

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN HỮU TÂN ĐẠT - 51702075
NGÔ MINH HIẾU – 51702017

**XÂY DỰNG WEBSITE ĐĂNG TIN
TUYỂN DỤNG CỦA DOANH NGHIỆP CHO
SINH VIÊN TDTU SỬ DỤNG CÔNG NGHỆ
NODEJS VÀ REACT**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2
KỸ THUẬT PHẦN MỀM**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN HỮU TÂN ĐẠT - 51702075

NGÔ MINH HIẾU – 51702017

**XÂY DỰNG WEBSITE ĐĂNG TIN
TUYỂN DỤNG CỦA DOANH NGHIỆP CHO
SINH VIÊN TDTU SỬ DỤNG CÔNG NGHỆ
NODEJS VÀ REACT**

DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KỸ THUẬT PHẦN MỀM

Người hướng dẫn:

GV DOÃN XUÂN THANH

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy **Doãn Xuân Thành** – Giảng viên hướng dẫn của chúng em trong việc thực hiện dự án công nghệ thông tin 2, và khoa Công Nghệ Thông Tin đã cho em cơ hội thực hiện đề tài này và cũng như là cơ hội em được tiếp xúc với một số công nghệ được xem là trang bị cho em trong môi trường làm việc thực tế, cũng như trong việc hoàn thành đề tài xây dựng ứng dụng này.

Vì áp dụng 1 số kiến thức ở trường cũng như tìm hiểu và ứng dụng một số công nghệ mới nên chúng em sẽ không tránh khỏi được những sai sót và sự hạn chế của kiến thức tự tìm hiểu, đánh giá và trình bày về đề tài này. Rất mong nhận được sự quan tâm, góp ý của các thầy/cô để chung em biết mình thiếu sót những gì và giúp cho đề tài của chúng em được đầy đủ và hoành chính hơn. Chúng em xin chân thành cảm ơn.

TP. Hồ Chí Minh, ngày 03 tháng 12 năm 2022
Tác giả
(Ký tên và ghi rõ họ tên)

Nguyễn Hữu Tấn Đạt

Ngô Minh Hiếu

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH

TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Nhóm chúng em xin cam đoan đây là công trình nghiên cứu của riêng chúng em và được sự hướng dẫn khoa học của GV Doãn Xuân Thanh ;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào nhóm chúng em xin hoàn toàn chịu trách nhiệm về nội dung Báo cáo Dự án CNTT 2 của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 03 tháng 12 năm 2022

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Hữu Tân Đạt

Ngô Minh Hiếu

PHIẾU ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

Tên giảng viên hướng dẫn:

Ý kiến nhận xét:

.....

.....

Điểm tổng theo phiếu đánh giá rubrik:

TP. Hồ Chí Minh, ngày tháng năm 20
Giảng viên hướng dẫn
(Ký tên và ghi rõ họ tên)

XÂY DỰNG WEBSITE ĐĂNG TIN TUYỂN DỤNG CỦA DOANH NGHIỆP CHO SINH VIÊN TDTU SỬ DỤNG CÔNG NGHỆ NODEJS VÀ REACT

TÓM TẮT

- Giới thiệu các công nghệ và một số thuật ngữ được sử dụng trong quá trình xây dựng trang ứng dụng
 - o Docker, CI/CD, VPS, NGINX, NODEJS
- Xác định các tính năng cần có , xây dựng các bảng dữ liệu lưu trữ cần có để thực hiện hành vi API và giao diện tương ứng
- Quá trình xây dựng API và cách gọi đến API
- Quá trình xây dựng Giao diện.

MỤC LỤC

LỜI CẢM ƠN	i
TÓM TẮT	iii
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ.....	vi
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	x
CHƯƠNG 1 – GIỚI THIỆU	1
1.1. Giới thiệu về đề tài.....	1
1.1.1. Đặc tả	1
1.1.2. Usecase	13
1.1.3. Activity	27
1.2. Giới thiệu về công nghệ và thuật ngữ	45
1.2.1. Docker	45
1.2.2. CI/ CD	48
1.2.3. VPS	52
1.2.4. Nginx	54
1.2.5. NodeJS (ExpressJS)	56
1.2.6. ORM (PRISMA)	57
1.2.7. ReactJS	60
1.2.8. Redux	65
1.2.9. Typecript.....	68
1.2.10. Antd	71
1.3. Nội dung của chương này	72
CHƯƠNG 2 – QUÁ TRÌNH THIẾT LẬP CÁC CÔNG CỤ	73
2.1. Quá trình thiết lập server	73
2.2. Cách cài đặt gitlab CI/CD vào source code	80
CHƯƠNG 3 – KẾT QUẢ VÀ KẾT LUẬN	86
3.1. Kết quả	86
3.1.1. Giao diện	86

3.2. Kết luận.....	107
DANH MỤC TÀI LIỆU THAM KHẢO.....	109

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1.1 : Sơ đồ Class.....	12
Hình 1.2 : Sơ đồ usecase của trang ứng dụng	13
Hình 1.3 : Sơ đồ Activity – Đăng ký	27
Hình 1.4 : Sơ đồ Activity – Đăng Nhập.....	27
Hình 1.5 : Sơ đồ Activity – Tìm Kiếm	28
Hình 1.6 : Sơ đồ Activity – Đăng tải CV	28
Hình 1.7 : Sơ đồ Activity – Xóa CV	29
Hình 1.8 : Sơ đồ Activity – Đổi Tên CV	29
Hình 1.9 : Sơ đồ Activity – Nộp CV	30
Hình 1.10 : Sơ đồ Activity – Rút CV.....	30
Hình 1.11 : Sơ đồ Activity – Đánh Dấu Bài Đăng	31
Hình 1.12 : Sơ đồ Activity – Hủy Đánh Dấu Bài Đăng	31
Hình 1.13 : Sơ đồ Activity – Xem Các Bài Viết Đã Nộp CV	32
Hình 1.14 : Sơ đồ Activity – Xem Các Bài Viết Đã Đánh Dấu	32
Hình 1.15 : Sơ đồ Activity – Xóa bài đăng tuyển dụng.....	33
Hình 1.16 : Sơ đồ Activity – Sửa bài đăng tuyển dụng	33
Hình 1.17 : Sơ đồ Activity – Xem các đơn ứng tuyển của ứng viên	34
Hình 1.18 : Sơ đồ Activity – Cập nhật lại mật khẩu mới	35
Hình 1.19 : Sơ đồ Activity – Cập nhật lại thông tin của người dùng	35
Hình 1.20 : Sơ đồ Activity – Xem danh sách người dùng trong hệ thống	36
Hình 1.21 : Sơ đồ Activity – Xem danh sách bài đăng tuyển dụng của tất cả nhà tuyển dụng	36
Hình 1.22 : Sơ đồ Activity – Cập nhật thông tin nhóm tài khoản	37
Hình 1.23 : Sơ đồ Activity – Thêm mới loại nghề nghiệp.....	38
Hình 1.24 : Sơ đồ Activity – Cập nhật lại loại nghề nghiệp.....	39
Hình 1.25 : Sơ đồ Activity – Xóa loại nghề nghiệp.....	40

Hình 1.26 : Sơ đồ Activity – Thêm mới chuyên ngành	41
Hình 1.27 : Sơ đồ Activity – Cập nhật lại thông tin chuyên ngành	42
Hình 1.28 : Sơ đồ Activity – Xóa loại nghề nghiệp.....	43
Hình 1.29 : Sơ đồ Activity – Quên mật khẩu.....	44
Hình 1.30: Sơ đồ về kiến trúc của docker.....	46
Hình 1.31: Chi tiết và cách hoạt động và thao tác với docker	47
Hình 1.32: Quá trình hoạt động của CI/CD	48
Hình 1.33: Quá trình phát triển ứng dụng.....	49
Hình 1.34: Cách hoạt động của CI/CD	51
Hình 1.35: Tổng quan về quá trình hoạt động với CI/CD	51
Hình 1.36 : Nginx là gì.....	55
Hình 1.37: Mô hình hoạt động của Redux	65
Hình 1.38: Sơ đồ hoạt động của Redux.....	66
Hình 2.1: Hình ảnh thuê server	73
Hình 2.2: Gõ ssh-keygen để tạo ra bộ mã cho riêng mình.....	74
Hình 2.3: Thư mục .ssh chứa ssh-keygen	74
Hình 2.4: Hình ảnh tạo ssh key trong thư mục demo	75
Hình 2.5: Hình ảnh thư mục demo chứa ssh key ta vừa tạo với demo	75
Hình 2.6: Kết nối với server thuê được bằng ssh.....	76
Hình 2.7: Thêm ssh key vào máy chủ	76
Hình 2.8: Nội dung tập tin authorized_keys	77
Hình 2.9: Ta sẽ copy key trong file private key.....	78
Hình 2.10: Hình ảnh thể hiện máy chủ có các container đang hoạt động	80
Hình 2.11: Bước một tiến hành tạo dự án mới trên GitLab	80
Hình 2.12: Bước hai chọn chạy dự án mở rộng sử dụng CI/CD	80
Hình 2.13 : Bước ba chọn sử dụng dự án từ nền tảng GitHub	81
Hình 2.14: Màn hình yêu cầu ta nhập token ủy quyền	81
Hình 2.15: Bước bốn quay lại github để cài đặt	82

Hình 2.16: Bước năm chọn vào mục Developer settings	82
Hình 2.17: Tạo token ủy quyền.....	83
Hình 2.18: Bước tiếp theo điền các thông tin khi tạo token ủy quyền	83
Hình 2.19 : Sau khi khởi tạo token ta sẽ nhận được 1 dãy chữ cái ngẫu nhiên	84
Hình 2.20: Điền token ủy quyền cho gitlab	84
Hình 2.21: Màn hình sau khi ủy quyền.....	84
Hình 2.22: Dự án từ Github đã được đưa sang GitLab.....	85
Hình 3.1 Home Page - Banner	86
Hình 3.2 Home Page – Top Major.....	87
Hình 3.3: Home Page – Featured Job	88
Hình 3.4: Home Page – Recruiment	89
Hình 3.5: Home Page – How It Work.....	90
Hình 3.6: Footer	90
Hình 3.7: Header - UnAuth	91
Hình 3.8: Header - Candidate	91
Hình 3.9: Header – Recruiter	91
Hình 3.10: Header - Admin	91
Hình 3.11: Search page	92
Hình 3.12: Job Detail	94
Hình 3.13: Candidate – Profile	95
Hình 3.14: Candidate – CV Manager	96
Hình 3.15: Candidate – Bookmarks.....	97
Hình 3.16: Candidate – Applied Job.....	97
Hình 3.17: Candidate – Change paswords.....	98
Hình 3.18: Recruiter - Profile	98
Hình 3.19: Recruiter – Post Manager	99
Hình 3.20: Recruiter – New post step one	99
Hình 3.21: Recruiter – New post step two	100

Hình 3.22: Recruiter – New post step three	100
Hình 3.23: Recruiter – Edit Post	101
Hình 3.24: Recruiter – All Applicant.....	101
Hình 3.25: Recruiter – Change Password.....	102
Hình 3.26: Admin – Account Manager.....	102
Hình 3.27: Admin - Account Detail – Recruiter.....	103
Hình 3.28: Admin - Account Detail – Candidate	104
Hình 3.29: Admin – All Post	104
Hình 3.30: Admin – Account Type Manager	105
Hình 3.31: Admin - Job Type Manager	105
Hình 3.32 Link đến Website http://jobportaltdtu.com/	108

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC CHỮ VIẾT TẮT

CI/CD Chỉ quá trình tích hợp liên tục và phân phối liên tục

JS Là một ngôn ngữ lập trình JavaScript

TS Là một ngôn ngữ lập trình được phát triển bởi Microsoft và có cách viết tương tự như JavaScript nhưng chặt chẽ hơn và dễ phát hiện lỗi sai trong khi viết

CHƯƠNG 1 – GIỚI THIỆU

1.1.Giới thiệu về đề tài

Với nhu cầu tìm việc của nguồn lực trường, Trường đại học Tôn Đức Thắng muốn tăng cường cơ hội tìm việc cũng như có thêm một kênh thông tin về cơ hội việc làm cùng với các kênh khác như topcv, glint, hoặc facebook. Nhằm để đáp ứng với nhu cầu đó thì chúng em sẽ xem và xây dựng một kênh tìm việc cũng như đăng các tin tuyển dụng của các nhà tuyển dụng lên để tạo cơ hội việc làm cho sinh viên của trường

Kênh tìm kiếm việc này sẽ cho phép sinh viên tạo hồ sơ ứng tuyển của riêng mình cũng như dễ dàng tìm được công việc phù hợp dựa vào nguồn việc làm từ các đối tác lớn của nhà trường cũng như các công ty nhỏ và cần nguồn nhân lực chất lượng cao

1.1.1. Đặc tả

Để đáp ứng với nhu cầu xây dựng một trang ứng dụng cho phép sinh viên có thể tham gia ứng tuyển vào các công việc tìm năng do các đối tác của nhà trường đăng tin tuyển dụng , Sinh viên có thể tiếp xúc với nguồn việc làm ổn định và chất lượng cũng như cung cấp cho nhà tuyển dụng nguồn nhân lực tốt nhất. Trang ứng dụng này được xây dựng dựa trên dữ liệu trong các bảng như sau:

- Loại người dùng (User_Type)
 - id : để định danh - **Bigint autoincrement**
 - user_type_name : Tên của loại người dùng – **Nvarchar(1000)**
 - is_active : Loại người dùng có được kích hoạt hay không – **Bit (0 or 1)**
 - is_delete : Loại người dùng đã được xóa hay không – **Bit(0 or 1)**
 - create_date : Ngày khởi tạo loại người dùng – **Datetime**
 - create_user : Tên tài khoản tạo ra loại người dùng này - **Nvarchar(1000)**
 - update_date : Ngày cập nhật - **Datetime**
 - update_user : Người cập nhật – **Nvarchar(1000)**
 - delete_date : Ngày xóa - **Datetime**
 - delete_user : Người xóa – **Nvarchar(1000)**

Bảng Loại người dùng có chức năng chia và phân nhóm có các loại tài khoản được đăng kí ở trang web như là người quản trị (Admin) , Nhà tuyển dụng (Recruiter) và ứng viên (Candidate) giúp cho với mỗi loại người dùng sẽ có các đặc quyền riêng và giúp cho quá trình sử dụng trong trang web một cách thuận tiện nhất

- Tài khoản người dùng (User_Account)

- id : id định danh tài khoản – Bigint acutoincrement
- password : Mật khẩu đăng nhập – **Nvarchar(1000)**
- username : Tên đăng nhập – **Nvarchar(1000)**
- google_id : mã id của google - **Nvarchar(1000)**
- is_active : Tài khoản có được kích hoạt hay không - **Bit (0 or 1)**
- is_delete : Tài khoản có bị xóa hay không - **Bit (0 or 1)**
- create_date : Ngày tạo của tài khoản - **Datetime**
- user_type_id : id loại người dùng - **Datetime**
- first_name : Họ người dùng - **Nvarchar(1000)**
- last_name : Tên người dùng - **Nvarchar(1000)**
- full_name : Họ và tên đầy đủ - **Nvarchar(1000)**
- email : Email của người dùng và email sẽ đóng vai trò là nơi nhận thông báo từ trang ứng dụng - **Nvarchar(1000)**
- number_phone : Số điện thoại của người dùng - **Nvarchar(1000)**
- age : tuổi – **Int**
- gender : Giới tính - **Int**
- province_code : ID của tỉnh - **Nvarchar(1000)**
- district_code : ID của quận/huyện - **Nvarchar(1000)**
- ward_code : ID của phường/xã - **Nvarchar(1000)**
- address : địa chỉ cụ thể - **Nvarchar(1000)**
- avatar : Đường dẫn lưu trữ ảnh của người dùng - **Nvarchar(1000)**

Bảng tài khoản người dùng có chức năng lưu trữ các thông tin của người dùng giúp cho người dùng có thể đăng nhập vào trang web và giao tiếp với nhà tuyển dụng cũng như sử dụng các chức năng có của trang web

- Loại công việc (Job_Type)

- id : id định danh loại công việc - **Bigint autoincrement**
- job_type_name : Tên loại công việc - **Nvarchar(1000)**
- is_active : Loại công việc được kích hoạt hay không – **Bit (0 or 1)**
- is_delete : Loại công việc đã được xóa hay không – **Bit(0 or 1)**
- create_date : Ngày khởi tạo– **Datetime**
- create_user : Người khởi tạo - **Nvarchar(1000)**
- update_date : Ngày cập nhật - **Datetime**
- update_user : Người cập nhật – **Nvarchar(1000)**
- delete_date : Ngày xóa - **Datetime**
- delete_user : Người xóa – **Nvarchar(1000)**

Bảng loại công việc có chức năng lưu trữ các loại công việc được người quản trị khai báo, từ đó trên trang ứng dụng có thể thêm loại công việc mà người tuyển dụng muốn nhắm đến vào bài đăng tuyển dụng, giúp cho ứng viên có thể tìm kiếm các loại công việc mình muốn nộp đơn ứng tuyển như full time hay parttime

- Chuyên ngành (Majors)

- id : id định danh chuyên ngành - **Bigint autoincrement**
- job_type_name : Tên chuyên ngành- **Nvarchar(1000)**
- is_active : Chuyên ngành được kích hoạt hay không – **Bit (0 or 1)**
- is_delete : Chuyên ngành đã được xóa hay không – **Bit(0 or 1)**
- create_date : Ngày khởi tạo– **Datetime**
- create_user : Người khởi tạo - **Nvarchar(1000)**
- update_date : Ngày cập nhật - **Datetime**

- update_user : Người cập nhật – **Nvarchar(1000)**
- delete_date : Ngày xóa - **Datetime**
- delete_user : Người xóa – **Nvarchar(1000)**

Bảng chuyên ngành có nhiệm vụ lưu trữ các chuyên ngành có trên thị trường do người quản trị thêm vào giúp cho nhà tuyển dụng về chuyên ngành đó có thể thêm loại chuyên ngành vào bài đăng của mình cũng như giúp ứng viên học về chuyên ngành đó hoặc muốn có công việc về chuyên ngành đó có thể dễ dàng tìm kiếm được bài tuyển dụng một cách hợp lý

- Hồ sơ ứng tuyển (Apply_Profile)
 - profile_id : id định danh hồ sơ - **Bigint autoincrement**
 - profile_name : Tên hồ sơ ứng tuyển- **Nvarchar(1000)**
 - is_active : Hồ sơ ứng tuyển có được kích hoạt hay không – **Bit (0 or 1)**
 - is_delete : Hồ sơ ứng tuyển đã được xóa hay không – **Bit(0 or 1)**
 - create_date : Ngày khởi tạo– **Datetime**
 - create_user : Người khởi tạo - **Nvarchar(1000)**
 - update_date : Ngày cập nhật - **Datetime**
 - update_user : Người cập nhật – **Nvarchar(1000)**
 - delete_date : Ngày xóa - **Datetime**
 - delete_user : Người xóa – **Nvarchar(1000)**
 - user_id : Hồ sơ thuộc về tài khoản nào – **Bigint**

Bảng hồ sơ ứng tuyển được dùng vào mục đích là giúp cho ứng viên có thể tạo cho mình một hồ sơ có thể thu hút được nhà tuyển dụng hoặc được dùng để nộp vào các bài đăng phù hợp với mình

- Bài tuyển dụng (Recruitment_Post)
 - id : id định danh chuyên ngành - **Bigint autoincrement**
 - content : Nội dung bài viết - **Nvarchar(4000)**

- title : Tựa đề bài viết - **Nvarchar(1000)**
- recruiter_id : Id của nhà tuyển dụng - **Bigint**
- to_value : Mức lương khởi điểm - **Bigint**
- from_value : Mức lương tối đa - **Bigint**
- gender : Giới tính tuyển chọn - **Int**
- province_code : ID của tỉnh - **Nvarchar(1000)**
- district_code : ID của quận/huyện - **Nvarchar(1000)**
- ward_code : ID của phường/xã - **Nvarchar(1000)**
- address : địa chỉ cụ thể - **Nvarchar(1000)**
- is_active : Bài đăng có được kích hoạt hay không – **Bit (0 or 1)**
- is_delete : Bài đăng đã được xóa hay không – **Bit(0 or 1)**
- create_date : Ngày khởi tạo– **Datetime**
- create_user : Người khởi tạo - **Nvarchar(1000)**
- update_date : Ngày cập nhật - **Datetime**
- update_user : Người cập nhật – **Nvarchar(1000)**
- delete_date : Ngày xóa - **Datetime**
- delete_user : Người xóa – **Nvarchar(1000)**

Bảng bài tuyển dụng có công dụng là với mỗi nhà tuyển dụng có thể tạo ra bài đăng của mình và tìm được các ứng viên tiềm năng giúp tăng khả năng có được các ứng viên chất lượng đồng thời giúp cho ứng viên hiểu về công việc mà mình có thể nộp đơn vào với mức lương trong khoảng bao nhiêu , của nhà tuyển dụng nào , và cũng như là địa điểm làm việc ở đâu , giới tính mà bài đăng tập trung vào là gì?.

