus_k$ là hệ số thưởng nhằm phản ánh trường hợp đối tượng U có mức độ kỹ năng vượt quá yêu cầu ban đầu.

Nếu đối tượng U không có kỹ năng bổ trợ k :

$$P_k = 0$$

Nếu đối tượng U có kỹ năng bổ trợ k với trọng số w_k^U , điểm kỹ năng được xác định bởi:

$$base_k = \min \left(\frac{w_k^U}{weight_k}, 1 \right)$$

$$bonus_k = \begin{cases} \min \left((w_k^U - weight_k) \cdot 0.1, 0.1 \right), & w_k^U > weight_k, \\ 0, & \text{ngược lại} \end{cases}$$

$$P_k = (base_k + bonus_k) \cdot W_k$$

Điểm so khớp kỹ năng được chuẩn hóa theo tỷ lệ giữa tổng điểm đạt được và

tổng điểm tối đa:

$$SkillScore = \frac{\sum_{k \in K_J} P_k}{\sum_{k \in K_J} W_k}$$

Giá trị *SkillScore* (viết tắt là *S*) thường nằm trong khoảng $[0, 1]$, và có thể tăng lên tối đa 1.2 trong trường hợp đối tượng được so khớp có mức độ kỹ năng vượt trội so với yêu cầu.

Ngoài điểm số tổng hợp, hệ thống ghi nhận các chỉ số thống kê sau để phục vụ việc sinh lý do đề xuất:

$$mustCount = |\{k \in K_J \mid k \text{ là kỹ năng bắt buộc}\}|$$

$$matchedMustCount = |\{k \in K_J \mid k \text{ là kỹ năng bắt buộc và } k \in K_U\}|$$

$$matchedOptionalCount = |\{k \in K_J \mid k \text{ là kỹ năng bổ trợ và } k \in K_U\}|$$

$$missingMust = \{k \in K_J \mid k \text{ là kỹ năng bắt buộc và } k \notin K_U\}$$

Các chỉ số này được sử dụng để sinh lý do đề xuất, giúp giải thích rõ những kỹ năng đã đáp ứng và những kỹ năng còn thiếu.

b, So khớp lĩnh vực/ngành nghề

Việc so khớp lĩnh vực/ngành nghề được thực hiện giữa hai tập tag: T_J là tập tag của đối tượng tham chiếu (tin tuyển dụng hoặc nhà tuyển dụng) và T_U là tập tag của đối tượng được so khớp (ứng viên). Mỗi tag $t \in T_J$ được gán trọng số w_t^J nhằm phản ánh mức độ quan trọng của lĩnh vực/ngành nghề đó đối với đối tượng tham chiếu; tương tự, mỗi tag $t \in T_U$ có trọng số w_t^U .

Trong trường hợp một trong hai đối tượng không có dữ liệu tag (tức $T_J = \emptyset$ hoặc $T_U = \emptyset$), hệ thống gán điểm nền $TagScore = 0.2$ để tránh triệt tiêu hoàn toàn ảnh hưởng của tiêu chí ngành nghề trong điểm phù hợp tổng thể.

Khi cả hai phía đều có dữ liệu tag, điểm so khớp được tính dựa trên mức độ giao nhau có xét trọng số. Với mỗi tag $t \in T_J$, điểm đạt được được xác định:

$$P_t = \begin{cases} \min(w_t^U, w_t^J), & \text{nếu } t \in T_U, \\ 0, & \text{ngược lại.} \end{cases}$$

Điểm tối đa tương ứng của tag đó là:

$$M_t = w_t^J.$$

Tổng điểm thô và tổng điểm tối đa lần lượt là:

$$P_{\text{raw}} = \sum_{t \in T_J} P_t, \quad M_{\text{raw}} = \sum_{t \in T_J} M_t.$$

Điểm thô được chuẩn hóa:

$$R = \frac{P_{\text{raw}}}{M_{\text{raw}}}.$$

Sau đó hệ thống áp dụng phép chuẩn hóa tuyến tính để đảm bảo tiêu chí ngành nghề luôn có ảnh hưởng tối thiểu:

$$\text{TagScore} = 0.2 + 0.8 \cdot R,$$

trong đó $\text{TagScore} \in [0.2, 1]$.