- Lịch sử ứng tuyển (History_Apply_Job)
 - id : Mã định danh lịch sử ứng tuyển - **Bigint autoincrement**
 - user_id : id của người dùng - **Bigint**
 - post_id : id bài tuyển dụng - **Bigint**

- cv_id : id của cv ứng tuyển - **Bigint**
- is_active : Lịch sử ứng tuyển được kích hoạt hay không – **Bit (0 or 1)**
- is_delete : Lịch sử ứng tuyển đã được xóa hay không – **Bit(0 or 1)**
- create_date : Ngày khởi tạo– **Datetime**
- create_user : Người khởi tạo - **Nvarchar(1000)**
- update_date : Ngày cập nhật - **Datetime**
- update_user : Người cập nhật – **Nvarchar(1000)**
- delete_date : Ngày xóa - **Datetime**
- delete_user : Người xóa – **Nvarchar(1000)**

Bảng lịch sử ứng tuyển giúp cho ứng viên có thêm xem lại những cơ hội mình đã nộp đơn

- Loại công việc của bài viết (Recruitment_Post_Job_Type)
 - id : Mã định danh loại công việc của bài viết - **Bigint autoincrement**
 - post_id : id của bài tuyển dụng - **Bigint**
 - job_type_id :id của loại công việc - **Bigint**
 - is_active : Loại công việc của bài viết được kích hoạt hay không – **Bit (0 or 1)**
 - is_delete : Loại công việc của bài viết đã được xóa hay không – **Bit(0 or 1)**
 - create_date : Ngày khởi tạo– **Datetime**
 - create_user : Người khởi tạo - **Nvarchar(1000)**
 - update_date : Ngày cập nhật - **Datetime**
 - update_user : Người cập nhật – **Nvarchar(1000)**
 - delete_date : Ngày xóa - **Datetime**
 - delete_user : Người xóa – **Nvarchar(1000)**

Bảng loại công việc của bài viết có nhiệm vụ lưu trữ tất cả các loại công việc được thêm vào bài viết vì 1 bài viết có thể có nhiều loại công việc nên nhiệm vụ của bảng này là bảng nối để lưu được mối quan hệ 1 – n của bài viết và loại công việc

- Loại chuyên ngành của bài viết (Recruitment_Post_Majors)
 - id : Mã định danh chuyên ngành của bài viết - **Bigint autoincrement**
 - post_id : id của bài viết
 - majors_id : id của chuyên ngành
 - is_active : Loại chuyên ngành của bài viết có kích hoạt hay không – **Bit (0 or 1)**
 - is_delete : Loại chuyên ngành của bài viết đã được xóa hay không – **Bit(0 or 1)**
 - create_date : Ngày khởi tạo– **Datetime**
 - create_user : Người khởi tạo - **Nvarchar(1000)**
 - update_date : Ngày cập nhật - **Datetime**
 - update_user : Người cập nhật – **Nvarchar(1000)**
 - delete_date : Ngày xóa - **Datetime**
 - delete_user : Người xóa – **Nvarchar(1000)**

Bảng loại chuyên ngành của bài viết có nhiệm vụ lưu trữ tất cả các loại chuyên ngành được thêm vào bài viết vì 1 bài viết có thể thuộc về nhiều chuyên ngành nên nhiệm vụ của bảng này là bảng nối để lưu được mối quan hệ 1 – n của bài viết và loại chuyên ngành

- Đơn ứng tuyển (cv)
 - id : Mã định danh đơn - **Bigint autoincrement**
 - file_name_hash : Tên ngẫu nhiên của cv trên server - **Nvarchar(1000)**
 - file_name : tên file gốc - **Nvarchar(1000)**
 - user_id : id user của cv - **Bigint**
 - extname : Tên mở rộng của file - **Nvarchar(1000)**
 - is_active : Đơn ứng tuyển có kích hoạt hay không – **Bit (0 or 1)**
 - is_delete : Đơn ứng tuyển đã được xóa hay không – **Bit(0 or 1)**

- `create_date` : Ngày khởi tạo– **Datetime**
- `create_user` : Người khởi tạo - **Nvarchar(1000)**

Bảng cv có nhiệm vụ với mỗi ứng viên cần có nhiều cv và với mỗi cv trong đó sẽ tóm tắt về bản thân , trình độ học vấn , sở thích , cũng như là quá trình làm việc của bản thân , bảng này sẽ lưu đường dẫn tập tin Người dùng sẽ tải lên cv của cá nhân và bảng sẽ lưu đường dẫn cũng như tên của cv ở server người dùng có thể dùng nó để ứng tuyển , hoặc là tải xuống , xem lại

- Khu vực hành chính (administrative_region)
 - `id` : id định danh khu vực hành chính - **Int**
 - `name` : Tên khu vực hành chính - **Nvarchar(1000)**
 - `name_en` : Tên khu vực hành chính tiếng anh - **Nvarchar(1000)**
 - `code_name` : Mã khu vực hành chính - **Nvarchar(1000)**
 - `code_name_en` : Mã khu vực hành chính tiếng anh - **Nvarchar(1000)**

Bảng khu vực hành chính lưu các khu vực của nước Việt Nam như là Đông Nam Bộ , Đồng Bằng Sông Cửu Long

- Đơn vị hành chính (administrative_units)
 - `id` : id định danh khu vực hành chính - **Int**
 - `full_name` : Tên đầy đủ đơn vị hành chính - **Nvarchar(1000)**
 - `full_name_en` : Tên đầy đủ đơn vị hành chính với tiếng anh - **Nvarchar(1000)**
 - `short_name` : Tên gọi ngắn của đơn vị hành chính - **Nvarchar(1000)**
 - `short_name_en` : Tên gọi ngắn tiếng anh đơn vị hành chính- **Nvarchar(1000)**
 - `code_name` : Mã tên gọi của đơn vị hành chính- **Nvarchar(1000)**
 - `code_name_en` : Mã tên gọi tiếng anh của đơn vị hành chính- **Nvarchar(1000)**

Bảng Đơn vị hành chính sẽ lưu tên các đơn vị hành chính của nước Việt Nam như là Tỉnh , Thành Phố trực thuộc trung ương,

- Tỉnh thành phố (provinces)
 - id : Mã định danh tỉnh thành phố - **Int**
 - code : Mã của tỉnh thành phố - **Nvarchar(1000)**
 - name : Tên của tỉnh thành phố- **Nvarchar(1000)**
 - name_en: Tên tiếng anh của tỉnh thành phố- **Nvarchar(1000)**
 - full_name : Tên đầy đủ của tỉnh thành phố - **Nvarchar(1000)**
 - full_name_en : Tên tiếng anh đầy đủ của tỉnh thành phố- **Nvarchar(1000)**
 - code_name : Tên mã của tỉnh thành phố- **Nvarchar(1000)**
 - administrative_unit_id : Mã đơn vị hành chính - **Int**
 - adminstrartive_region_id : Mã khu vực hành chính – **Int**

Bảng tỉnh thành phố sẽ lưu lại tên , mã tỉnh phố , của 63 tỉnh thành của Việt Nam

- Quận/Huyện (districts)
 - id : Mã định danh quận huyện - **Int**
 - code : Mã của quận huyện- **Nvarchar(1000)**
 - name : Tên của tỉnh quận huyện- **Nvarchar(1000)**
 - name_en: Tên tiếng anh của quận huyện - **Nvarchar(1000)**
 - full_name : Tên đầy đủ của quận huyện - **Nvarchar(1000)**
 - full_name_en : Tên tiếng anh đầy đủ của quận huyện- **Nvarchar(1000)**
 - code_name : Tên mã của quận huyện - **Nvarchar(1000)**
 - administrative_unit_id : Mã đơn vị hành chính - **Int**
 - adminstrartive_region_id : Mã khu vực hành chính – **Int**
 - province_code : Mã tỉnh quận huyện thuộc - **Nvarchar(1000)**

Bảng Quận/Huyện lưu trữ thông tin của từng quận/Huyện trong 63 tỉnh thành của Việt Nam và đơn vị hành chính

- Phường/Xã (Wards)

- id : Mã định danh phường xã - **Int**
- code : Mã của phường xã- **Nvarchar(1000)**
- name : Tên của phường xã- **Nvarchar(1000)**
- name_en: Tên tiếng anh của phường xã- **Nvarchar(1000)**
- full_name : Tên đầy đủ của phường xã - **Nvarchar(1000)**
- full_name_en : Tên tiếng anh đầy đủ của phường xã- **Nvarchar(1000)**
- code_name : Tên mã của phường xã - **Nvarchar(1000)**
- district_code : Mã Quận/Huyện thuộc về - **Nvarchar(1000)**
- administrative_unit_id : Mã đơn vị hành chính – **Int**

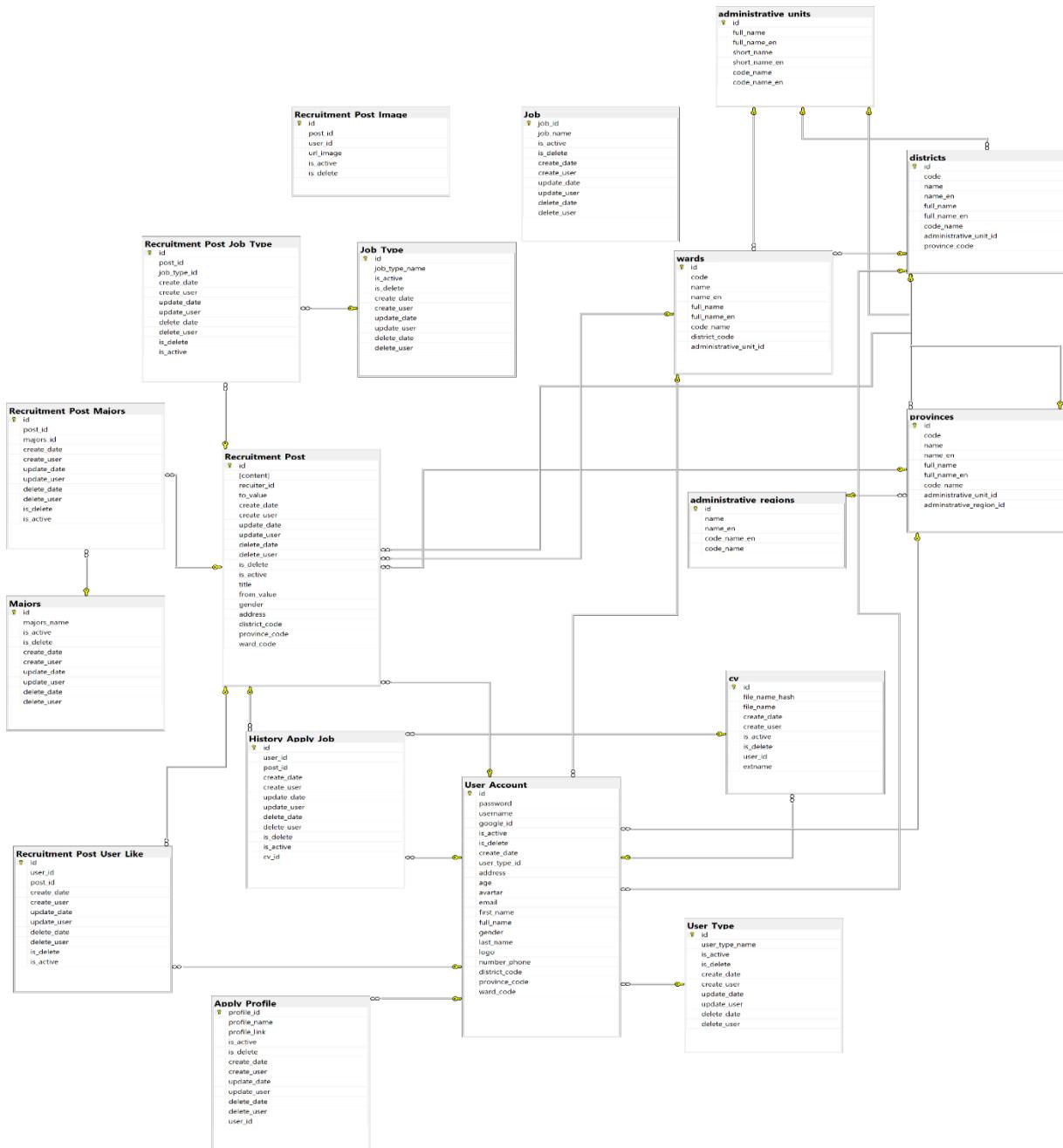
Bảng Phường/Xã sẽ lưu thông tin của phường xã của nhiều Quận/Huyện của tỉnh và đơn vị hành chính

- Bài tuyển dụng đánh dấu của người dùng (Recruitment_Post_User_Like)
 - id : Mã định danh lượt đánh dấu bài tuyển dụng - **Bigint autoincrement**
 - user_id : Mã định danh của người đánh dấu- **Bigint**
 - post_id : Mã định danh của bài viết- **Bigint**
 - is_active : Lượt đánh dấu có kích hoạt hay không – **Bit (0 or 1)**
 - is_delete : Lượt đánh dấu đã được xóa hay không – **Bit(0 or 1)**
 - create_date : Ngày khởi tạo– **Datetime**
 - create_user : Người khởi tạo - **Nvarchar(1000)**
 - update_date : Ngày cập nhật - **Datetime**
 - update_user : Người cập nhật – **Nvarchar(1000)**
 - delete_date : Ngày xóa - **Datetime**
 - delete_user : Người xóa – **Nvarchar(1000)**

Bảng lưu bài viết được đánh dấu bởi người dùng có chức năng lưu trữ thông tin bài đăng người ứng viên cảm thấy phù hợp nhưng chưa muốn ứng tuyển và sơ việc không tìm thấy được bài đăng thì người ứng viên có thể đánh dấu và ứng tuyển sau

Người dùng sẽ được chia làm 3 nhóm tài khoản và thao tác với một số chức năng nhất định là

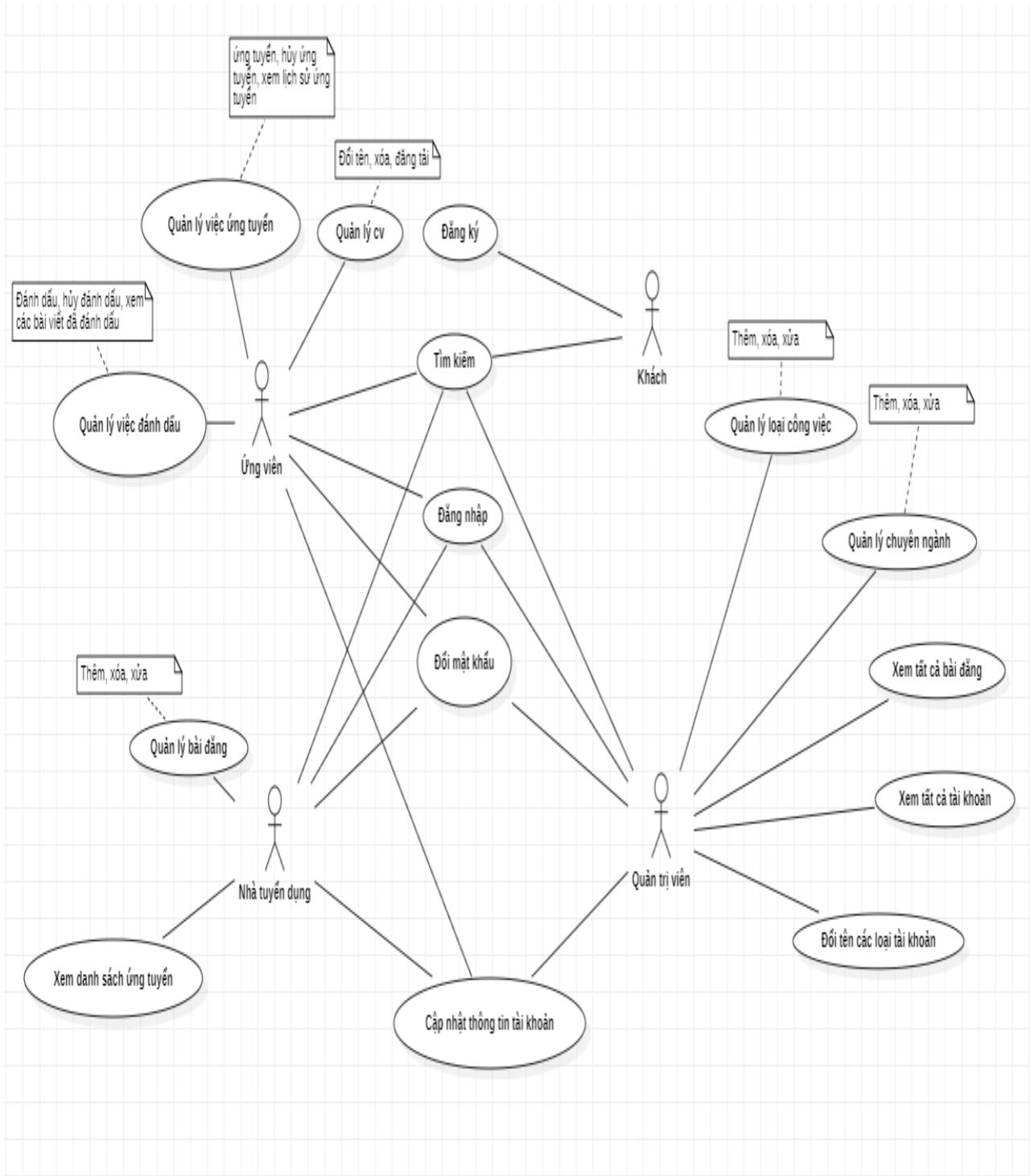
- **Ứng viên :** Có thể là sinh viên của trường hoặc là các ứng viên có nhu cầu tham gia vào trang web thông qua việc đăng ký
- **Nhà tuyển dụng :** Là các đối tác của quý nhà trường về các vị trí việc làm chất lượng và có nhu cầu tìm đến các nguồn nhân lực trẻ mới ra trường chưa có kinh nghiệm để tuyển làm thực tập sinh, hoặc là Fresher để đào tạo cho phù hợp với nhu cầu của danh nghiệp
- **Người quản lý:** Có vai trò là người quan sát , giám sát các hoạt động của trang web cũng như khai báo và quản lý các tài nguyên của trang ứng dụng để giúp cho cả 2 loại người dùng là ứng viên và nhà tuyển dụng có thể tiếp xúc với nhau một cách thuận tiện nhất.



Hình 1.1: Sơ đồ Class

1.1.2. Usecase

- Sơ đồ usecase



Hình 1.2 : Sơ đồ usecase của trang ứng dụng

- Đặt tả usecase
- Đăng nhập

Use case name	Đăng Nhập					
Scenario	Cần đăng nhập khi muốn thực hiện các tính đăng cù thể của từng loại người dùng					
Triggering Event	Khi người dùng có nhu cầu thao tác với các tác vụ bên trong thì cần thực hiện đăng nhập					
Brief Description	Người dùng có thể chọn trang đăng nhập để đăng nhập					
Actor	Admin, Candidate, Recruiter					
Precondition	Người dùng phải nhập đúng tài khoản và mật khẩu mới có thể vào được chương trình					
Flow of Event	<table border="1"> <tr> <th>Actor</th> <th>System</th> </tr> <tr> <td> 1. Người dùng vào trang đăng nhập 2. Thực hiện thao tác đăng nhập vào hệ thống thông qua tên tài khoản và mật khẩu đã đăng ký </td> <td> 2.1 Hệ Thống sẽ thực hiện việc kiểm tra tên tài khoản và mật khẩu mà nhân viên vừa nhập vào có trong kho dữ liệu hay không </td> </tr> </table>	Actor	System	1. Người dùng vào trang đăng nhập 2. Thực hiện thao tác đăng nhập vào hệ thống thông qua tên tài khoản và mật khẩu đã đăng ký	2.1 Hệ Thống sẽ thực hiện việc kiểm tra tên tài khoản và mật khẩu mà nhân viên vừa nhập vào có trong kho dữ liệu hay không	
Actor	System					
1. Người dùng vào trang đăng nhập 2. Thực hiện thao tác đăng nhập vào hệ thống thông qua tên tài khoản và mật khẩu đã đăng ký	2.1 Hệ Thống sẽ thực hiện việc kiểm tra tên tài khoản và mật khẩu mà nhân viên vừa nhập vào có trong kho dữ liệu hay không					
Exception Condition	2.2 Nếu người dùng thực hiện việc đăng nhập không thành công thì sẽ hiện thông báo là tài khoản mật khẩu không hợp lệ					

- Search

Use case name	Search					
Scenario	Tìm kiếm các bài đăng theo nhiều tiêu chí					
Triggering Event	Khi người dùng có nhu cầu tìm kiếm các bài đăng					
Brief Description	Người dùng truy cập trang search					
Actor	Người dùng thường, admin, candidate, recruiter					
Precondition						
Flow of Event	<table border="1"> <tr> <th>Actor</th> <th>System</th> </tr> <tr> <td> 1. Người dùng vào trang search 2. Tùy chỉnh bộ lọc theo tiêu chí </td> <td> 2.1 Hệ Thống sẽ tìm kiếm và trả về các bài đăng đáp ứng đủ điều kiện mà người dùng đã lọc </td> </tr> </table>	Actor	System	1. Người dùng vào trang search 2. Tùy chỉnh bộ lọc theo tiêu chí	2.1 Hệ Thống sẽ tìm kiếm và trả về các bài đăng đáp ứng đủ điều kiện mà người dùng đã lọc	
Actor	System					
1. Người dùng vào trang search 2. Tùy chỉnh bộ lọc theo tiêu chí	2.1 Hệ Thống sẽ tìm kiếm và trả về các bài đăng đáp ứng đủ điều kiện mà người dùng đã lọc					
Exception Condition						

- Đăng ký

Use case name	Đăng Ký	
Scenario	Khi người dùng muốn đăng ký tài khoản	
Triggering Event	Khi người dùng có nhu cầu thao tác với các tác vụ bên trong thì cần thực hiện đăng ký tài khoản nếu chưa có tài khoản	
Brief Description	Người dùng có thể chọn trang đăng ký để đăng ký	
Actor	Người dùng thường	
Precondition	Người dùng phải nhập các thông tin cần thiết	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng vào trang đăng ký 2. Đăng ký tên và mật khẩu 3. Điền các thông tin cần thiết 	<p>2.1 Hệ Thống sẽ thực hiện việc kiểm tra tên tài khoản và email xem đã có trong hệ thống chưa, nếu chưa sẽ thêm tài khoản vào hệ thống</p>
Exception Condition	2.2 Nếu người dùng đăng ký với tên đăng nhập và email đã có trong hệ thống sẽ báo với người dùng email hoặc tên tài khoản đã tồn tại	

- Đăng tải CV

Use case name	Upload cv	
Scenario	Candidate upload cv	
Triggering Event	Khi candidate có nhu cầu upload cv	
Brief Description	Người dùng truy cập trang trang candidate và upload cv	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang candidate 3. Vào tab cv manager 4. Upload cv 	<p>2.1 Hệ thống sẽ lưu dữ liệu vào cơ sở dữ liệu</p>
Exception Condition		

- Đổi tên CV

Use case name	Đổi tên cv	
Scenario	Candidate đổi tên cv	
Triggering Event	Khi candidate có nhu cầu đổi tên cv	
Brief Description	Người dùng truy cập trang candidate và đổi tên cv	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang candidate 3. Vào tab cv manager 4. Đổi tên cv 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ lưu dữ liệu mới vào cơ sở dữ liệu
Exception Condition		

- Xóa CV

Use case name	Xóa cv	
Scenario	Candidate xóa cv	
Triggering Event	Khi candidate có nhu cầu xóa cv	
Brief Description	Người dùng truy cập trang candidate và xóa cv	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang candidate 3. Vào tab cv manager 4. Xóa cv 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ xóa cv đó khỏi cơ sở dữ liệu
Exception Condition		

- Nộp CV

Use case name	Nộp cv	
Scenario	Candidate nộp cv	
Triggering Event	Khi candidate có nhu cầu nộp cv vào một bài đăng	
Brief Description	Người dùng truy cập trang search và nộp cv vào 1 bài đăng	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang search 3. Tìm 1 bài đăng 4. Nộp cv 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ ghi nhận cv nộp vào bài đăng và gửi mail thông báo cho nhà tuyển dụng của bài đăng đó
Exception Condition		

- Rút CV

Use case name	Hủy nộp cv	
Scenario	Candidate hủy nộp cv	
Triggering Event	Khi candidate có nhu cầu hủy nộp cv vào một bài đăng	
Brief Description	Người dùng truy cập trang candidate và hủy nộp cv vào 1 bài đăng	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang candidate 3. Vào tab applied post 4. Hủy nộp cv 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ xóa ghi nhận về cv đã nộp vào vài viết
Exception Condition		