Ngoài điểm số, hệ thống ghi nhận danh sách các tag trùng khớp giữa hai đối tượng; để đảm bảo tính trực quan khi hiển thị, chỉ tối đa ba tag trùng khớp đầu tiên (theo thứ tự ưu tiên của T_J) được sử dụng như các lý do đề xuất.

c, So khớp địa điểm

Nếu một trong hai phía thiếu dữ liệu vị trí (ứng viên hoặc đối tượng tham chiếu), hệ thống gán điểm nền và trạng thái trung tính:

$$\text{LocScore} = 0.3, \quad \text{matched} = \text{false}, \quad \text{level} = \text{neutral}.$$

Nếu có đủ dữ liệu, hệ thống kiểm tra mức độ trùng khớp theo tiêu chí so khớp chuỗi (ví dụ: một chuỗi chứa chuỗi còn lại). Khi đó:

$$\text{LocScore} = \begin{cases} 1, & \text{nếu vị trí trùng khớp (match),} \\ 0.3, & \text{nếu không trùng khớp.} \end{cases}$$

Giá trị LocScore (viết tắt là L) nằm trong khoảng $[0.3, 1]$. Việc đặt ngưỡng nền 0.3 giúp điểm phù hợp tổng thể không bị giảm quá mạnh trong trường hợp địa điểm không trùng nhau hoặc thiếu dữ liệu vị trí.

d, So khớp mức lương

Tiêu chí so khớp mức lương được xây dựng dựa trên hai giá trị:

- S_E : mức lương kỳ vọng của ứng viên (*expected*);
- S_J : mức lương đại diện của đối tượng tham chiếu (tin tuyển dụng hoặc nhà tuyển dụng), ký hiệu *jobSalary*.

Hàm trả về bộ giá trị ($SalaryScore, level, comparable$), trong đó:

- $SalaryScore$: điểm so khớp lương;
- $level \in \{\text{near}, \text{lower}, \text{higher}, \text{null}\}$: nhãn giải thích quan hệ giữa S_J và S_E ;
- $comparable$: cờ logic cho biết hai giá trị có đủ dữ liệu để so sánh hay không.

Nếu một trong hai phía thiếu dữ liệu (tức S_E không tồn tại hoặc S_J không tồn tại), hệ thống không thể so sánh trực tiếp và gán trạng thái trung tính:

$$SalaryScore = 1, \quad level = \text{null}, \quad comparable = \text{false}.$$

Khi S_E và S_J đều tồn tại, đặt tỷ lệ:

$$r = \frac{S_J}{S_E}.$$

Khi đó $comparable = \text{true}$ và $SalaryScore$ được tính theo các quy tắc sau:

- Bằng nhau (near). Nếu $S_J = S_E$:

$$SalaryScore = 1, \quad level = \text{near}.$$

- Thấp hơn kỳ vọng (lower). Nếu $S_J < S_E$:

$$SalaryScore = \frac{S_J}{S_E} = r, \quad level = \text{lower}.$$

Giá trị này nằm trong $(0, 1)$ và phản ánh mức độ “thiếu hụt” của lương so với kỳ vọng.