- Xem lịch sử nộp CV

Use case name	Xem lịch sử nộp cv	
Scenario	Candidate xem lịch sử cv	
Triggering Event	Khi candidate có nhu cầu xem lịch sử nộp cv	
Brief Description	Người dùng truy cập trang candidate và xem lịch sử nộp cv	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang candidate 3. Vào tab applied post 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ trả về lịch sử nộp cv của user đó
Exception Condition		

- Đánh dấu

Use case name	Đánh dấu bài đăng	
Scenario	Candidate đánh dấu bài đăng	
Triggering Event	Khi candidate có nhu cầu đánh dấu bài đăng	
Brief Description	Người dùng truy cập trang search và đánh dấu bài đăng	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang search 3. Tìm bài đăng 4. Đánh dấu bài đăng 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ ghi nhận và lưu thông tin vào cơ sở dữ liệu
Exception Condition		

- Hủy đánh dấu

Use case name	Hủy đánh dấu bài đăng	
Scenario	Candidate hủy đánh dấu bài đăng	
Triggering Event	Khi candidate có nhu cầu hủy đánh dấu bài đăng	
Brief Description	Người dùng truy cập trang candidate và hủy đánh dấu bài đăng	
Actor	Candidate	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate 2. Người dùng vào trang candidate 3. Vào tab bookmark manager 4. Hủy đánh dấu bài đăng 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ ghi nhận và xóa thông tin vào cơ sở dữ liệu
Exception Condition		

- Cập nhật thông tin tài khoản

Use case name	Cập nhật thông tin	
Scenario	Người dùng cập nhật thông tin	
Triggering Event	Khi người dùng có nhu cầu cập nhật thông tin	
Brief Description	Người dùng truy cập trang candidate hoặc recruiter và cập nhật thông tin	
Actor	Candidate, Recruiter	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate hoặc Recruiter	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate hoặc recruiter 2. Người dùng vào trang candidate hoặc recruiter 3. Vào tab profile 4. Cập nhật thông tin 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ ghi nhận và lưu thông tin mới vào cơ sở dữ liệu

- Đổi mật khẩu

Use case name	Thay đổi mật khẩu	
Scenario	Người dùng Thay đổi mật khẩu	
Triggering Event	Khi người dùng có nhu cầu thay đổi mật khẩu	
Brief Description	Người dùng truy cập trang candidate hoặc recruiter và thay đổi mật khẩu	
Actor	Candidate, Recruiter	
Precondition	Người dùng phải đăng nhập với loại tài khoản Candidate hoặc Recruiter	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống với tài khoản candidate hoặc recruiter 2. Người dùng vào trang candidate hoặc recruiter 3. Vào tab change password 4. Thay đổi mật khẩu 	<p>2.1 Hệ thống sẽ ghi nhận và lưu thông tin mới vào cơ sở dữ liệu</p>

- Cập nhật loại tài khoản

Use case name	Cập nhật lại thông tin của các loại tài khoản	
Scenario	Để người quản trị có thể thay đổi thông tin của loại tài khoản đăng ký	
Triggering Event	Khi người quản trị có thể thay đổi thông tin của loại tài khoản	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) sẽ có thể cập nhật lại tên cho loại tài khoản	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người quản trị đăng nhập vào hệ thống 2. Chọn loại tài khoản cần cập nhật 3. Cập nhật thông tin 	<p>2.1 Hệ thống sẽ lưu lại thay đổi của loại tài khoản</p>

- Xem danh sách người dùng

Use case name	Xem tất cả danh sách người dùng	
Scenario	Để người quản lý có thể quản lý người dùng	
Triggering Event	Khi người quản trị có thể xem hệ thống có những người dùng nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) sẽ có thể xem toàn bộ người dùng của hệ thống	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người quản trị đăng nhập vào hệ thống 2. Chọn vào danh sách người dùng 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ lấy ra tất cả danh sách người dùng hợp lệ có trong hệ thống

- Xem tất cả bài đăng

Use case name	Xem tất cả bài đăng tuyển dụng trong hệ thống	
Scenario	Để người quản trị có thể quản lý bài đăng tuyển dụng	
Triggering Event	Khi người quản trị có thể xem hệ thống có tất cả bài đăng của những nhà tuyển dụng nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) sẽ có thể xem toàn bộ danh sách bài đăng của tất cả nhà tuyển dụng	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người quản trị đăng nhập vào hệ thống 2. Chọn vào danh sách bài đăng 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ lấy ra tất cả danh sách bài đăng tuyển dụng hợp lệ có trong hệ thống

- Thêm loại nghề nghiệp

Use case name	Thêm mới loại nghề nghiệp	
Scenario	Để người quản trị có thể thêm mới bất kì loại nghề nghiệp mà người quản trị muốn	
Triggering Event	Khi người quản trị có thể thực hiện thao tác thêm mới bất kì loại nghề nghiệp nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) , chọn thêm mới loại nghề nghiệp , và điền các thông tin cần thiết	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người quản trị đăng nhập vào hệ thống 2. Chọn loại thêm mới loại nghề nghiệp 3. Điền đầy đủ các thông tin của loại nghề nghiệp 4. Lưu 	<p>2.1 Hệ thống sẽ lưu lại giá trị của loại nghề nghiệp đó và trên trang web sẽ hiện thị loại nghề nghiệp đó</p>

- Xem danh sách ứng tuyển

Use case name	Xem ứng viên đã ứng tuyển	
Scenario	Để nhà tuyển dụng có thể xem ứng viên đã ứng tuyển và bài đăng của nhà tuyển dụng	
Triggering Event	Khi nhà tuyển dụng vào trang xem các ứng viên	
Brief Description	Nhà tuyển dụng đăng nhập vào giao diện Recruiter, và xem các ứng viên	
Actor	Recruiter	
Precondition	Phải là người dùng có quyền recruiter	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Nhà tuyển dụng đăng nhập vào hệ thống 2. Vào trang xem các ứng viên 	<p>2.1 Hệ thống hiện các ứng viên đã nộp cv vào các post của người dùng</p>
Exception Condition		

- Cập nhật loại nghề nghiệp

Use case name	Cập nhật loại nghề nghiệp	
Scenario	Để người quản trị có thể cập nhật lại thông tin của bất kì loại nghề nghiệp mà người quản trị muốn	
Triggering Event	Khi người quản trị có thể thực hiện thao tác cập nhật thông tin của bất kì loại nghề nghiệp nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) , chọn loại nghề nghiệp, tiến hành cập nhật và điền các thông tin cần cập nhật	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	<p>Actor</p> <ol style="list-style-type: none"> 1. Người quản trị đăng nhập vào hệ thống 2. Chọn loại nghề nghiệp cần cập nhật 3. Chính sửa thông tin loại nghề nghiệp theo ý muốn 4. Lưu 	<p>System</p> <p>2.1 Hệ thống sẽ lưu lại giá trị đã thay đổi của loại nghề nghiệp đó và trên trang web sẽ hiện thị loại nghề nghiệp đó</p>
Exception Condition		

- Xóa loại nghề nghiệp

Use case name	Xóa loại nghề nghiệp	
Scenario	Người quản trị có thể xóa bất kì loại nghề nghiệp mà người quản trị muốn	
Triggering Event	Khi người quản trị có thể thực hiện thao tác xóa bất kì loại nghề nghiệp nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) , chọn loại nghề nghiệp, tiến hành cập nhật và điền các thông tin cần cập nhật	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	<p>Actor</p> <ol style="list-style-type: none"> 1. Người quản trị đăng nhập vào hệ thống 2. Chọn loại nghề nghiệp cần cập nhật 3. Xóa loại nghề nghiệp theo ý muốn 	<p>System</p> <p>2.1 Hệ thống sẽ xóa loại nghề nghiệp đó</p>

- Thêm chuyên ngành

Use case name	Thêm mới chuyên ngành	
Scenario	Để người quản trị có thể thêm mới bất kì chuyên ngành mà người quản trị muốn	
Triggering Event	Khi người quản trị có thể thực hiện thao tác thêm mới bất kì chuyên ngành nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) , chọn thêm mới chuyên ngành, và điền các thông tin cần thiết	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	Actor 1. Người quản trị đăng nhập vào hệ thống 2. Chọn loại thêm mới chuyên ngành 3. Điền đầy đủ các thông tin của chuyên ngành 4. Lưu	System 2.1 Hệ thống sẽ lưu lại giá trị của chuyên ngành đó

- Cập nhật loại nghề nghiệp

Use case name	Cập nhật chuyên ngành	
Scenario	Để người quản trị có thể cập nhật lại thông tin của bất kì chuyên ngành mà người quản trị muốn	
Triggering Event	Khi người quản trị có thể thực hiện thao tác cập nhật thông tin của bất kì chuyên ngành nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) , chọn chuyên ngành, tiến hành cập nhật và điền các thông tin cần cập nhật	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	Actor 1. Người quản trị đăng nhập vào hệ thống 2. Chọn chuyên ngành cần cập nhật 3. Chính sửa thông tin chuyên ngành theo ý muốn 4. Lưu	System 2.1 Hệ thống sẽ lưu lại giá trị đã thay đổi của chuyên ngành

- Xóa loại nghề nghiệp

Use case name	Xóa chuyên ngành	
Scenario	Người quản trị có thể xóa bất kỳ chuyên ngành mà người quản trị muốn	
Triggering Event	Khi người quản trị có thể thực hiện thao tác xóa bất kỳ chuyên ngành nào	
Brief Description	Người quản trị đăng nhập vào giao diện admin (Quản trị) , chọn chuyên ngành, tiến hành cập nhật và điền các thông tin cần cập nhật	
Actor	Admin	
Precondition	Phải là người dùng có quyền admin	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Người quản trị đăng nhập vào hệ thống 2. Chọn chuyên ngành cần cập nhật 3. Xóa chuyên ngành theo ý muốn 	2.1 Hệ thống sẽ xóa chuyên ngành đó
Exception Condition		

- Thêm bài đăng

Use case name	Thêm mới bài đăng	
Scenario	Để nhà tuyển dụng có thể thêm mới bài đăng mà nhà tuyển dụng muốn	
Triggering Event	Khi nhà tuyển dụng thực hiện thao tác thêm mới bài đăng	
Brief Description	Nhà tuyển dụng đăng nhập vào giao diện Recruiter, chọn thêm bài đăng mới, và điền các thông tin cần thiết	
Actor	Recruiter	
Precondition	Phải là người dùng có quyền recruiter	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Nhà tuyển dụng đăng nhập vào hệ thống 2. Chọn thêm bài đăng mới 3. Điền đầy đủ các thông tin của bài đăng 4. Đăng bài 	2.1 Hệ thống sẽ lưu bài đăng đó và hiện trên trang tìm kiếm
Exception Condition		

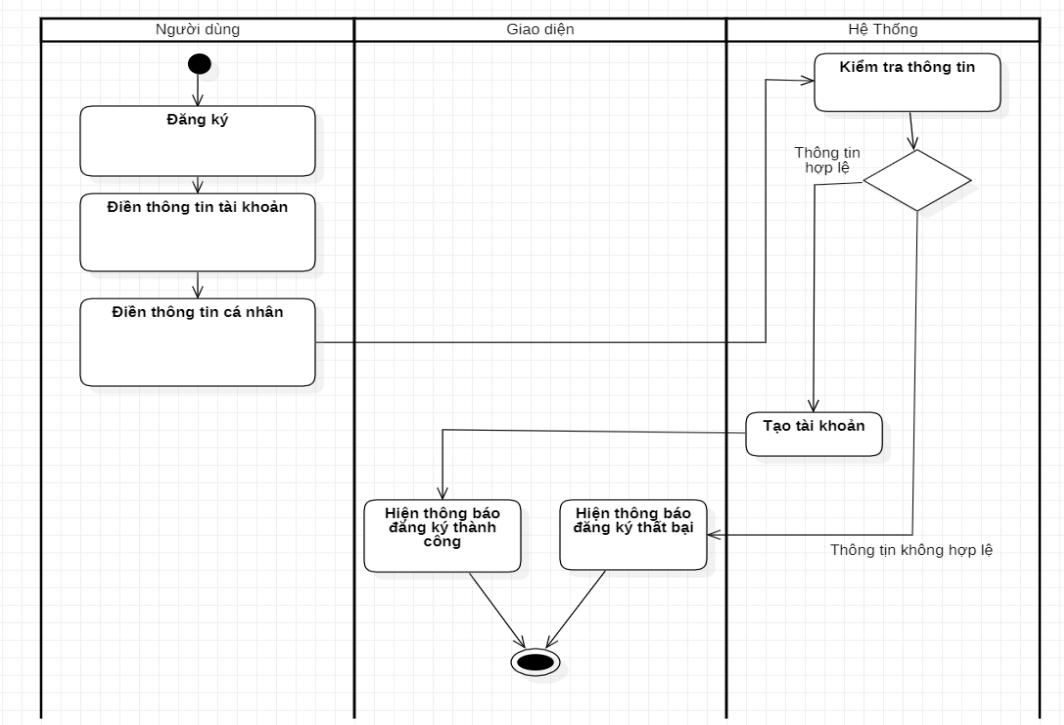
- Cập nhật bài đăng

Use case name	Cập nhật bài đăng	
Scenario	Để nhà tuyển dụng có thể cập nhật bài đăng mà nhà tuyển dụng muốn	
Triggering Event	Khi nhà tuyển dụng thực hiện thao tác cập nhật bài đăng	
Brief Description	Nhà tuyển dụng đăng nhập vào giao diện Recruiter, chọn cập nhật bài đăng, và cập nhật các thông tin cần thiết	
Actor	Recruiter	
Precondition	Phải là người dùng có quyền recruiter	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Nhà tuyển dụng đăng nhập vào hệ thống 2. Chọn bài đăng muốn cập nhật 3. Cập nhật thông tin của bài đăng 4. Lưu 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ lưu lại thay đổi của bài đăng đó và hiện trên trang tìm kiếm
Exception Condition		

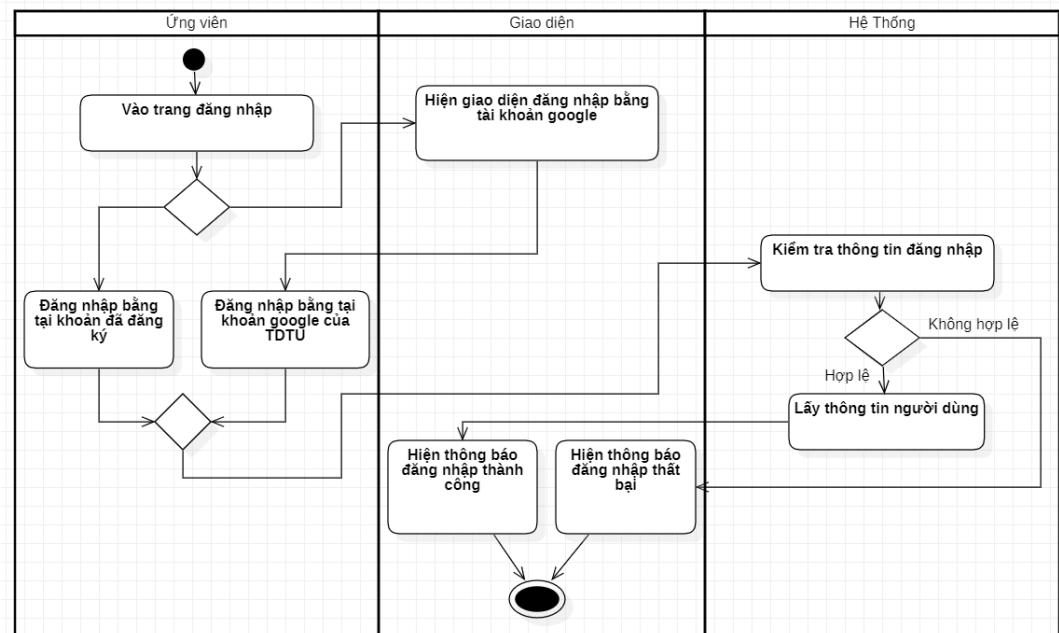
- Xóa bài đăng

Use case name	Xóa bài đăng	
Scenario	Để nhà tuyển dụng có thể xóa bài đăng mà nhà tuyển dụng muốn	
Triggering Event	Khi nhà tuyển dụng thực hiện thao tác xóa bài đăng	
Brief Description	Nhà tuyển dụng đăng nhập vào giao diện Recruiter, và xóa bài đăng.	
Actor	Recruiter	
Precondition	Phải là người dùng có quyền recruiter	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. Nhà tuyển dụng đăng nhập vào hệ thống 2. Chọn bài đăng muốn xóa 3. Xóa bài đăng 	<ol style="list-style-type: none"> 2.1 Hệ thống sẽ xóa bài đăng đó và ẩn trên trang tìm kiếm
Exception Condition		

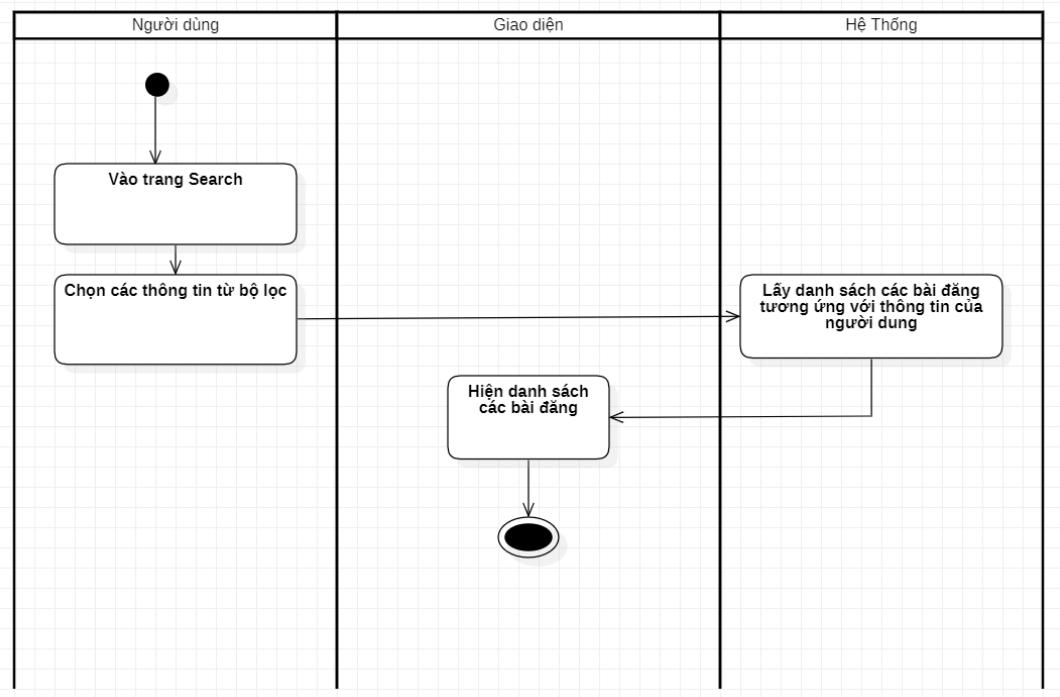
1.1.3. Activity



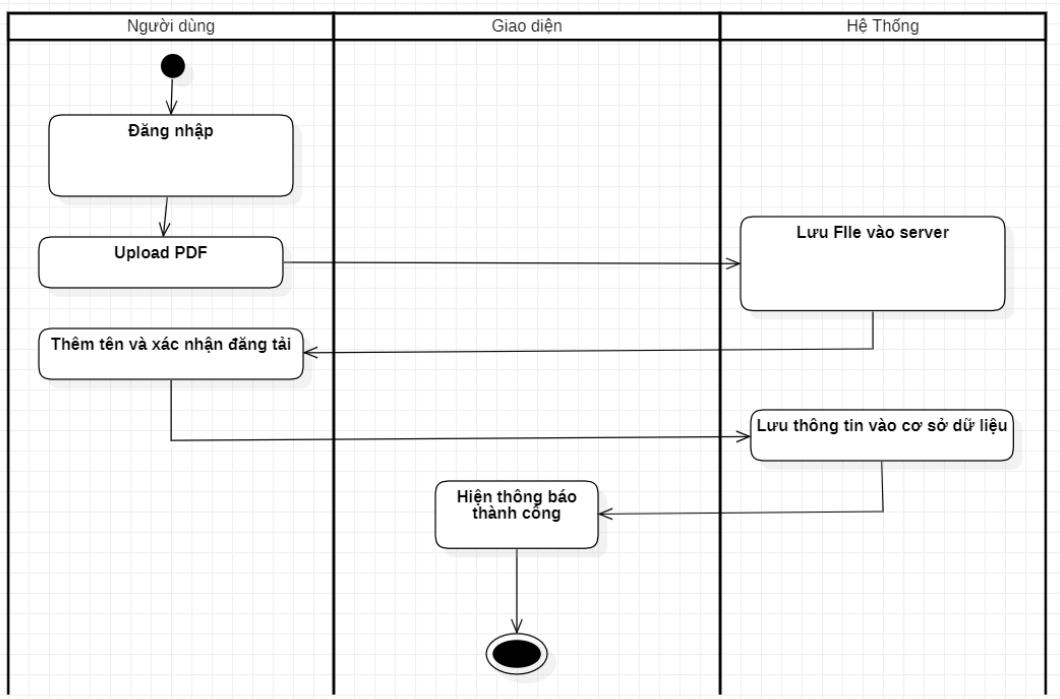
Hình 1.3 : Sơ đồ Activity – Đăng ký



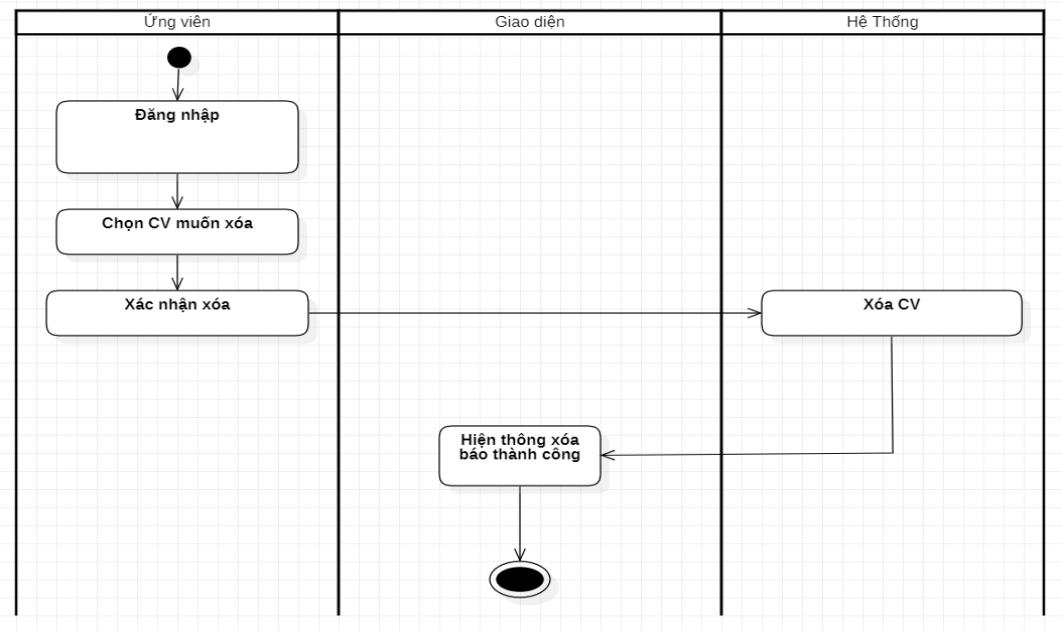
Hình 1.4 : Sơ đồ Activity – Đăng Nhập



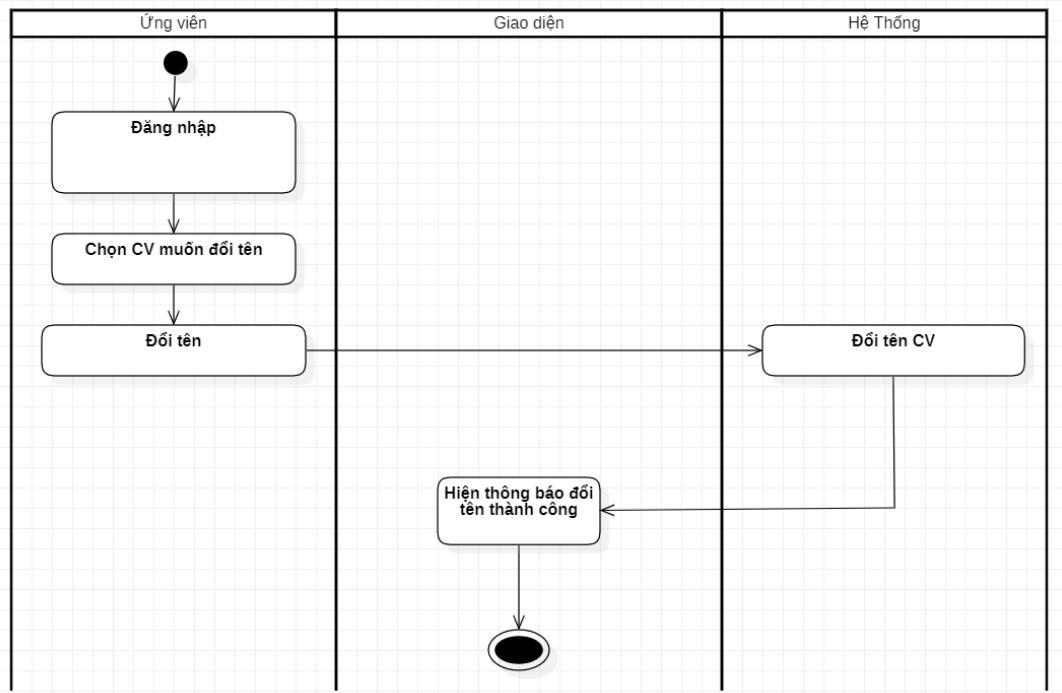
Hình 1.5 : Sơ đồ Activity – Tìm Kiếm



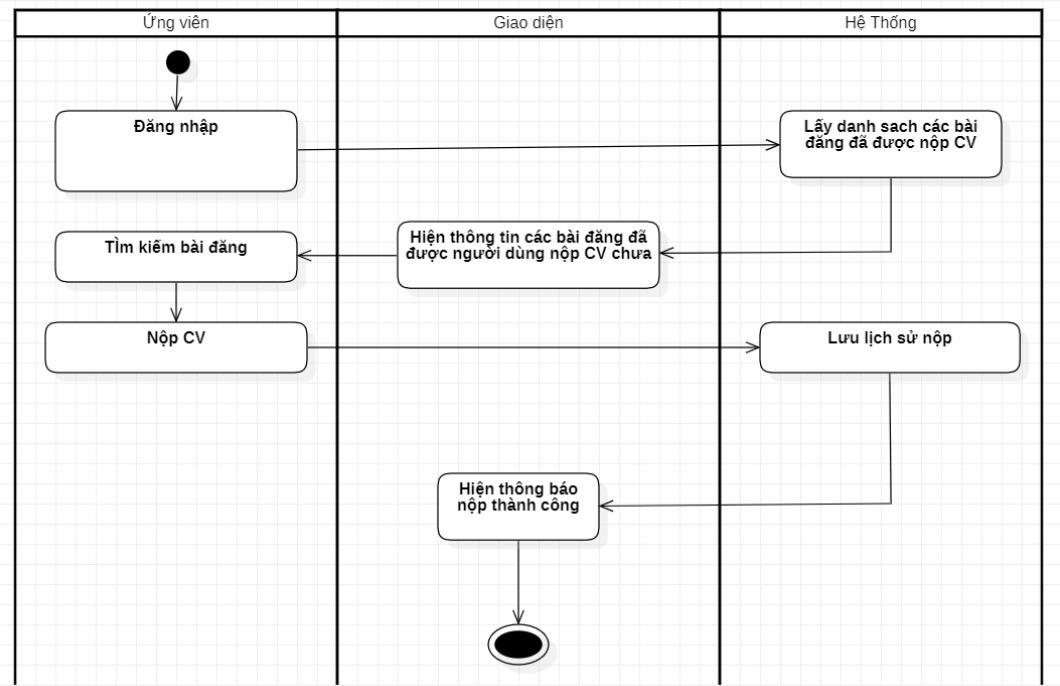
Hình 1.6 : Sơ đồ Activity – Đăng tải CV



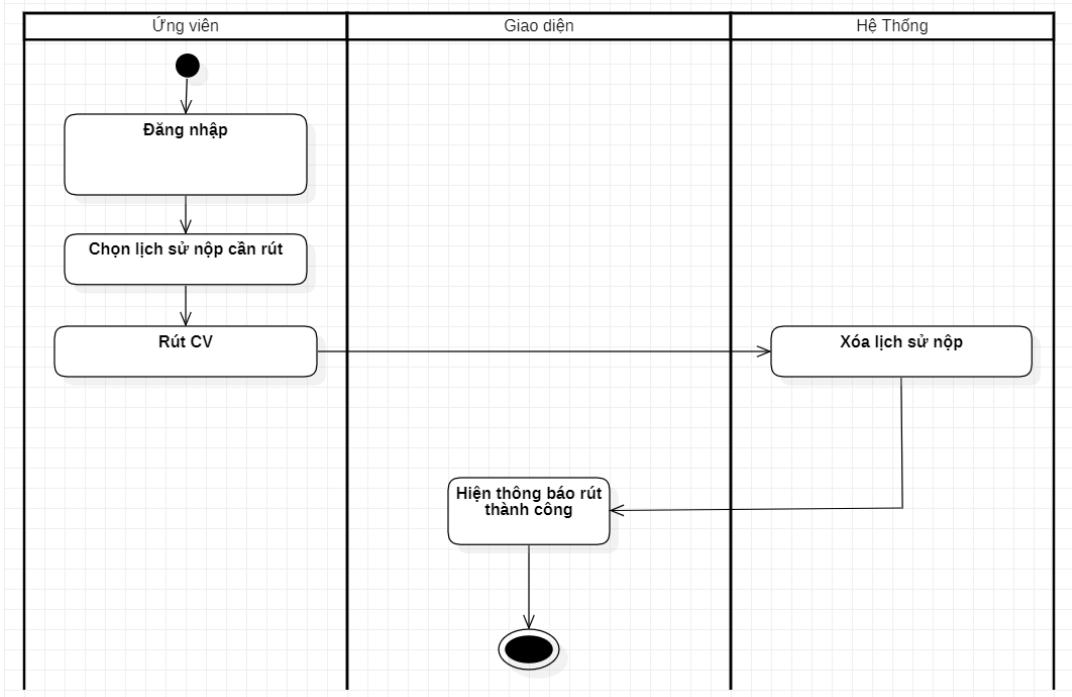
Hình 1.7 : Sơ đồ Activity – Xóa CV



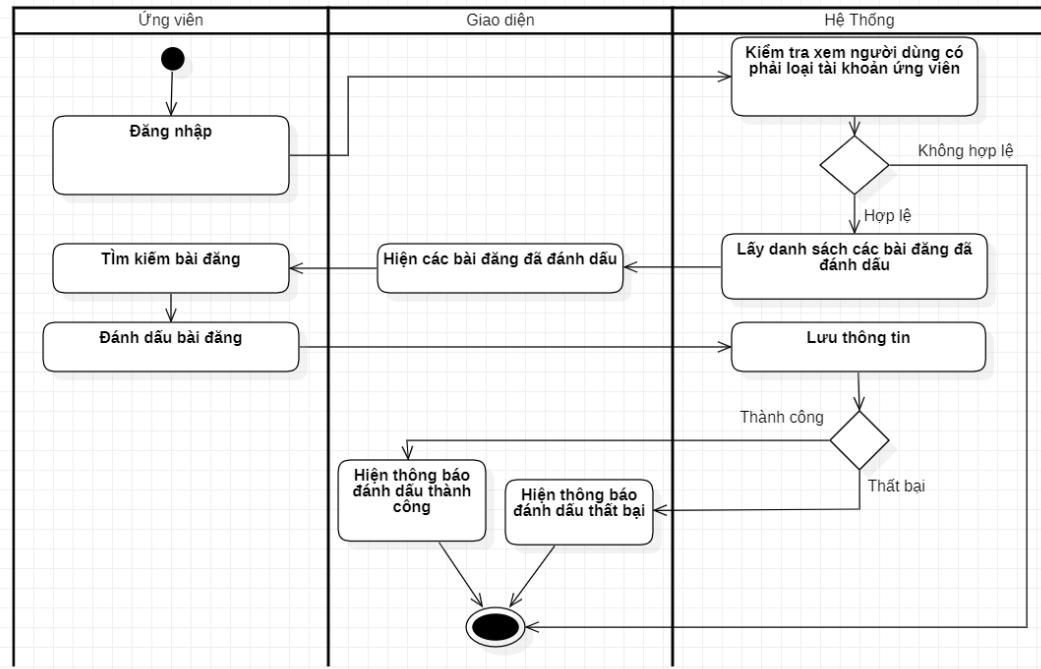
Hình 1.8 : Sơ đồ Activity – Đổi Tên CV



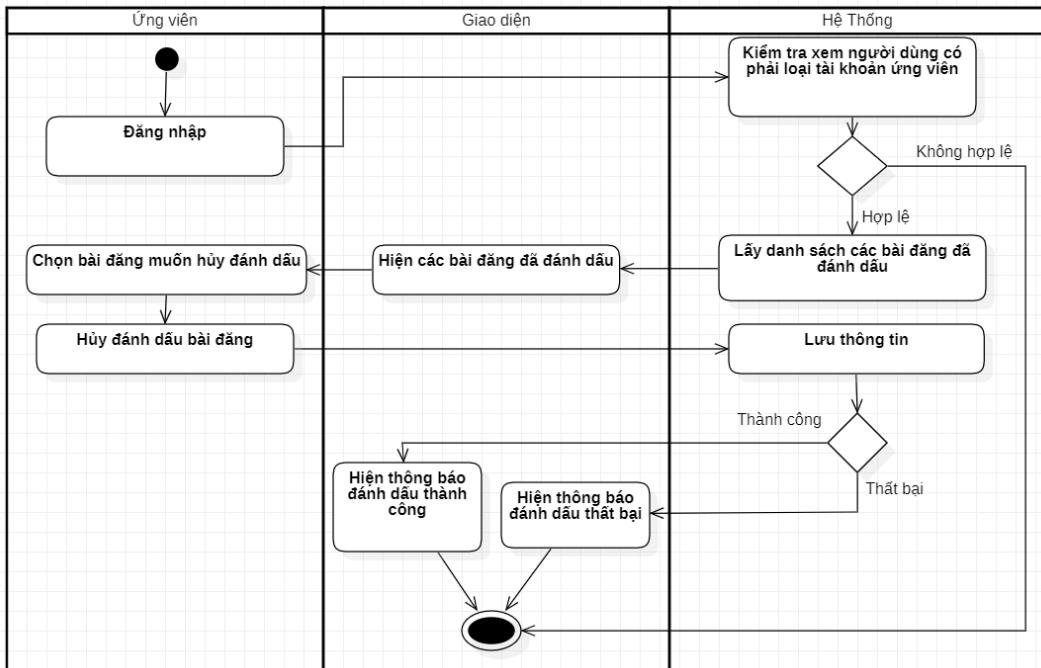
Hình 1.9 : Sơ đồ Activity – Nộp CV



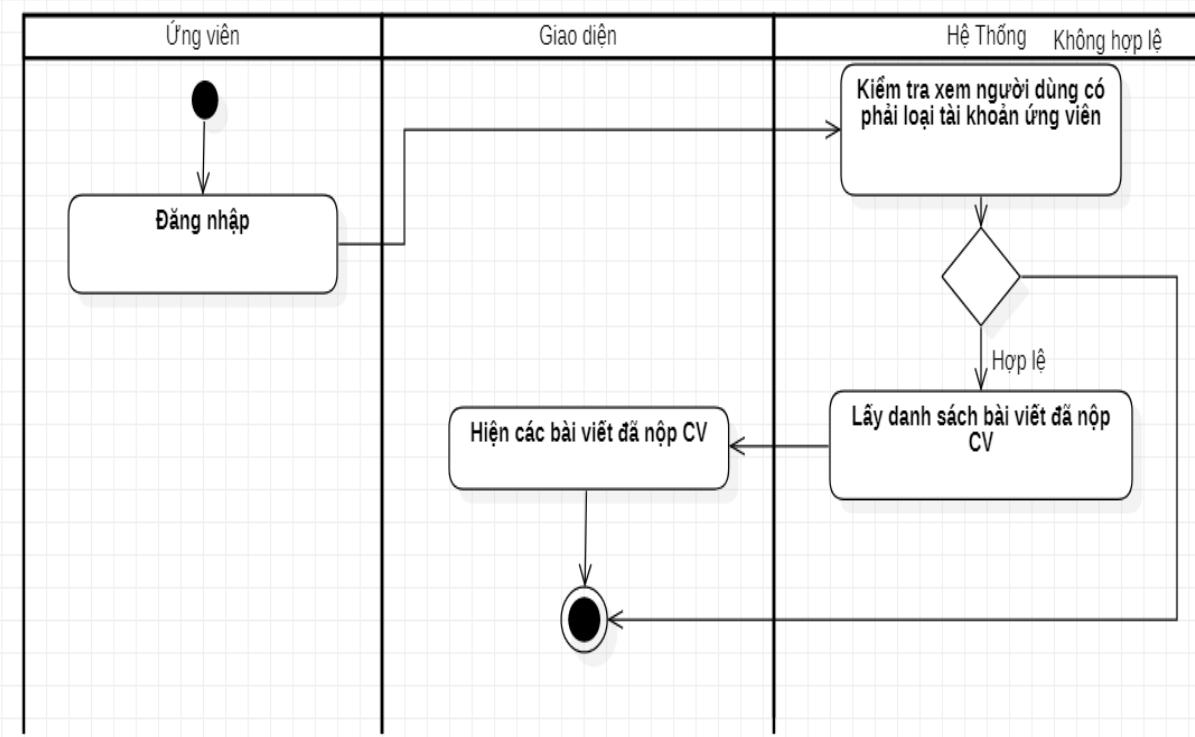
Hình 1.10 : Sơ đồ Activity – Rút CV



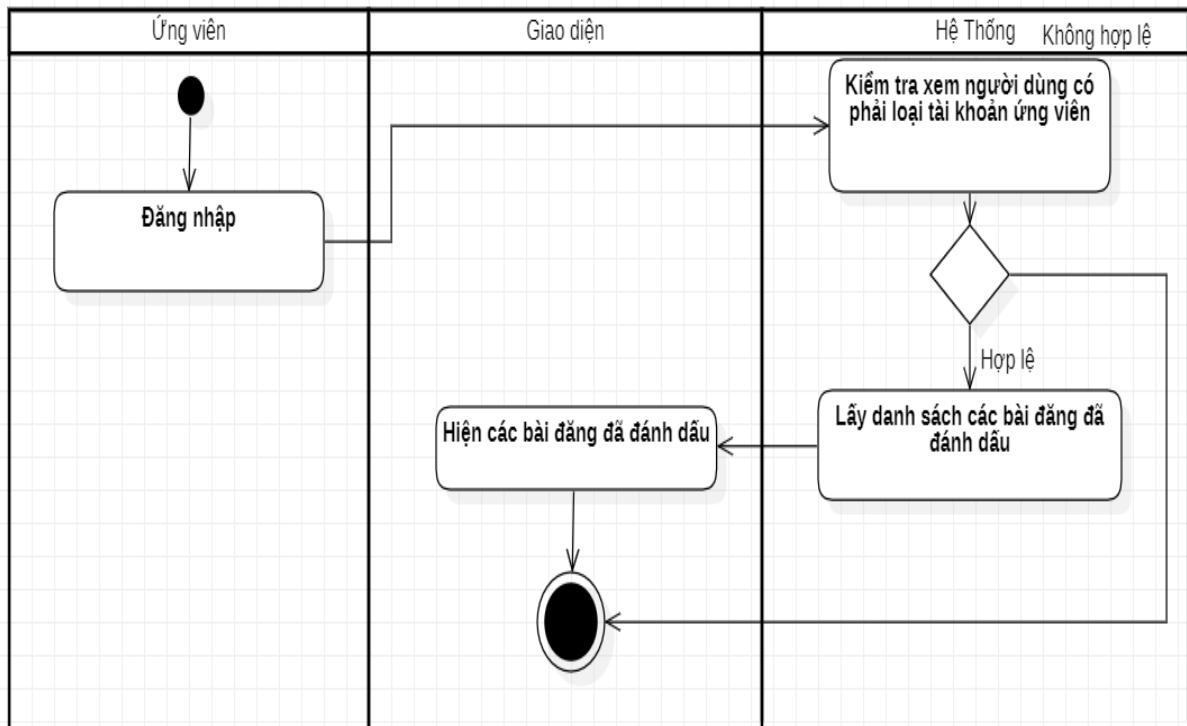
Hình 1.11 : Sơ đồ Activity – Đánh Dấu Bài Đăng



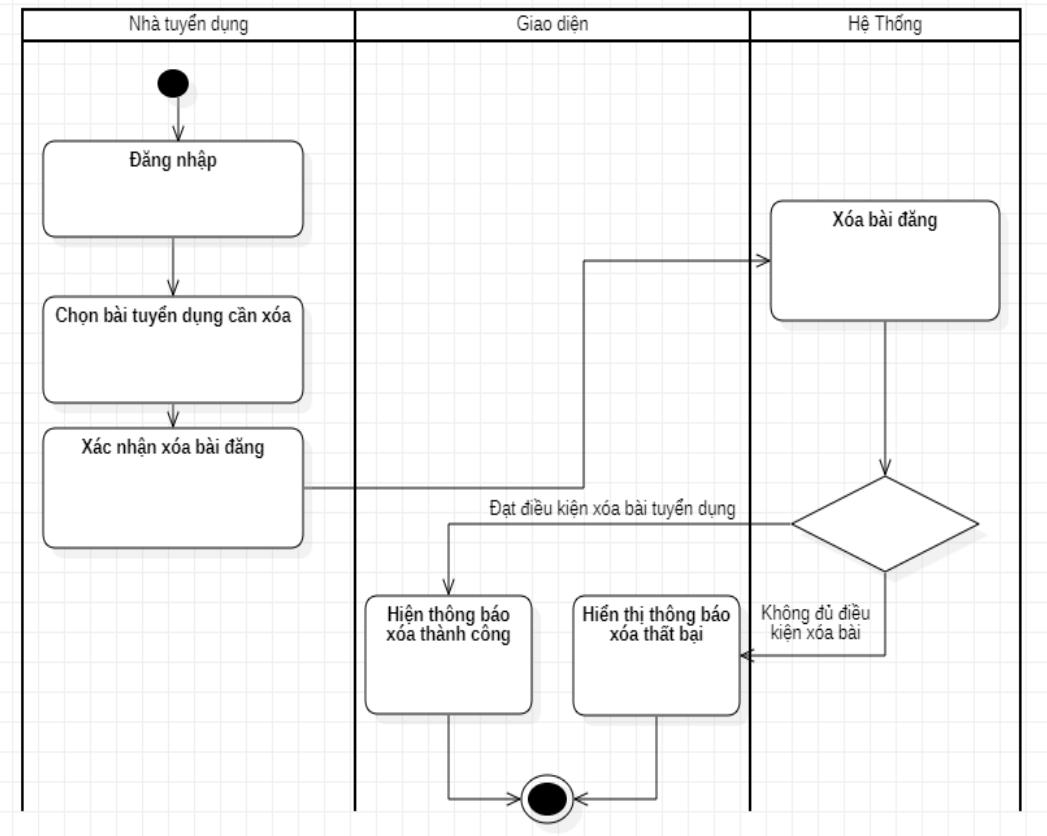
Hình 1.12 : Sơ đồ Activity – Hủy Đánh Dấu Bài Đăng



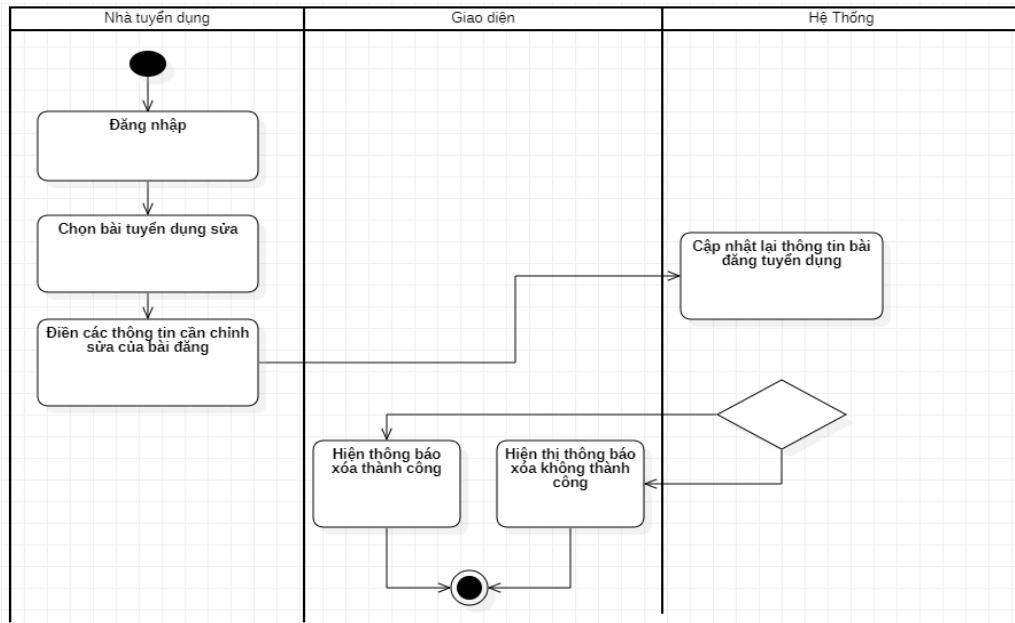
Hình 1.13 : Sơ đồ Activity – Xem Các Bài Viết Đã Nộp CV