- Cao hơn kỳ vọng (higher). Nếu $S_J > S_E$, hệ thống áp dụng cơ chế thưởng theo ngưỡng tỷ lệ r :

$$SalaryScore = \begin{cases} 1.1, & 1 < r \leq 1.25, \\ 1.25, & 1.25 < r \leq 1.5, \\ 1.3, & 1.5 < r \leq 1.75, \\ 1.4, & 1.75 < r \leq 2, \\ 1.5, & r > 2, \end{cases} \quad level = \text{higher}.$$

Như vậy, $SalaryScore$ có thể lớn hơn 1 trong trường hợp mức lương tham chiếu cao hơn kỳ vọng của ứng viên, nhằm thể hiện hiệu ứng “thưởng” khi cơ hội đáp

ứng tốt về mặt thu nhập.

e, Công thức tính FitScore theo từng ngữ cảnh đề xuất

Đánh giá mức phù hợp của ứng viên cho một tin tuyển dụng (ứng tuyển/đơn ứng tuyển). Trong ngữ cảnh này, tin tuyển dụng là đối tượng tham chiếu và mục tiêu là đánh giá ứng viên có đáp ứng yêu cầu của công việc hay không. Do đó tiêu chí kỹ năng được ưu tiên cao:

$$Fit_{U \rightarrow J} = 0.4S + 0.25T + 0.25P + 0.1L.$$

Gợi ý tin tuyển dụng cho ứng viên (đề xuất việc làm). Trong ngữ cảnh này, ứng viên là đối tượng tham chiếu và mục tiêu là đánh giá công việc có phù hợp mong muốn của ứng viên hay không. Vì vậy hệ thống tăng trọng số cho tag/ngành nghề và lương:

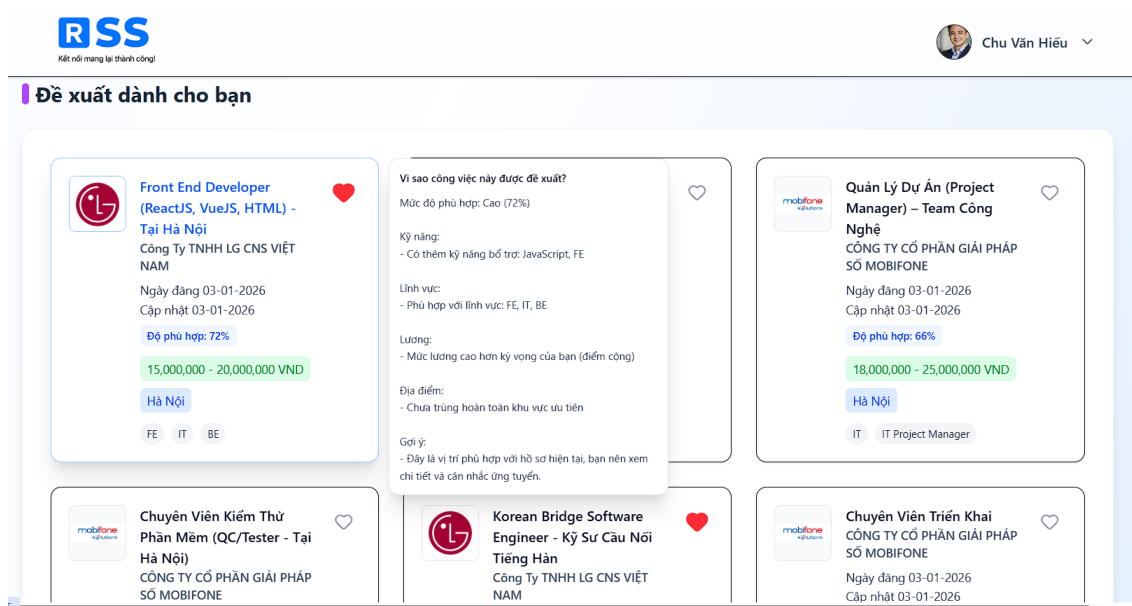
$$Fit_{J \rightarrow U} = 0.15S + 0.4T + 0.3P + 0.15L.$$

Gợi ý ứng viên cho nhà tuyển dụng (đề xuất ứng viên). Trong ngữ cảnh này, nhà tuyển dụng (hồ sơ nhu cầu tuyển dụng) là đối tượng tham chiếu và mục tiêu là đánh giá ứng viên có phù hợp với nhu cầu của nhà tuyển dụng hay không. Do đó kỹ năng được đặt trọng số cao nhất:

$$Fit_{U \rightarrow R} = 0.5S + 0.2T + 0.15P + 0.15L.$$

Trong cả ba công thức, các trọng số đều được chọn sao cho tổng bằng 1, nhằm đảm bảo FitScore là tổ hợp tuyến tính của các điểm thành phần và có thể so sánh trực tiếp giữa các đối tượng trong cùng một ngữ cảnh.