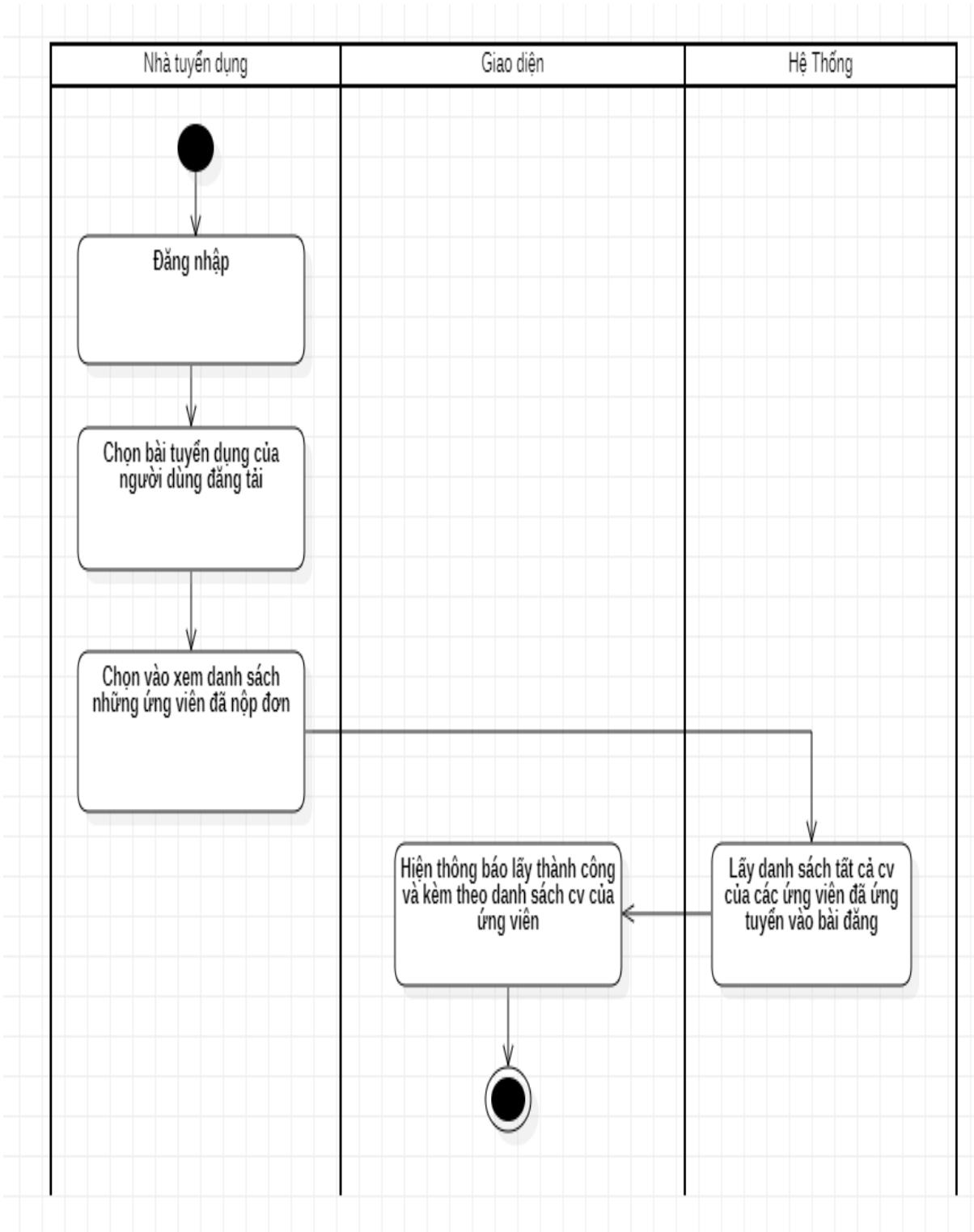
Hình 1.14 : Sơ đồ Activity – Xem Các Bài Viết Đã Đánh Dấu



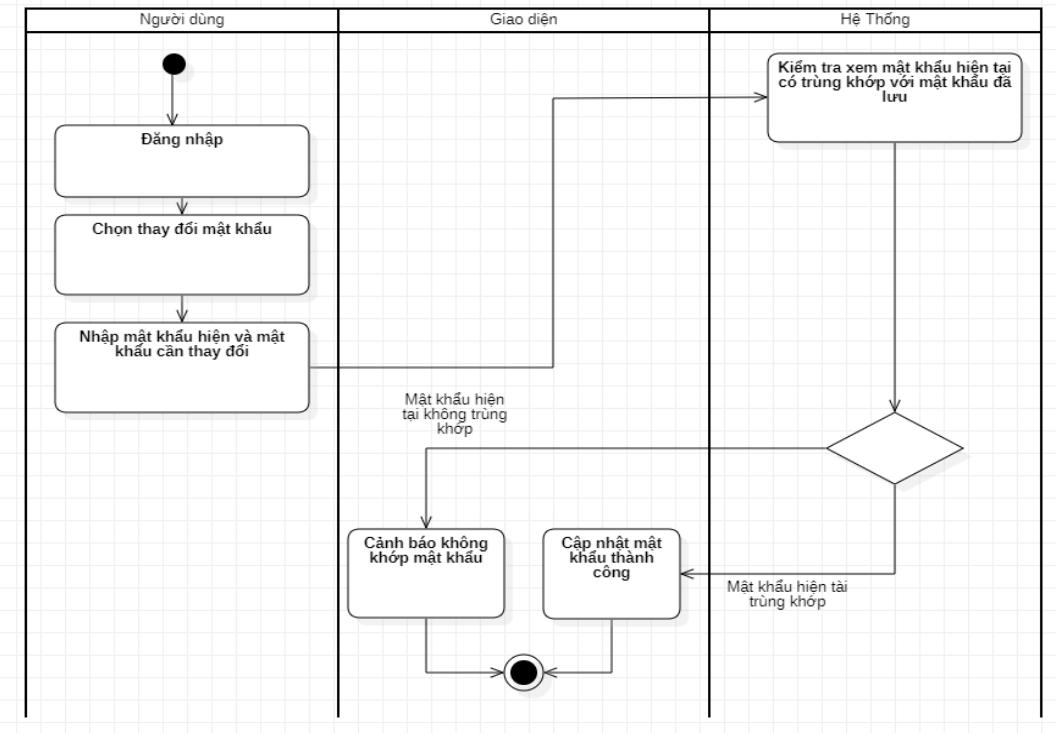
Hình 1.15 : Sơ đồ Activity – xóa bài đăng tuyển dụng



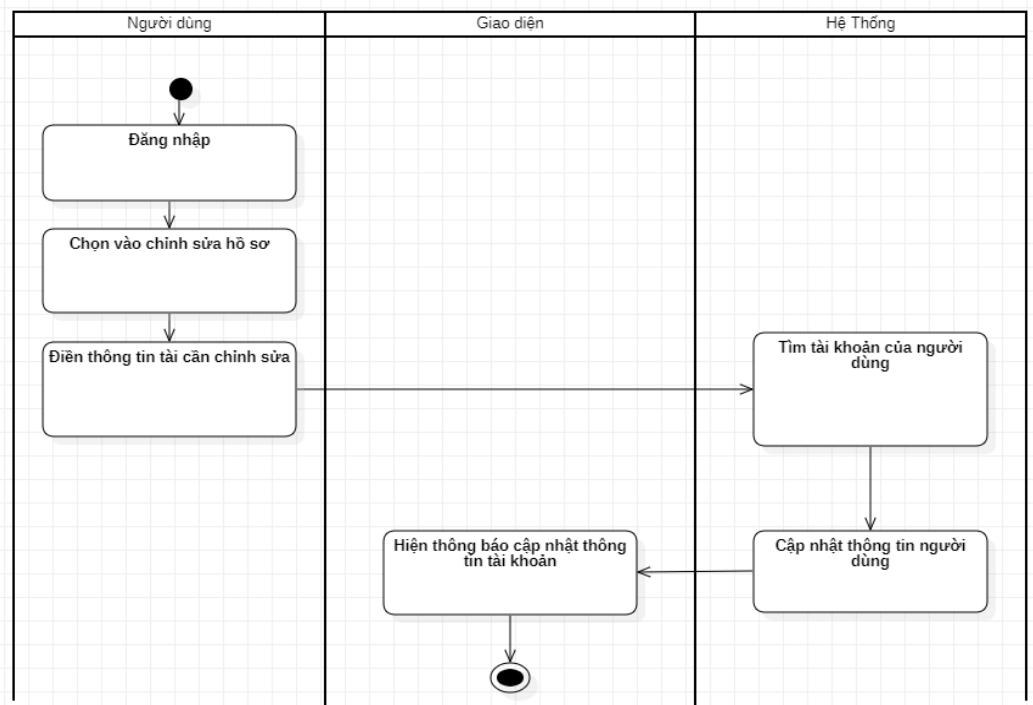
Hình 1.16 : Sơ đồ Activity – sửa bài đăng tuyển dụng



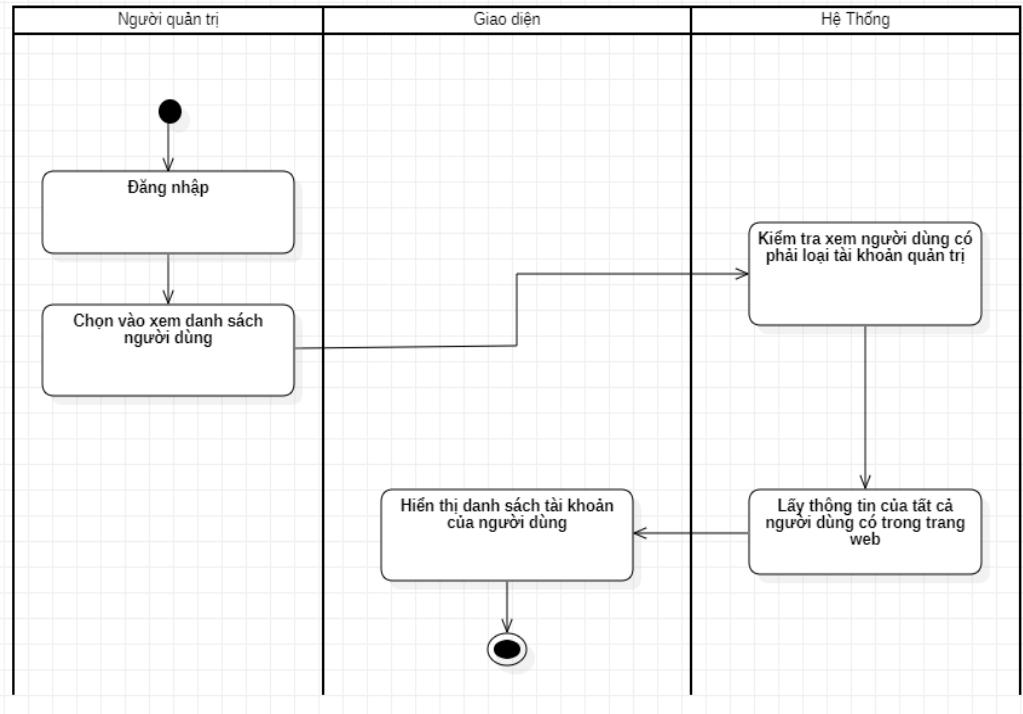
Hình 1.17 : Sơ đồ Activity – xem các đơn ứng tuyển của ứng viên



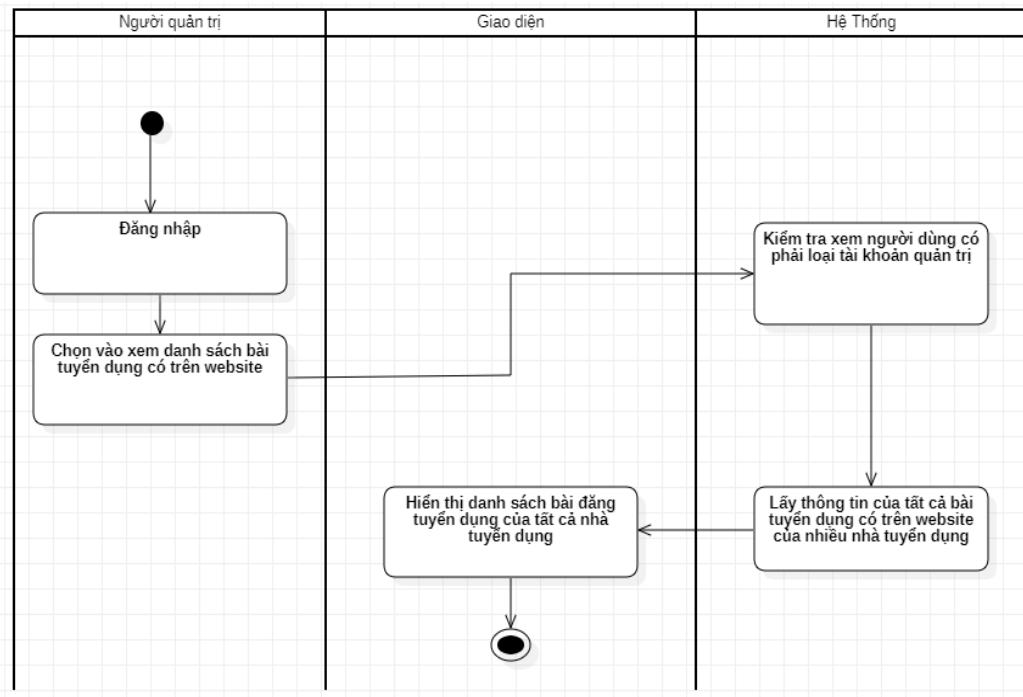
Hình 1.18 : Sơ đồ Activity – cập nhật lại mật khẩu mới



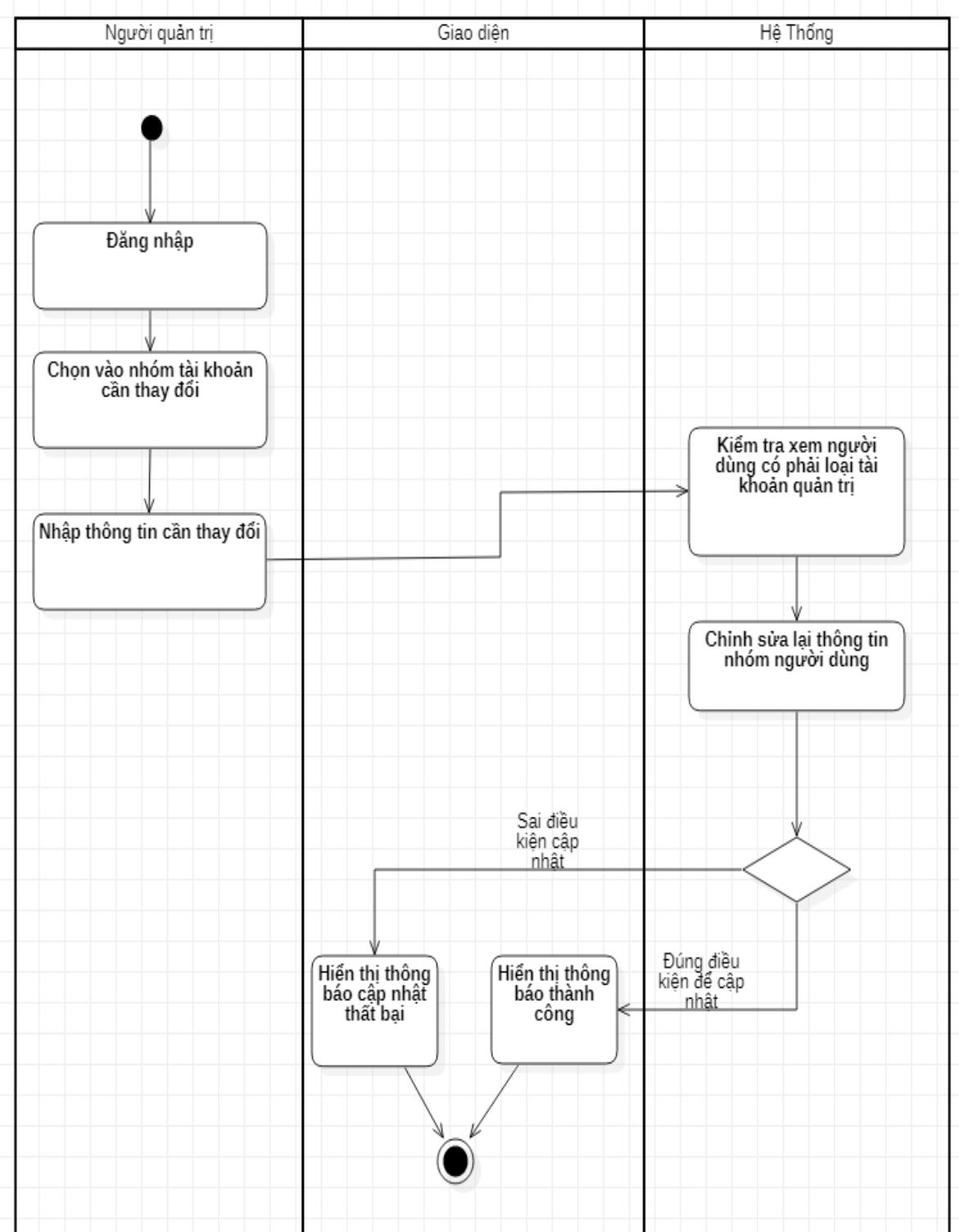
Hình 1.19 : Sơ đồ Activity – cập nhật lại thông tin của người dùng



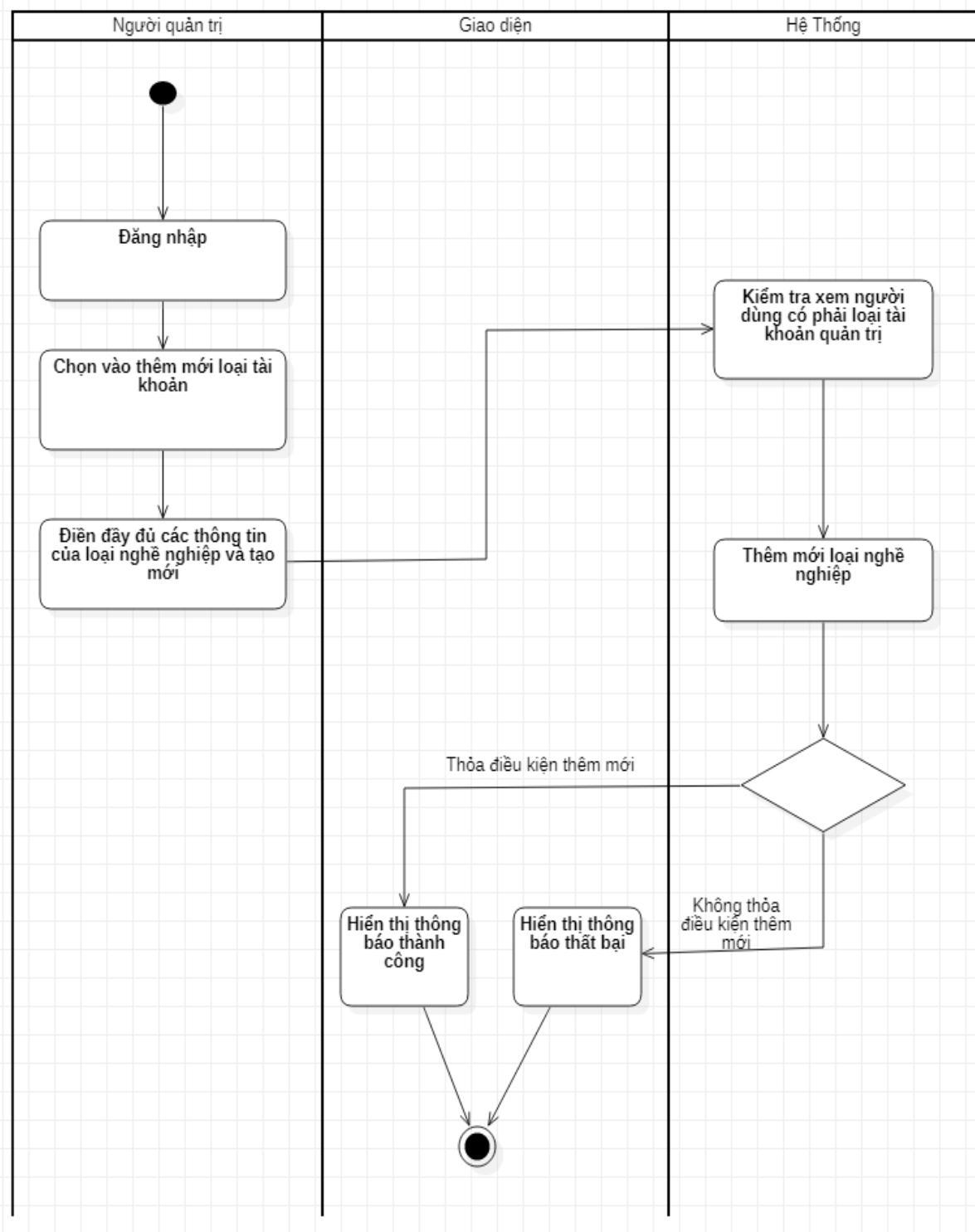
Hình 1.20 : Sơ đồ Activity – xem danh sách người dùng trong hệ thống



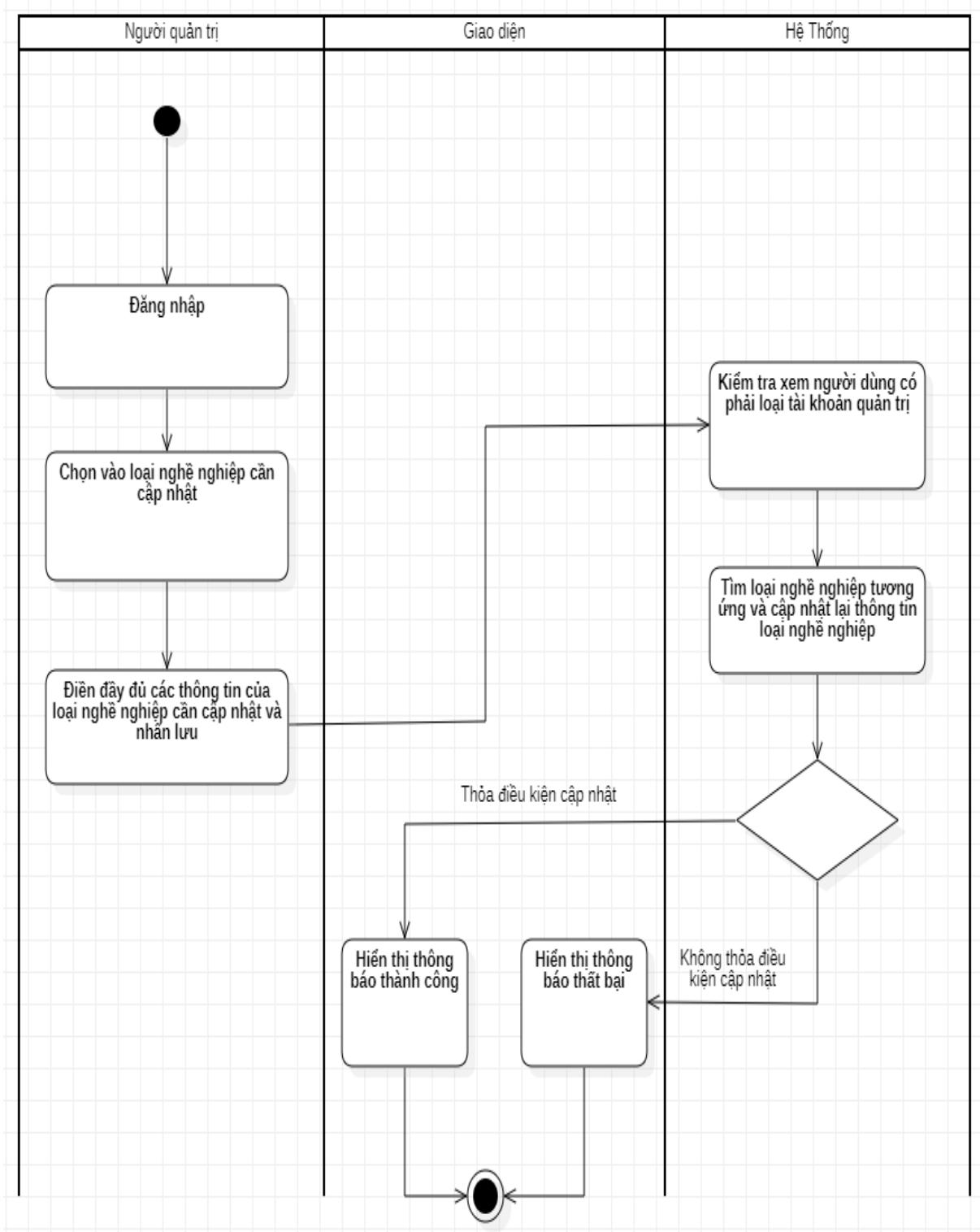
Hình 1.21 : Sơ đồ Activity – xem danh sách bài đăng tuyển dụng của tất cả nhà tuyển dụng



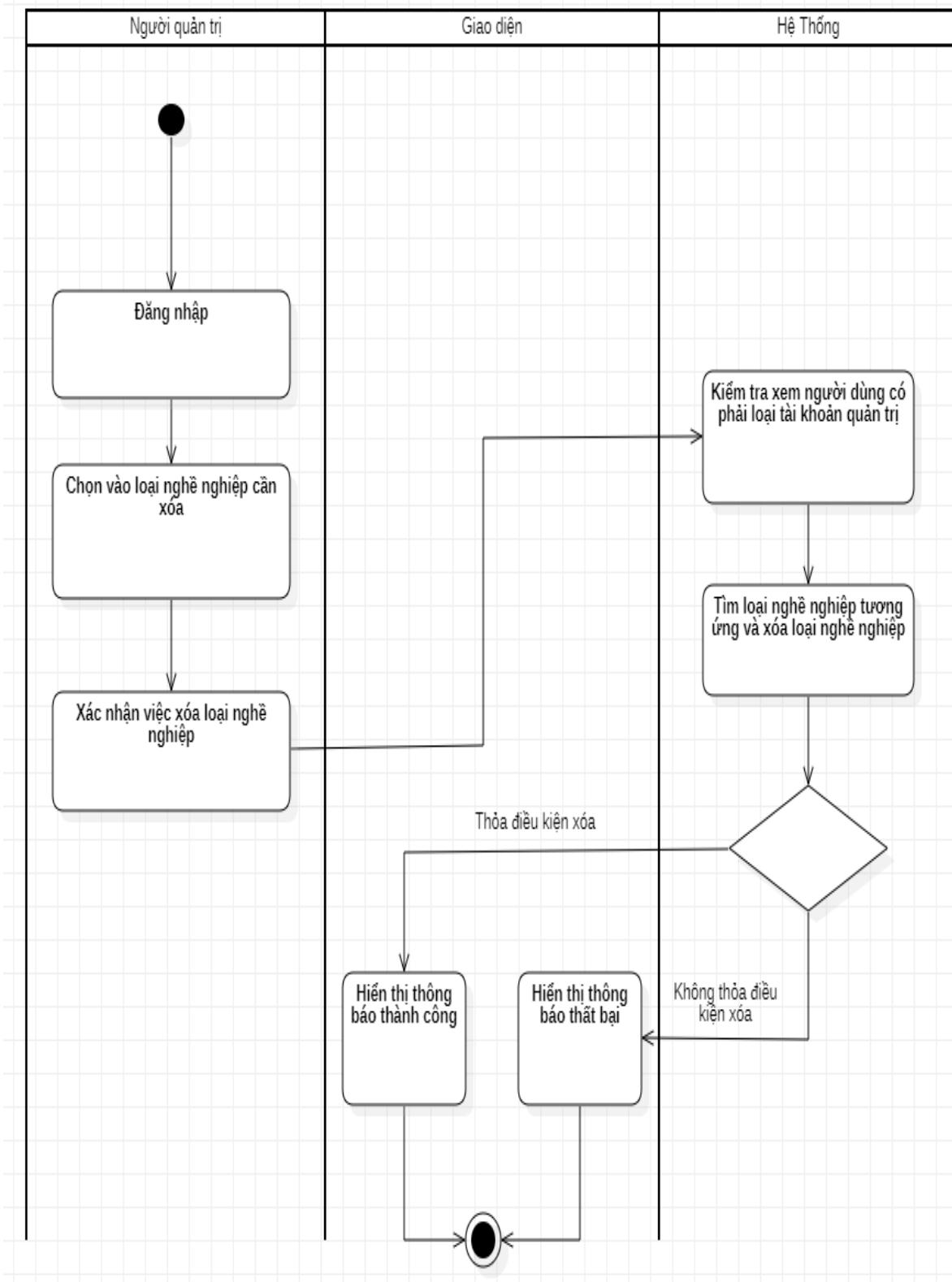
Hình 1.22 : Sơ đồ Activity – Cập nhật thông tin nhóm tài khoản



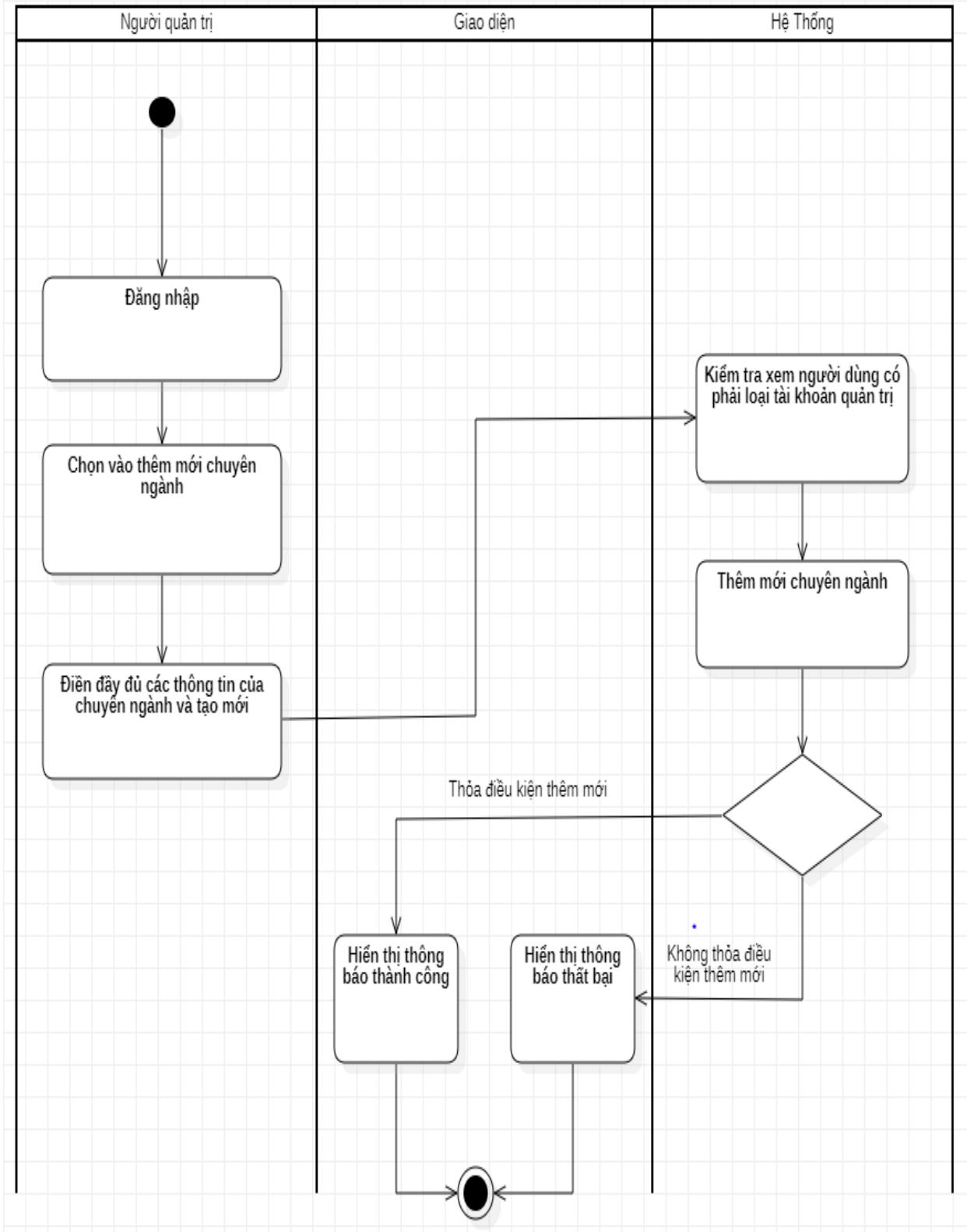
Hình 1.23 : Sơ đồ Activity – Thêm mới loại nghề nghiệp



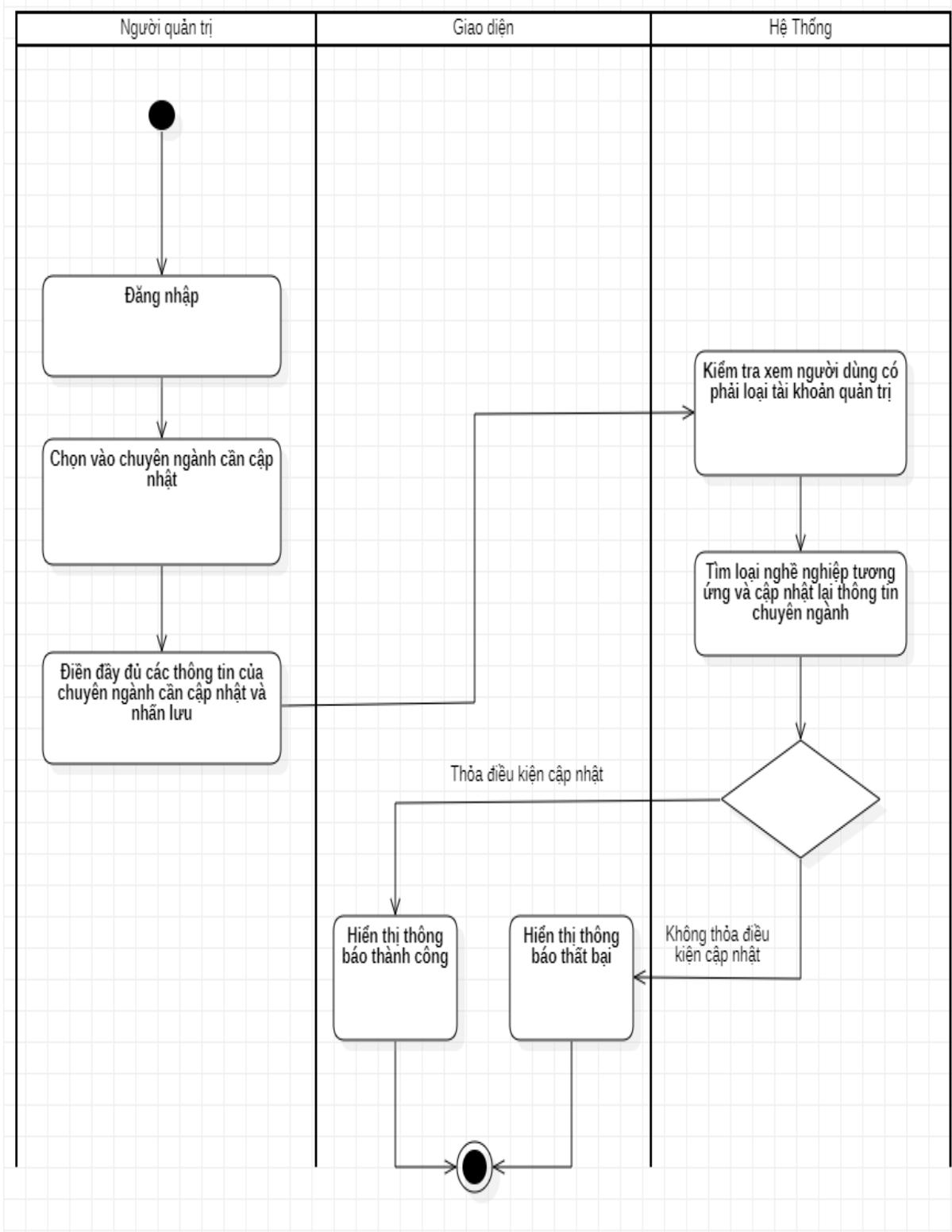
Hình 1.24 : Sơ đồ Activity – Cập nhật lại loại nghề nghiệp



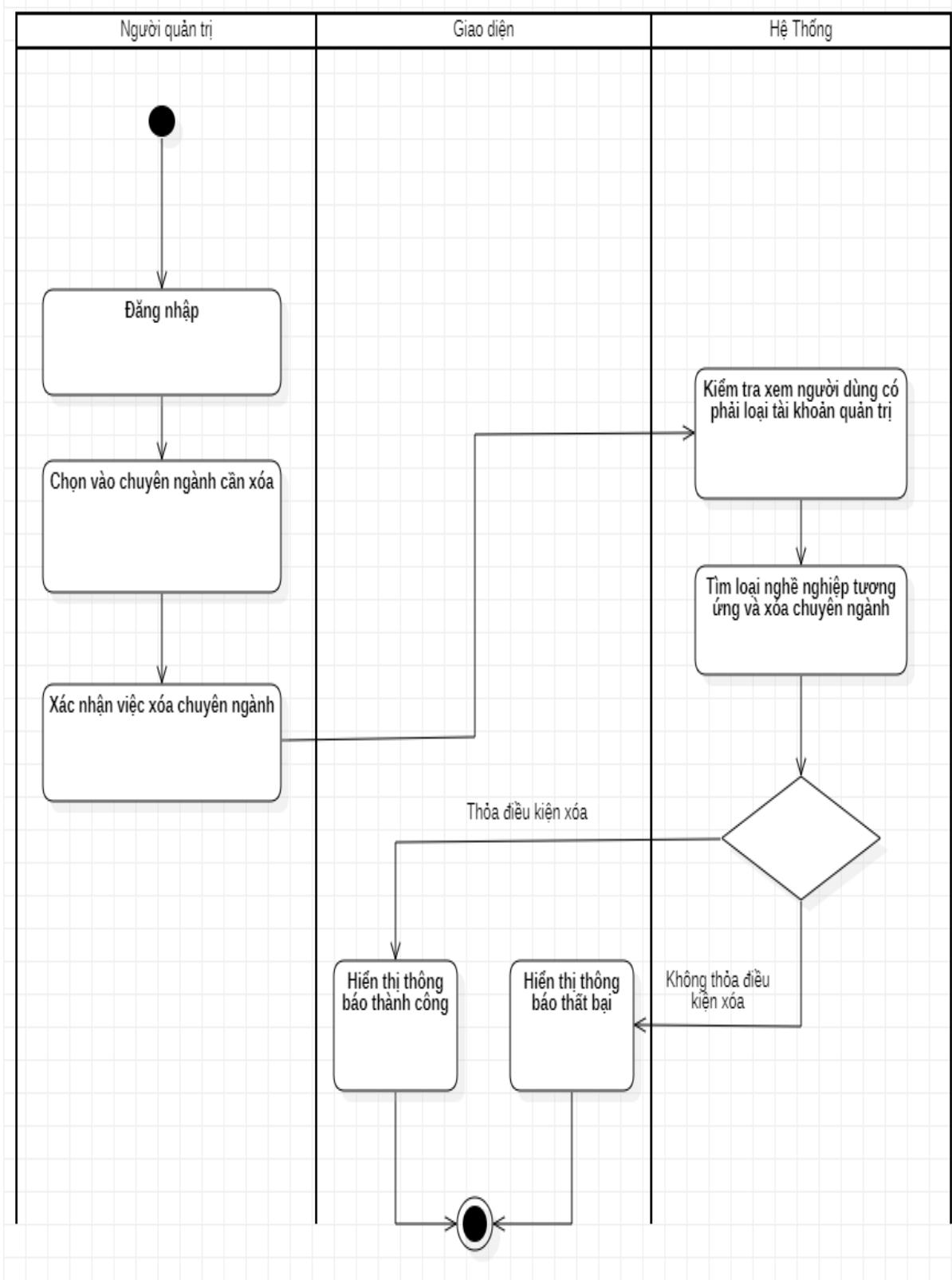
Hình 1.25 : Sơ đồ Activity – Xóa loại nghề nghiệp



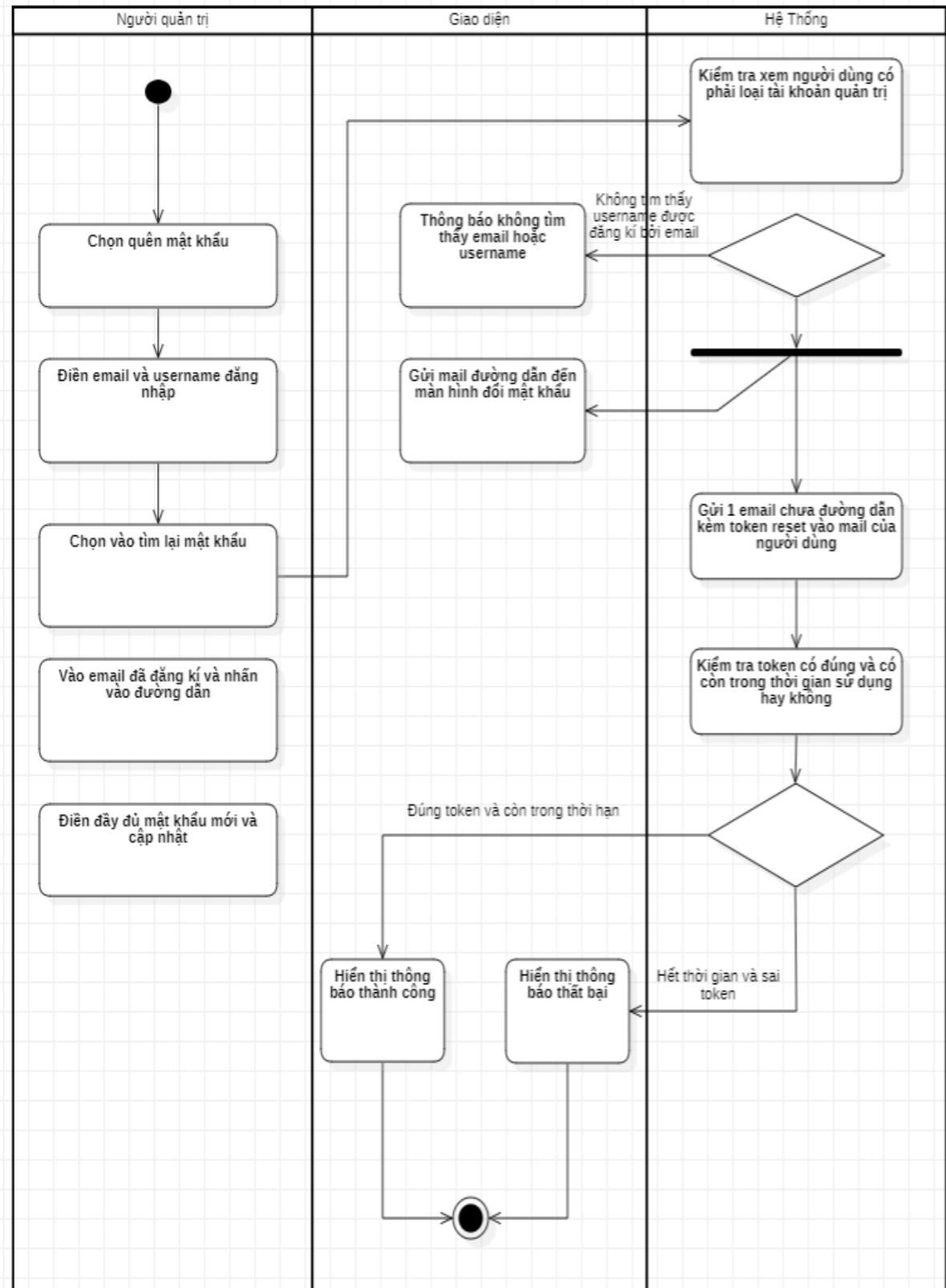
Hình 1.26 : Sơ đồ Activity – Thêm mới chuyên ngành



Hình 1.27 : Sơ đồ Activity – Cập nhật lại thông tin chuyên ngành



Hình 1.28 : Sơ đồ Activity – xóa loại nghề nghiệp



Hình 1.29 : Sơ đồ Activity – Quên mật khẩu

1.2.Giới thiệu về công nghệ và thuật ngữ

1.2.1. Docker

Để một source code được khởi động và hoạt động trơn tru ở một máy tính đòi hỏi máy tính đó phải có đầy đủ tài nguyên như môi trường , các biến , một số thư viện , cũng như một số công cụ để hỗ trợ cho ngôn ngữ và công nghệ đó hoạt động. Với Docker bạn có thể quản lý các tài nguyên của source code một cách dễ dàng và giảm độ trễ giữ việc viết code với việc chạy chúng trong sản phẩm của bản thân.