5.3.4 Kết quả thực hiện



Hình 5.1: Điểm phù hợp và lý do đề xuất

Hình 5.1 là là kết quả tính đề xuất một tin tuyển dụng cho ứng viên dựa theo công thức đã trình bày.

5.4 Gửi thông báo thông minh

5.4.1 Đặt vấn đề

Trong các hệ thống tuyển dụng, email là kênh tương tác phổ biến vì tận dụng được thông tin liên hệ mà người dùng đã đăng ký. Tuy nhiên, nếu hệ thống chỉ gửi email theo kiểu *thông báo thuần túy* mà không xét đến trạng thái và nhu cầu thực tế của người dùng, trải nghiệm sẽ nhanh chóng bị suy giảm.

Ví dụ, ứng viên đã tìm được việc làm nhưng vẫn tiếp tục nhận email giới thiệu công việc mới, gây cảm giác phiền hà. Tương tự, nhà tuyển dụng có thể nhận quá nhiều email không liên quan hoặc lặp lại, dẫn đến tình trạng *spam* và làm giảm mức độ tin cậy đối với hệ thống. Khi đó, hệ thống không còn được xem là một cơ chế đề xuất thông minh, mà chỉ là một kênh gửi thông tin đại trà.

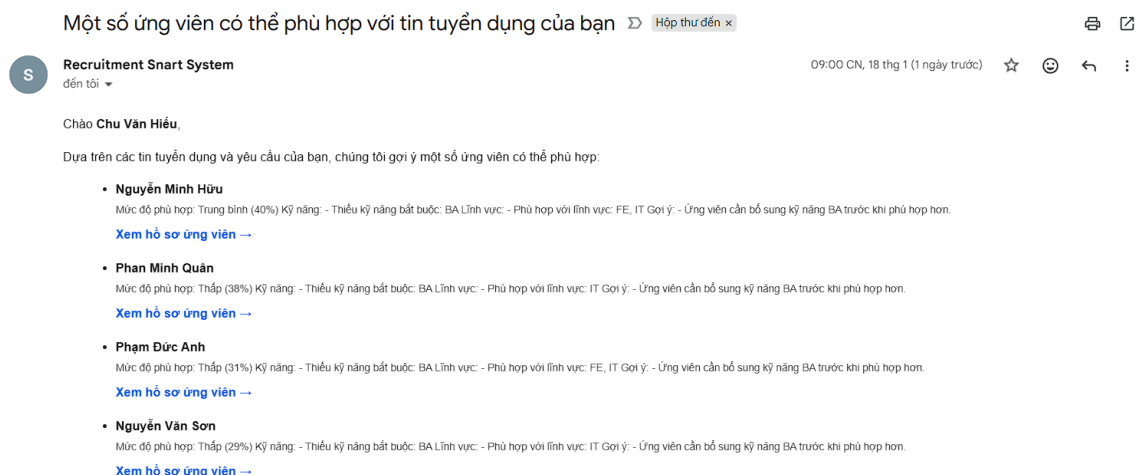
5.4.2 Giải pháp

Để đảm bảo email thực sự mang tính cá nhân hóa và hữu ích, hệ thống áp dụng các cơ chế sau:

- **Tránh gửi trùng lặp:** Hệ thống lưu vết các đề xuất đã gửi và gắn nhãn cho từng đối tượng đã được thông báo (tin tuyển dụng/ứng viên). Nhờ đó, cùng một nội dung sẽ không bị gửi lặp lại nhiều lần cho cùng một người dùng trong một khoảng thời gian nhất định.