VD: Nếu một máy tính đang chạy một phiên bản môi trường ví dụ như NodeJs ở phiên bản 16 nhưng ở một máy của người đồng nghiệp lại chạy NodeJs ở phiên bản cao hơn hoặc thấp hơn điều này sẽ dẫn đến việc là khi source code chạy ở máy mình hoạt động bình thường nhưng khi sang máy của bạn hoặc đồng nghiệp lại không được vì không hỗ trợ cách viết JS đó hoặc thiếu gì đó và dẫn đến máy thời gian rất nhiều

Chính vì thế thì Docker(Container) được sinh ra để giải quyết những vấn đề này

- Docker là một công cụ để cung cấp cho việc Build , Deploy, và Run source code ứng dụng hay là một trang web nào đó một cách dễ dàng. Và docker ban đầu được viết bằng ngôn ngữ Python và sau này là GoLang
- Để thực hiện các việc như Build , Deploy , Run như thế thì Docker áp dụng 1 thuật ngữ là Containers. Containers cho phép người lập trình viên đóng gói source code với tất cả các điều kiện mà source code cần để chạy được như là thư viện , môi trường , hay là công cụ hỗ trợ.
- Dựa vào Container sau khi ta khai báo và đóng gói các thư viện và công cụ cần thiết thì ta có thể đưa nó vào máy chủ chạy bất kì hệ điều hành nào cũng được và phổ biến nhất ở đây là Linux, Ubuntu , Centos,... . Và có thể là Windows của Microsoft tùy thuộc vào máy chủ mà người lập trình viên đó thao tác
- Sự độc lập và bảo mật cho phép bạn chạy riêng biệt nhiều Container cùng một máy chủ. Các container hoạt động chiếm một lượng dung lượng của máy chủ tùy theo độ lớn nhỏ của dự án mà ta đưa vào nó.

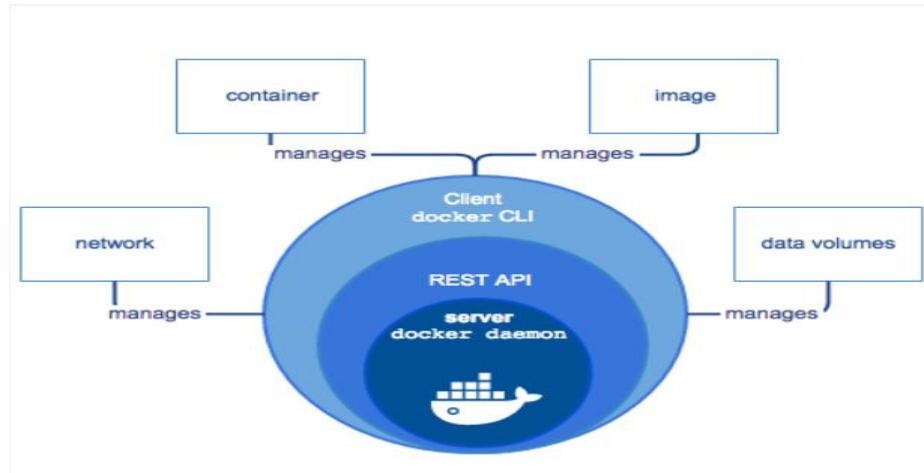
- Docker cung cấp các công cụ và nền tảng để quản lý quá trình hoạt động và vòng đời container của bạn.
 - o Phát triển ứng dụng của bạn và các thành phần hỗ trợ trong containers
 - o Container cũng đóng vai trò là đơn vị cung cấp và thử nghiệm ứng dụng hay trang web của bạn
 - o Nếu bạn có nhu cầu đưa ứng dụng hay trạng web của mình lên môi trường sản phẩm và đưa ra cho mọi người sử dụng dưới dạng là 1 container hoặc là một dịch vụ nào đó cho dù môi trường đó là nằm ở máy của bạn hay là nằm trên một máy chủ khác ở điện toán đám mây hoặc có thể là sử dụng cả hai.

Docker Engine là gì ?

Một tiến trình Docker daemon có nhiệm vụ các bước ban đầu để tạo và quản lý các container, image , volume , network

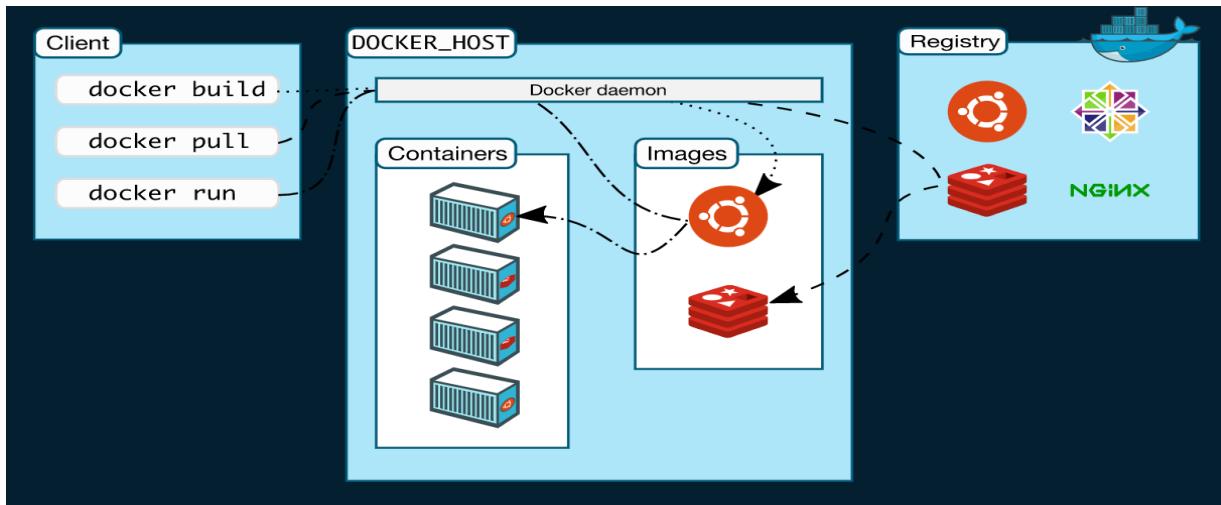
Docker Client sẽ giao tiếp với Docker daemon thông qua API theo chuẩn REST (tức có nghĩa sẽ có các phương thức GET , POST , PUT , DELETE) để yêu cầu Docker daemon thực thi các lệnh mà người dùng gọi đến

Docker CLI được xem là Client hay được hiểu là giao diện tương tác giữa người dùng ở đây là sử dụng các công cụ hỗ trợ như là Command Line (CMD), Window Terminal, Để gửi các yêu cầu tương ứng đến docker daemon thông qua API



Hình 1.30: Sơ đồ về kiến trúc của docker

Kiến trúc của Docker là Client – Server và giao tiếp qua REST API (Cổng mạng và một số phương thức khác) để có thể gửi các yêu cầu đến Docker daemon. Việc giao tiếp này thực hiện các công việc như Run, Deploy, Build các docker container. Ta có thể cấu hình cho Docker Client – Docker Daemon trên cùng 1 máy , hoặc có thể dùng Docker Client để giao tiếp từ xa với Docker Daemon. Và minh họa cho kiến trúc của docker như sau



Hình 1.31: Chi tiết và cách hoạt động và thao tác với docker

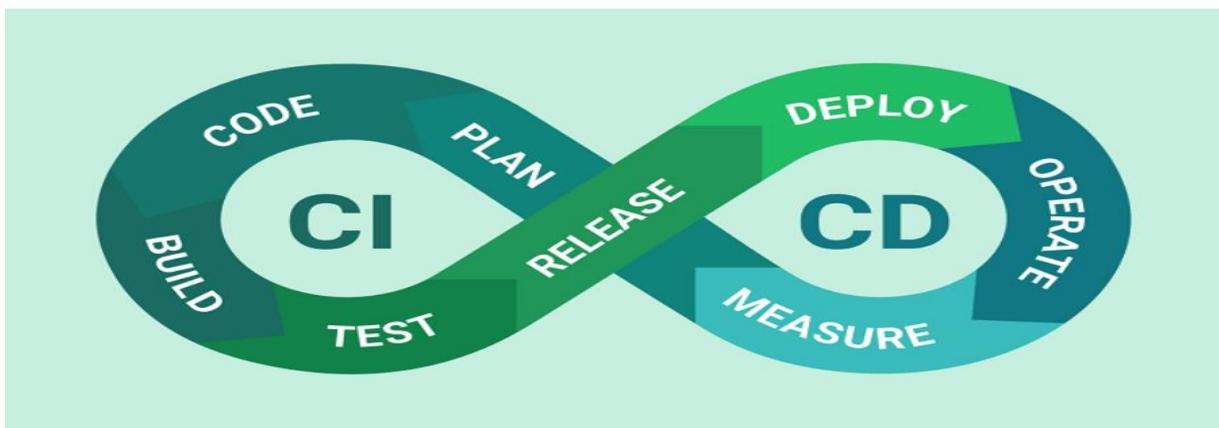
Một số thành phần khác của docker

- Daemon Docker lắng nghe các yêu cầu của người dùng thông qua Docker API và quản lý các Docker đối tượng như image, container, network và volume. Một daemon cũng có thể giao tiếp với các daemon khác để quản lý dịch vụ Docker.
- Docker client là cách tốt nhất để người thiết lập docker giao tiếp với nó. Khi bạn sử dụng các câu lệnh như docker run, client thông qua Docker API sẽ gửi các câu lệnh này đến dockerd, thực hiện chúng. Docker client có thể giao tiếp với nhiều hơn một daemon Docker.
- Docker registry sẽ là nơi lưu trữ các hình ảnh Docker. Docker Hub là nơi lưu trữ Docker image public (public registry) mà bất kỳ ai cũng có thể sử dụng và Docker được định cấu hình mặc định để tìm kiếm hình ảnh trên Docker Hub. Ngoài ra, người dùng có thể cấu hình các tư vấn đăng ký khác nhau để lưu trữ hình ảnh Docker.

Khi người dùng sử dụng câu lệnh “docker pull” hoặc “docker run”, các image only sẽ được tải về dựa trên registry đã được cấu hình trước đó. Khi người dùng sử dụng câu lệnh “docker push”, image cũng sẽ được tải lên docker hub mà người dùng đã cấu hình từ trước đó.

- Docker Image là template read-only (chỉ cho phép đọc) được xem là một bản ghi của phần mềm và có thể chạy để tạo Docker container. Image sẽ được sử dụng để đóng gói các ứng dụng và các thành phần đi kèm của ứng dụng, được lưu trữ ở server hoặc một nơi lưu trữ như là docker hub.
- Docker Container được xem là 1 hình thức chạy của image. Bạn có thể khởi tạo, dừng hoặc xóa container bằng cách sử dụng Docker API hoặc CLI. Bạn có thể kết nối container đến 1 hoặc nhiều network
- Docker volume được thiết kế để làm nơi lưu trữ các dữ liệu độc lập của riêng container đó với quá trình container còn hoạt động.
- Docker network : Cung cấp một mạng lưới riêng biệt mà chỉ tồn tại giữa container và server, giúp các container có thể giao tiếp được với nhau một cách dễ dàng.
- Docker Service cho phép bạn mở rộng các container thông qua nhiều Docker daemon, Service docker ta có thể dụng chính service Docker để xây dựng 1 bản ghi trong chính docker đó.

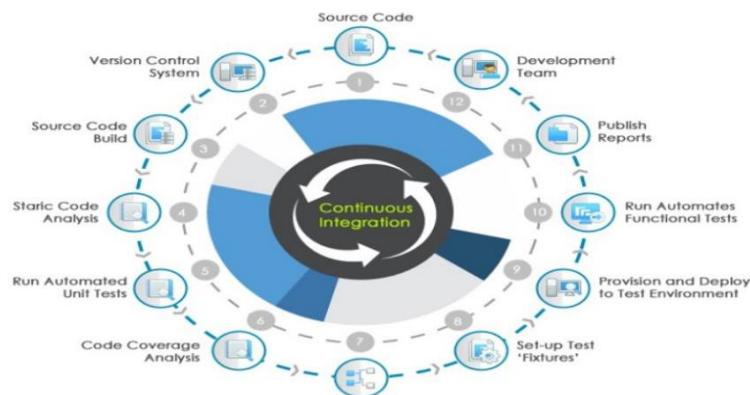
1.2.2. CI/ CD



Hình 1.32: Quá trình hoạt động của CI/CD

CI (Continuous Integration) là cách thức phát triển phần mềm bằng cách tích hợp liên tục có nghĩa là nếu trong một nhóm những người cùng một dự án khi thực hiện một tính năng hay tham gia vào việc bảo trì một tính năng nào đó và sinh ra các đoạn mã mới thay cho đoạn mã cũ và đưa đoạn mã trên nhánh của người đó lên nhánh tổng (main) hoặc là nhánh develop thì ở đây CI sẽ thực hiện công việc của nó là tự động chạy toàn bộ source code cùng với đoạn mã mới của người viết đoạn mã vừa mới đưa lên và kiểm tra xem độ chính xác về mặt quy tắc viết mã của ngôn ngữ mà source đó sử dụng hay không. Nếu không thì sẽ tiếp tục bước tiếp theo nhưng khi có vấn đề xảy ra thì sẽ báo lỗi cho người viết đoạn mã đó chưa khớp với đoạn mã trên nơi tổng mà các nhánh của các thành viên đều về đó, và cũng như đưa ra những gợi ý cho người viết mã là một đề xuất để sửa. CI cũng đảm bảo mã nguồn của bạn sau gom mã của thành viên trong nhóm lại sẽ có thể hoạt động ở bất kỳ điều kiện nào và không xảy ra lỗi. Quá trình Build Run và Deploy sẽ được chạy và cho người quản lý source code tiến trình nó xảy ra như thế nào.

Quá trình đưa mã code từ nhánh cá nhân lên nhánh chính (main) thì quá trình CI hoạt động sẽ được hoạt động tự động nếu người quản lý source code cấu hình và cài đặt trước . Nên quá trình gom source code -> kiểm tra -> phản hồi sẽ diễn ra liên tục và tự động. Giúp người xây dựng mã có thể tiết kiệm thời gian để xây dựng tính năng khác hoặc sửa lỗi nếu có và sau khi quá trình kết thúc thì người quản lý có thể quay lại xem được quá trình build và test.



Hình 1.33: Quá trình phát triển ứng dụng.

Lợi ích của việc sử dụng CI

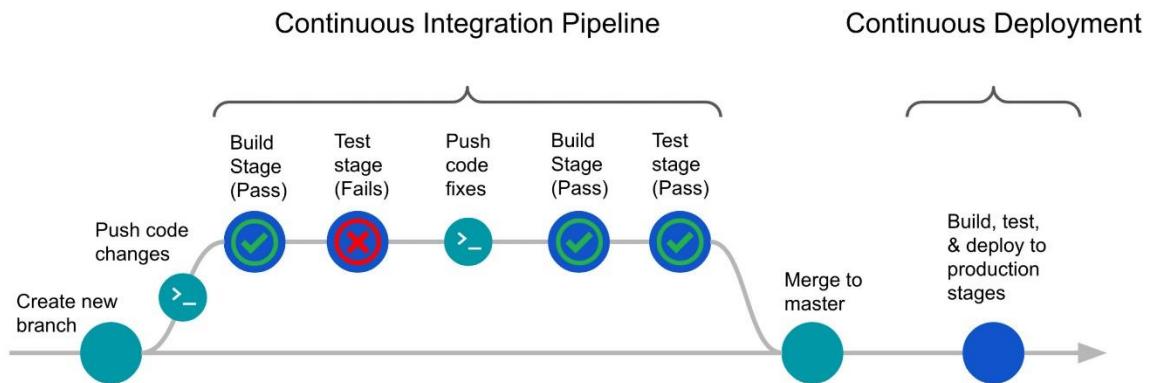
- Giảm thiểu rủi ro nhờ việc phát hiện lỗi và sửa lỗi sớm, tăng năng suất cũng như chất lượng phần mềm nhờ việc tự động test.
- Giảm thiểu những quy trình thủ công lặp đi lặp lại cho việc tích hợp thủ công (build css, js, migrate, test...), thay vì đó là build tự động, chạy test tự động
- Sinh ra phần mềm có thể deploy ở bất kì thời gian, địa điểm

CD là gì : Thì đi kèm với CI là việc tích hợp liên tục thì còn một quá trình nữa gọi là chuyển giao liên tục thì nếu source code qua quá trình CI không có lỗi xảy ra thì sẽ sang một giai đoạn gọi là CD để tiến hành kiểm tra những thay đổi về mã nguồn đã được build lên và mã nguồn trong môi trường kiểm thử trước đó. Từ đó nhanh chóng nhận biết được lỗi và sửa chữa lỗi đó kịp thời vì đôi khi sản phẩm ở local sẽ hoạt động nhưng lúc build hoặc deploy lại không hoạt động.

Do sự tiện lợi của 2 quá trình này thì người quản lý mã nguồn thường tích hợp cả hai CI/CD với nhau để tiết kiệm thời gian , chi phí và tài nguyên bộ nhớ cho 2 quá trình này, CD sẽ sử dụng giai đoạn cuối cùng của CI để bắt đầu quá trình chuyển giao của chính nó

Trong quá trình CD thực hiện tiến trình của mình thì có thể mã nguồn sẽ trải qua một số phần testing được người quản lý mã nguồn thiết lập và một số bài test bao gồm UI Testing , API Testing, và Trong quá trình deploy cũng sẽ được Testing.... CD có một thuật ngữ được gọi là Deployment Pipeline người quản lý quan sát và chia nhỏ quá trình deploy từ bản build đưa lên server ra sản phẩm đầu cuối hoặc sản phẩm test thành các giai đoạn. Mỗi giai đoạn sẽ đáp ứng những mục đích riêng bao gồm xác định chất lượng của các tính năng và tránh lỗi ảnh hưởng đến người dùng. Pipeline sẽ thể hiện phản hồi cho nhóm lập trình viên thông báo khi có lỗi trong việc bổ sung tính năng mới hoặc sửa chữa các tính năng đã có sẵn. Deployment Pipeline là quy trình để chuyển phần mềm từ sản phẩm local đến bản build và đến tay người dùng cuối

Việc kết hợp 2 quá trình này thì ta sẽ được chu trình hoạt động của CICD gói gọn trong 3 bước là Xây dựng -> Kiểm tra -> Triển khai



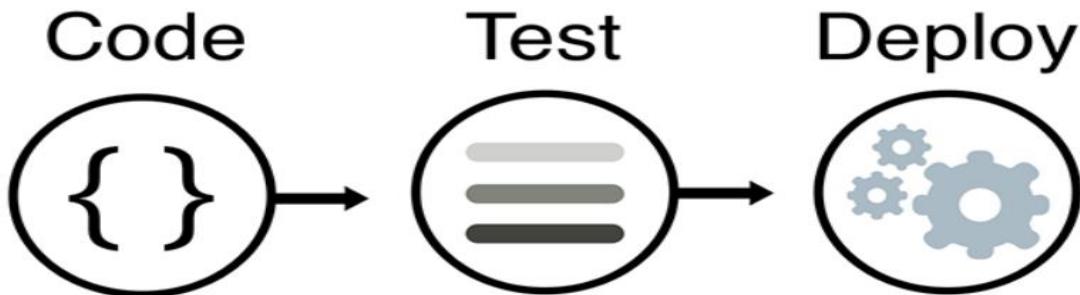
Hình 1.34: Cách hoạt động của CI/CD

Lợi ích của CI/CD

Giảm thiểu rủi ro không đáng có : Trong quá trình từ mã nguồn thành bản build và đưa bản build lên server nếu phát hiện ra lỗi sẽ dừng quá trình lại và báo cho nhà lập trình viên để sửa giúp giảm thiểu rủi ro

Thay đổi code nhỏ : Sinh ra bản build và deploy nó ngay khi mã nguồn có sự thay đổi dù là nhỏ nhất

Hạn chế những ảnh hưởng của lỗi: CI/CD được thiết kế với hệ thống sao cho khi có lỗi nào đó xảy ra thì những kết quả không ổn sẽ bị giới hạn trong phạm vi ảnh hưởng chỉ riêng container ta chỉ định build và deploy



Hình 1.35: Tổng quan về quá trình hoạt động với CI/CD

Nhược điểm của CI/CD là gì ?

- Cần kiến thức cũng như thời gian cập nhật nhanh chóng

- Hiểu rõ nguyên lý vận hành của chương trình.
- Nhiều hệ thống không hỗ trợ.
- Đòi hỏi nguồn tài nguyên lớn.

1.2.3. VPS

VPS là viết tắt của (Virtual Private Server) hay còn gọi là máy chủ riêng ảo. Được cung cấp và cho thuê bởi các công ty dịch vụ server, máy chủ.

VPS hoạt động như một server để người dùng tạo ra trang web hay ứng dụng trên các tài nguyên ta cài đặt được giành riêng cho máy chủ đó.

Người cài đặt VPS có toàn quyền kiểm soát bằng người dùng có quyền cao nhất(Root , quyền admin), và thông qua SSH key để điều khiển cài đặt các thư viện, hoặc cấu hình VPS theo ý muốn của người cài đặt.

Thường thì VPS sẽ được thuê và có những option các hệ điều hành có sẵn khi người muốn thuê lựa chọn các dịch vụ và trả tiền để thành toán phí duy trì của cái máy ảo đó ở máy chủ của công ty hosting đó một số hosting phổ biến như AWS , Digital Ocean , Microsoft Azure , Ở Việt Nam thì có Mắt Bão , trên điện toán đám mây

VPS an toàn và ổn định tùy theo tình trạng mạng của bạn kết nối và ổn định hơn so với host website vì bạn không chia sẻ không gian lưu trữ với người khác. Và nó cũng có giá thành thấp hơn so với thuê một server riêng

Thường thì các máy chủ ảo sẽ không có tên miền để thay thế cho việc truy cập đến máy chủ đó mà phải truy cập qua IP của VPS đó được sinh ra khi bạn tiến hành thuê VPS đó

VPS là một server ảo hoạt động tương tự như một máy chủ vật lý. Máy chủ, hay server, là một máy tính chứa source code và cơ sở dữ liệu cho ứng dụng của bạn, Môi trường và các thư viện cần thiết. Bất kể khi nào người dùng truy cập vào ứng dụng của bạn trên trình duyệt thì sẽ gửi yêu cầu đến con server của bạn và lúc này dựa vào đường truyền internet các file cần thiết để chạy 1 trang web sẽ được truyền xuống máy tính của người dùng và hiện thị nó trên trình duyệt

Trên thực tế thì VPS là máy chủ ảo được cấu hình qua mạng như nó sẽ được hình thành trên một server vật lý, và mỗi VPS là sẽ có phần tài nguyên riêng biệt của nó và không bị ảnh hưởng bởi các VPS khác

Các yếu tố lưu ý đến VPS khi tiến hành thuê

- **Ram** : Trong máy tính, RAM là loại bộ nhớ chính, nếu VPS của bạn có nhiều RAM thì khả năng truy xuất dữ liệu càng tốt. Bởi vì khi sử dụng VPS bạn sẽ cần RAM để xử lý các vấn đề như xử lý truy vấn nhập xuất của database
- **SWAP** hay là bộ nhớ ảo có nhiệm vụ lưu lại các hành động trong quá khứ trong trường hợp bộ nhớ RAM bị quá tải (overload). Hay nói cách dễ hiểu hơn, SWAP là nơi lưu trữ trên ổ cứng thay vì một bộ nhớ độc lập. Tuy nhiên, không phải VPS nào cũng hỗ trợ SWAP
- **Disk** : Ổ cứng/Đĩa cứng hay còn gọi là Disk là không gian lưu trữ được sử dụng để lưu các file cài đặt của hệ điều hành và các file của mã nguồn website. Có 2 loại ổ đĩa được sử dụng phổ biến, đó là:
 - o HDD (Hard Disk Drive): Là loại ổ đĩa thông dụng được sử dụng trên máy tính.
 - o SSD (Solid State Drive): SSD hay còn gọi là ổ cứng bán dẫn được sử dụng để lưu trữ dữ liệu nhưng tốc độ truy xuất dữ liệu nhanh hơn so với HDD nhiều lần

Do vậy, giá thành của ổ cứng SSD sẽ có đắt hơn so với loại ổ HDD.