- Cho phép người dùng chủ động quản lý thông báo: Hệ thống hỗ trợ bật/tắt nhận thông báo, đồng thời cho phép khóa tài khoản hoặc tạm dừng nhận email. Cơ chế này đảm bảo người dùng có quyền xác nhận rõ ràng về việc có muốn tiếp tục nhận thông báo từ hệ thống hay không.
- Tự động tắt thông báo khi người dùng không tương tác: Hệ thống theo dõi số lượng email đã gửi nhưng người dùng không có phản hồi gián tiếp (ví dụ: không đăng nhập/truy cập hệ thống trong một khoảng thời gian). Nếu vượt ngưỡng cho phép, hệ thống tự động tạm dừng gửi thông báo để hạn chế làm phiền.
- Xác thực hai chiều theo trạng thái nhận thông báo: Khi ứng viên hoặc nhà tuyển dụng tắt nhận thông báo, hệ thống không chỉ dừng gửi email cho người đó mà còn ngừng sử dụng hồ sơ của họ như một đối tượng để đề xuất cho phía còn lại. Cách tiếp cận này giúp tránh tình trạng đề xuất các đối tượng không có nhu cầu tương tác.
- Hai chế độ gửi: định kỳ và theo sự kiện: Hệ thống hỗ trợ gửi email định kỳ (tổng hợp theo thời gian) và gửi theo sự kiện (khi có tin tuyển dụng mới hoặc có đề xuất nổi bật). Việc kết hợp hai chế độ giúp hạn chế bỏ sót thông tin quan trọng nhưng vẫn kiểm soát tần suất gửi, đảm bảo thông báo đến đúng người quan tâm.

5.4.3 Kết quả thực hiện



Hình 5.2: Email thông báo cho nhà tuyển dụng

Hình 5.2 minh họa nội dung email thông báo mà nhà tuyển dụng nhận được từ hệ thống.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Thị trường tuyển dụng luôn gặp vấn đề về việc nhà tuyển dụng gặp khó khăn khi tìm ứng viên phù hợp với nhu cầu và người đi ứng tuyển không tìm thấy công việc phù hợp với mong muốn, khả năng của mình. Từ đó, sinh viên cho rằng nguyên nhân quan trọng nằm ở cách tiếp cận và xử lý thông tin của cả hai phía. Đây cũng là cơ sở để xây dựng một hệ thống website tuyển dụng thông minh theo hướng đề xuất.

Hiện nay trên thị trường có nhiều nền tảng tuyển dụng như TopCV, ITviec,... Các nền tảng này đem lại lợi ích lớn trong việc đăng tải và tiếp cận thông tin tuyển dụng, tuy nhiên cách tiếp cận phổ biến vẫn dựa nhiều vào mô hình đăng tin và tìm kiếm thủ công hoặc ưu tiên hiển thị theo gói dịch vụ/quảng cáo. Vì vậy, mức độ cá nhân hóa và khả năng gợi ý “đúng nhu cầu” giữa hai phía còn hạn chế, đặc biệt khi dữ liệu tăng nhanh và người dùng không có đủ thời gian sàng lọc.

Hệ thống trong đồ án tập trung vào mục tiêu chính là đề xuất, do đó đây vừa là điểm mạnh vừa là thách thức. Việc ước lượng mức độ phù hợp giữa nhiều đối tượng (ứng viên – tin tuyển dụng – nhà tuyển dụng) với số lượng thông tin rất lớn là vấn đề khó, đòi hỏi cơ chế biểu diễn dữ liệu và công thức tính điểm phù hợp có khả năng mở rộng và giải thích được.

6.2 Hướng phát triển

Mặc dù hệ thống đã đáp ứng các yêu cầu đặt ra ban đầu và đạt được mục tiêu cốt lõi là hỗ trợ đề xuất giữa ứng viên và nhà tuyển dụng, để có thể tiến tới triển khai thực tế hoặc thương mại hóa vẫn cần một quá trình cải tiến liên tục về cả chức năng, hiệu năng và trải nghiệm người dùng. Từ kinh nghiệm trong quá trình thực hiện đồ án, sinh viên xác định hai hướng phát triển chính để tăng tính hoàn thiện và khả năng cạnh tranh của hệ thống.

Thứ nhất, hệ thống hiện tại chủ yếu tập trung vào việc đưa thông tin và gợi ý phù hợp đến người dùng, nhưng chưa hỗ trợ đầy đủ một quy trình tuyển dụng trọn vẹn. Trong tương lai, hệ thống có thể mở rộng theo hướng cung cấp thêm các tiện ích và nghiệp vụ tuyển dụng, chẳng hạn như công cụ tạo và quản lý CV, hỗ trợ phỏng vấn trực tuyến, quản trị quy trình tuyển dụng theo từng doanh nghiệp, cũng như các chức năng theo dõi trạng thái ứng tuyển và đánh giá ứng viên. Việc bổ sung các thành phần này giúp hệ thống không chỉ dừng lại ở vai trò kết nối mà còn trở thành một nền tảng hỗ trợ tuyển dụng toàn diện, nâng cao mức độ gắn kết của

người dùng.

Thứ hai, mô hình đề xuất cần tiếp tục được nâng cấp để tăng tính thuyết phục và cải thiện trải nghiệm. Hệ thống có thể phát triển theo hướng chuẩn hóa dữ liệu đầu vào tốt hơn (đặc biệt với kỹ năng, địa điểm và mức lương), tinh chỉnh cơ chế tính điểm theo phản hồi thực tế, và tăng khả năng giải thích kết quả đề xuất để người dùng hiểu rõ lý do phù hợp. Bên cạnh đó, việc khai thác hành vi người dùng như lượt xem, lượt lưu, lượt ứng tuyển hoặc phản hồi của nhà tuyển dụng có thể giúp hệ thống dần học được sở thích và xu hướng, từ đó nâng cao chất lượng gợi ý theo thời gian.

Hệ thống website tuyển dụng thông minh là kết quả của quá trình thực hiện nghiêm túc và cũng thể hiện định hướng nghiên cứu của sinh viên trong lĩnh vực đề xuất. Sinh viên kỳ vọng trong tương lai hệ thống sẽ tiếp tục được cải tiến về cả chức năng lẫn chất lượng đề xuất, qua đó đáp ứng tốt hơn nhu cầu của nhiều nhóm người dùng và phù hợp hơn với bối cảnh tuyển dụng thực tế.

TÀI LIỆU THAM KHẢO

TÀI LIỆU THAM KHẢO

- [1] *ITviec*. [Online]. Available: <https://itviec.com/>.
- [2] *TopCV*. [Online]. Available: <https://www.topcv.vn/viec-lam>.
- [3] *Design pattern là gì?* [Online]. Available: <https://topdev.vn/blog/design-pattern-la-gi/>.
- [4] *ReactJS là gì?* [Online]. Available: <https://2001ab.io/blog/reactjs-la-gi>.
- [5] *TypeScript là gì?* [Online]. Available: <https://topdev.vn/blog/typescript-la-gi/>.
- [6] *So sánh TypeScript với JavaScript*. [Online]. Available: <https://topdev.vn/blog/so-sanh-typescript-voi-javascript/>.
- [7] *Tailwind CSS*. [Online]. Available: <https://www.careerlink.vn/cam-nang-viec-lam/tu-van-nghe-nghiep/tailwind-css>.
- [8] *NodeJS*. [Online]. Available: <https://www.careerlink.vn/cam-nang-viec-lam/tu-van-nghe-nghiep/tailwind-css>.
- [9] *Tìm hiểu về Content-based Filtering – Phần 1*. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-content-based-filtering-phuong-phap-goi-y-dua-theo-noi-dung-phan-1-V3m5WGBg5O7>.
- [10] *Layered Architecture*. [Online]. Available: <https://lorifpeterson.com/?p=24>.