- **CPU Core** : là lõi xử lý của CPU. Một máy chủ riêng sẽ có lượng core nhất định và nó được chia cho các VPS. Thông thường, số core càng cao thì khả năng xử lý dữ liệu càng tốt. Số lượng core trung bình ở các gói VPS là từ 1 core đến 3 core.
- **Bandwidth và transfer** : 2 từ này đều có nghĩa là băng thông, chính là lưu lượng được phép truyền tải dữ liệu đi và các tập tin có trong server.
- **IP (Internet Protocol)** : là số lượng địa chỉ IP mà nhà cung cấp dịch vụ VPS sẽ cấp cho bạn, và có thể điều hướng tên miền tới địa chỉ IP này.

Các công ty cho thuê dịch vụ máy chủ ảo riêng biệt như thế sẽ cho người thuê toàn quyền quyết định về hệ điều hành, cũng như cho phép chọn cấu hình của phân vùng VPS đó như sử dụng bao nhiêu RAM xử lý , Bao nhiêu GB bộ nhớ , Thì Ram và bộ nhớ càng cao ứng dụng của bạn sẽ hoạt động một cách trơn tru hơn nhưng bù lại chi phí của của người thuê bỏ ra để duy trì nó sẽ càng cao hơn.

Ưu điểm của VPS là gì ?

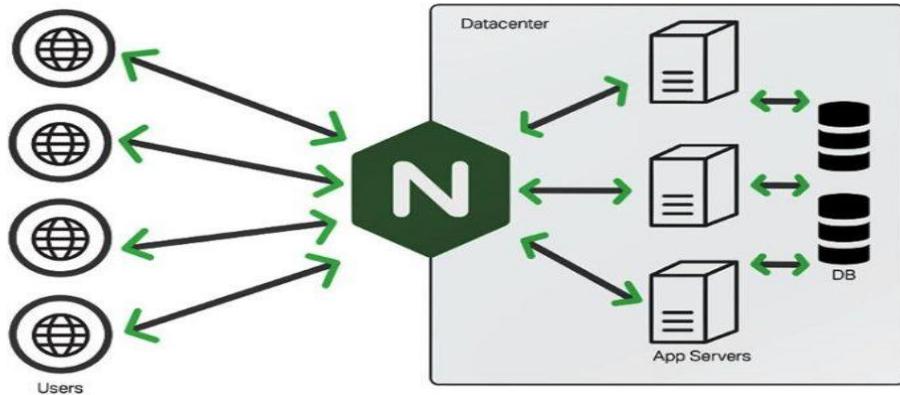
- Nhanh hơn và độ đáng tin cậy cao hơn server thường
- Đảm bảo được về số liệu của server và sức mạnh vi xử lý theo ý người thiết lập.
- Có độ riêng tư cao hơn, vì files và databases sẽ được lưu trữ theo từng server nên sẽ không bị tìm thấy bởi server khác
- Dễ dàng nâng cấp.
- IP riêng. Chính vì điều này khả năng chống DDoS cũng cao hơn.

Nhược điểm

- Giá thành đòi hỏi phải rất cao.
- Cần ít nhiều kiến thức kỹ thuật để quản lý hệ thống cũng như thao tác từ xa với server.
- Cấu hình server không đúng có thể tạo ra lỗ hổng bảo mật.

1.2.4. Nginx

Để quản lý quá trình truy cập trang ứng dụng cũng như đưa trang ứng dụng dịch vụ của mình cho mọi người sử dụng mạng có thể tìm thấy và sử dụng trang web của mình thì công cụ hỗ trợ và cần phải được người cấu hình trang web cài đặt và sử dụng đó là Nginx hoặc là Apache và các công cụ này gọi chung là web server chúng giúp cho trang web của ta có thể được mọi người tìm kiếm hoặc truy cập thông qua các trình duyệt như goolge chrome hay cốc cốc, ...



Hình 1.36 : Nginx là gì

Nginx là được biết là một web server mã nguồn mở. Nginx sử dụng kiến trúc đơn luồng với tác dụng là tối đa hiệu năng và sự ổn định. Ngoài việc nginx có thể tận dụng được khả năng của máy chủ HTTP, Có thể dùng với giao thức HTTPS nếu có đăng kí khóa SSL còn có thể hoạt động như một proxy server cho email, reverse proxy, cân bằng tải cho server (load balancing), Lưu lại lịch sử thao tác của người dùng với trang web thông qua trình duyệt (HTTP caching). Nginx là một công cụ và thao tác với nginx là một kiến thức cần có đối với những lập trình viên liên quan đến web , quản trị hệ thống (system administrator) , Devops, và một số kiến thức khác về hệ điều hành mà server đang chạy.

Một số công ty lớn cũng sử dụng Nginx làm web server cho các trang ứng dụng của họ như LinkedIn, Facebook , Cisco , Twitter

Nginx giúp tối ưu hóa về năng suất cho trang web của người lập trình. Đối với các web server truyền thống sẽ hoạt động theo việc chia luồng cho yêu cầu cụ thể là cứ với mỗi yêu cầu cho trang web thì server sẽ sinh ra 1 luồng xử lý dành riêng cho yêu cầu đó nhưng khi ta tăng số lượng yêu cầu cho server lên rất nhiều thì sẽ sinh ra rất nhiều luồng và khi đó server sẽ dẫn đến việc là thiếu tài nguyên để đáp ứng cho một yêu cầu.

Còn riêng đối với nginx là web server sẽ hoạt động không dựa trên luồng mà Nginx được xây dựng trên kiến trúc xử lý bất đồng bộ và điều hướng tài nguyên cho các yêu cầu một cách linh hoạt.

Một số tiện ích của nginx như

- Cho phép xử lý nhiều kết nối cùng một lúc với các bộ nhớ thấp.

- Tối đa việc hoạt động với các tập tin tĩnh (Html, Css , JS , file ảnh) và lập ra các chỉ mục tập tin phù hợp .
- Có khả năng tăng tốc reverse proxy bằng cách bộ nhớ đệm giúp cân bằng tải đơn giản hơn với khả năng chịu lỗi vô cùng cao.
- Nginx có khả năng hỗ trợ thực hiện mã hóa theo mã SSL và TLS.
- Cấu hình của Nginx có thể tùy biến giúp lưu trữ lại nhật ký truy vấn vào trang ứng dụng một cách dễ dàng.
- Nginx có khả năng điều hướng với các mã lỗi từ 300 – 500 tùy mã mà trả về trang tương ứng cũng như thông tin lỗi
- Nginx có thể hạn chế của truy vấn vào trang ứng dụng nếu ta thiết lập.
- Nginx giúp giới hạn số kết nối đồng thời cũng như truy vấn từ 1 địa chỉ IP.

1.2.5. NodeJS (ExpressJS)

NodeJS được xem là nền tảng để viết và chạy các mã nguồn thuộc về ngôn ngữ JavaScript, Nó cho phép từ việc ngôn ngữ JavaScript từ một ngôn ngữ chỉ hoạt động trên các client web thành một thứ mà các lập trình viên có thể sử dụng nó để tạo ra các ứng dụng độc lập trong nhiều lĩnh vực như

- Ứng dụng trò chuyện trong thời gian thực
- Internet of Things (IoT)
- Truyền dữ liệu
- Xây dựng các SPA (Single-page application) phức tạp
- Các ứng dụng REST dựa trên API: JavaScript được sử dụng trong cả frontend lẫn backend của trang ứng dụng.

ExpressJS là một Framework, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển back-end cho web hoặc mobile

- Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng về việc thiếu thư viện trong khi làm việc với Framework này.

- Về performance: Express cung cấp thêm về các tính năng (feature) để lập trình viên viết mã nguồn tốt hơn mà không làm giảm tốc độ của NodeJS.

1.2.6. ORM (PRISMA)

SQL server hay Microsoft SQL Server, nó từ viết tắt của MSSQL Server. Được xem là một trình quản lý cơ sở dữ liệu được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ cho những dữ liệu dựa theo tiêu chuẩn RDBMS.

SQL Server có khả năng cung cấp đầy đủ các công cụ cho việc quản lý cơ sở dữ liệu từ giao diện GUI đến sử dụng ngôn ngữ cho việc truy vấn SQL. Điểm mạnh của SQL điểm mạnh của nó là có thể kết hợp và hoạt động với các nền tảng khác nhau và tương thích tốt với các ngôn ngữ viết API như Javascript , c#,

Để người viết code giao tiếp với SQL server sẽ được giao tiếp với nhau thông qua một ngôn ngữ gọi là SQL , T-SQL

- T-SQL là một trong những loại ngôn ngữ truy vấn được phát triển và thuộc quyền sở hữu của Microsoft và được gọi với cái tên Transact-SQL. Nó thường cung cấp thêm rất nhiều các cấu trúc câu lệnh giúp truy vấn dữ liệu,.....
- SQL Server Management Studio là một loại công cụ giao diện chính cho máy chủ cơ sở của chính dữ liệu SQL, thông thường thì nó hỗ trợ cho cả môi trường 64 bit và 32 bit.

Thì hiện nay trong thực tế có một số cách thực hiện tạo Database mà không cần thông qua GUI của SQL server mà dùng các đối tượng được định nghĩa và nhờ vào thư viện để ánh xạ các đối tượng đó vào trong cơ sở dữ liệu để hình thành các bảng và mối quan hệ của chúng thì kĩ thuật đó được gọi là ORM

Vậy ORM là từ viết tắt của (Object Relational Mapping), là một kỹ thuật/cơ chế lập trình thực hiện ánh xạ đối tượng vào cơ sở dữ liệu được thiết lập trong các ngôn ngữ lập trình hướng đối tượng như Java, C#, TypeScript,

Việc sử dụng ORM cho phép lập trình viên thao tác với database 1 cách hoàn toàn tự nhiên, dễ hiểu thông qua các đối tượng. Lập trình viên không cần tới loại database, kiểu dữ liệu trong database...

Ưu điểm của ORM:

- ORM giúp lập trình viên tập trung vào lập trình hướng đối tượng
- Tính độc lập: Làm việc được với nhiều loại database (hệ quản trị cơ sở dữ liệu), nhiều kiểu dữ liệu khác nhau.
- Đơn giản, dễ sử dụng
- Năng suất hơn
- Khả năng sử dụng lại code.

Nhược điểm của ORM:

- Khả năng truy vấn bị hạn chế, nhiều trường hợp ta vẫn phải dùng câu lệnh truy vấn SQL để lấy dữ liệu từ database.
- Khó tối ưu câu lệnh SQL (do câu lệnh SQL được ORM tự động sinh ra).

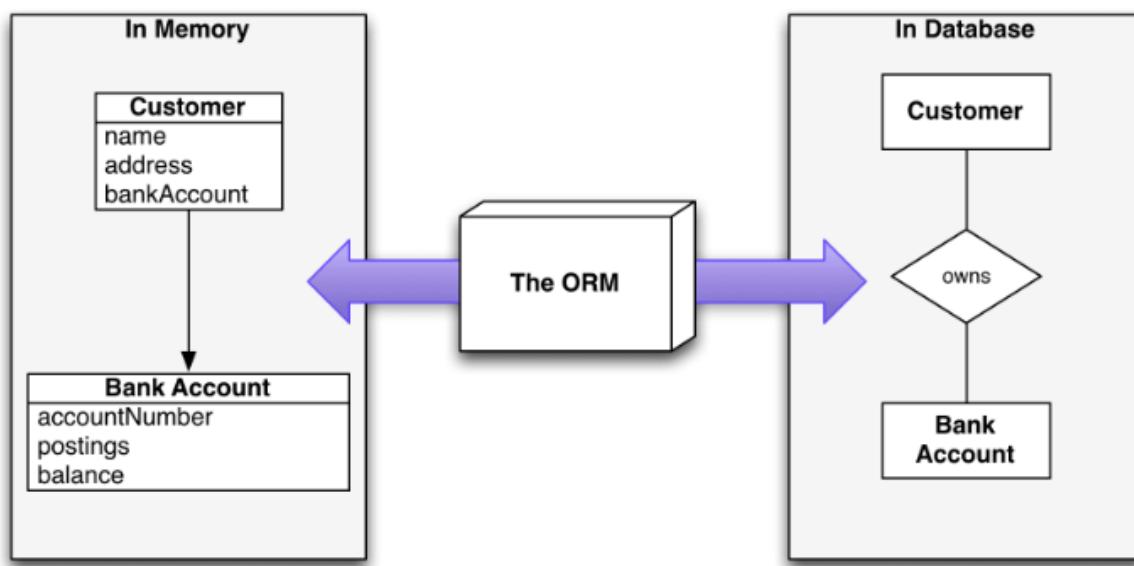
Một thư viện hỗ trợ việc ánh xạ dữ liệu từ đối tượng trong khi xây dựng back-end bằng NodeJS đó là thư viện Prisma . Thư viện giúp ánh xạ các đối tượng ta khai báo từ dạng model được quy định thành các bảng trong cở sở dữ liệu mà ta muốn thì ở đây chúng em chọn SQL Server

Prisma là một Object Relational Mapping (ORMs) được dùng để xây dựng các máy chủ như GraphQL Server, SQL Server, RESTful APIs, microservice, ... Prisma đơn giản là 1 layer nằm giữa Webserver và Database. Prisma giúp chúng ta giao tiếp với db một cách dễ dàng hơn.

Prisma bao gồm ba phần chính:

- Prisma Client : Trình tạo truy vấn an toàn và được tạo tự động cho Node.js và TypeScript.
- Prisma Migrate : Một hệ thống di chuyển và mô hình hóa dữ liệu khai báo.
- Prisma Studio : Một GUI để xem và chỉnh sửa dữ liệu trong cơ sở dữ liệu của bạn.

Cách truyền thông mà Webserver giao tiếp với Database là thông qua các câu lệnh SQL Query như SELECT, UPDATE hay DELETE. Giờ đây, nhờ vào các công cụ ORMs nói chung và Prisma nói riêng. Chúng tạo ra một tầng abstraction giữa Webserver và Database. Điều này giúp cho lập trình viên dễ dàng trong việc thao tác với Database. Thay vì viết những câu lệnh SQL khô khan, có thể sai bát cứ lúc nào thì chúng ta có thể viết các hàm tương ứng.



Hình 1.37: Ánh xạ đối tượng vào cơ sở dữ liệu

Trên đây là 2 bảng được khởi tạo trong database được khởi tạo thành 2 bảng là User_type và User_Account thì các giá trị bắt buộc không được bỏ trống thì sẽ không có dấu ? sau kiểu dữ liệu

Và cũng như khóa ngoại của bảng User_account sẽ là id trong bảng User_type

Thì với thư viện Prisma ta có thể ánh xạ các model thành các bảng trong cở sở dữ liệu và thư viện này có hỗ trợ một số loại cở sở dữ liệu phổ biến hiện nay như: PostgreSQL , MySQL, SQLite, MongoDB, SQL Server

Để tạo bảng trong các cở sở dữ liệu ta chỉ cần thực hiện các thao tác sau

- npx prisma studio : để mở GUI cho người viết code thấy được giao diện của prisma cùng với các bảng của mình

- npx prisma migrate : Để khởi tạo các bảng lưu lại quá trình thay đổi cũng như các file có dạng SQL ứng với mỗi migrate sẽ có 1 file SQL sinh ra và Prisma sẽ dựa vào tất cả các file SQL ở từng thư mục trong migrate để khởi tạo cơ sở dữ liệu cũng như là cập nhật các thuộc tính trong quá trình ta thao tác với các model
- Sau khi chạy câu lệnh này ta trong SQL Server ta khởi tạo và gán địa chỉ vào config để khi tạo từ prisma thì các bảng sẽ vào cơ sở dữ liệu đó sẽ hình thành các bản như các model mà ta đã khai báo trong Prisma ở NodeJS

1.2.7. ReactJS

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI, Nó được phân loại thành kiểu “V” trong mô hình MVC (Model-View-Controller). Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

React được sử dụng bởi hàng trăm công ty lớn trên thế giới, bao gồm Netflix, Airbnb, American Express, Facebook, WhatsApp, eBay, và Instagram.

Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code)

1.2.7.1. JSX

JSX (nói ngắn gọn cho JavaScript extension) là một cú pháp mở rộng cho JavaScript. Nó giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản.

JSX = Javascript + XML. Nó transform cú pháp dạng gần như XML về thành Javascript. Giúp người lập trình có thể code ReactJS bằng cú pháp của XML thay vì sử dụng Javascript. Các XML elements, attributes và children được chuyển đổi thành các đối số truyền vào React.createElement.

XML là viết tắt của từ eXtensible Markup Language, hay còn gọi là ngôn ngữ đánh dấu mở rộng do W3C đề nghị với mục đích tạo ra các ngôn ngữ đánh dấu khác. Đây là một tập hợp con đơn giản có thể mô tả nhiều loại dữ liệu khác nhau nên rất hữu ích trong việc chia sẻ dữ liệu giữa các hệ thống. Diễn hình nhất là ngôn ngữ đánh dấu siêu văn bản HTML sử dụng cú pháp của XML để tạo nên và nó có các bộ phần tử và thuộc tính không mềm dẻo nên chỉ có tác dụng trong việc trình bày dữ liệu trên trình duyệt Browser.

1.2.7.2. Tại sao lại nên dùng JSX

Việc sử dụng JSX trong ReactJS là không bắt buộc. Bạn có thể sử dụng chỉ JS thuần thôi. Thay vì tách biệt các công nghệ một cách giả tạo bằng cách đưa định nghĩa giao diện và logic vào những tệp khác nhau, React tách bạch mối quan hệ bằng những đơn vị rời rạc gọi là “components” chứa cả hai cái trên. Nhưng có rất nhiều lý do cho việc nên sử dụng JSX trong ReactJS đây.

- Thứ nhất, JSX với cú pháp gần giống XML, cấu trúc cây khi biểu thị các attributes, điều đó giúp ta dễ dàng định nghĩa, quản lý được các component phức tạp, thay vì việc phải định nghĩa và gọi ra nhiều hàm hoặc object trong javascript. Khi nhìn vào cấu trúc đó cũng dễ dàng đọc hiểu được ý nghĩa của các component. Code JSX ngắn hơn, dễ hiểu hơn code JS.
- Thứ 2, JSX không làm thay đổi ngữ nghĩa của Javascript
- Thứ 3, với cách viết gần với các thẻ HTML, nó giúp những developers thông thường (ví dụ như các designer) có thể hiểu được một cách dễ dàng, từ đó có thể viết hoặc sửa code mà không gặp nhiều khó khăn.

1.2.7.3. Components

Components cho phép bạn chia UI thành các phần độc lập, có thể tái sử dụng, và hoàn toàn tách biệt nhau. Về mặt khái niệm, components cũng giống như các hàm Javascript. Chúng nhận vào bất kì đầu vào nào (còn được gọi là “props”) và trả về các React elements mô tả những gì sẽ xuất hiện trên màn hình.

Có 2 kiểu components 1 là function components 2 là class components nhưng chúng ta chỉ bàn về function components ở đây.

Hàm này là một React component hợp lệ về nó nhận đầu vào là một tham số object “props” (viết tắt của properties) với dữ liệu và trả về một React element. Chúng ta gọi các components này là “function components” vì chúng là các hàm JavaScript theo đúng nghĩa đen.

Chú ý: Luôn luôn bắt đầu tên của component bằng chữ in hoa.

React xử lí các components bắt đầu với chữ thường giống như các DOM tags. Ví dụ, `<div />` biểu diễn HTML div tag, nhưng `<Welcome />` biểu diễn một component và yêu cầu Welcome nằm trong scope.

Các components có thể tham chiếu đến các components khác tại đầu ra của chúng. Điều này cho phép chúng ta sử dụng cùng một component abstraction cho mọi mức độ chi tiết. Một button, form, dialog, màn hình: trong React apps, chúng đều được hiển thị như là các components.

Thông thường, các React apps mới tạo sẽ có một App component ở tầng cao nhất. Thế nhưng, nếu bạn tích hợp React vào ứng dụng hiện có, bạn có thể bắt đầu bằng cách tiếp cận bottom-up với một component nhỏ như là Button và dần dần đi lên các tầng trên cùng của cây kế thừa giao diện.

Đừng ngại ngần việc tách components thành các components nhỏ hơn. Chia các components ngay từ đầu là một công việc không đơn giản, nhưng bù lại chúng ta sẽ có được một tập hợp các components có thể tái sử dụng trong các ứng dụng lớn hơn khác. Một nguyên tắc quan trọng đó là nếu một phần UI của bạn được sử dụng lại nhiều lần

1.2.7.4. Lifecycle

Lifecycle là vòng đời của 1 component react. Chúng ta có thể tương tác lifecycle của 1 component thông qua các giai đoạn của lifecycle:

Có 3 giai đoạn chính **Mounting, Updating, và Unmounting.**
Mounting: Đây là giai đoạn đầu tiên của 1 component đó là khi component được khởi tạo

Updating: Đây là giai đoạn sau khi component được khởi tạo. mà được cập nhật mỗi khi state hoặc props thay đổi

Unmounting: Đây là giai đoạn khi component được xóa khỏi DOM

1.2.7.5. Props và State

Props là 1 từ viết ngắn gọn của properties. props là 1 đối tượng, nó lưu trữ các giá trị của các attribute (thuộc tính) của một thẻ (Tag). Là cách mà component có thể nhận được các giá trị của thuộc tính truyền vào từ bên ngoài vào, và là cách mà các component có thể nói chuyện với nhau.

Dữ liệu của của React là một chiều có nghĩa là nó chỉ đi từ component cha xuống component con

State có chức năng tương tự như props cũng lưu trữ thông tin về component, nhưng là lưu trữ dữ liệu động của một component.

State là dữ liệu động , nó cho phép một component theo dõi thông tin thay đổi giữa các kết xuất (render) và làm cho nó có thể tương tác.

State chỉ có thể được sử dụng ở trong một component sinh ra nó.

Nếu dự đoán được một component cần quản lý state, thì nó nên được tạo ở trong một class component chứ không phải là một function component. Vì function components là stateless component có nghĩa là nó không có state để lưu trữ thông tin, thay vào đó nó sử dụng các hook để lưu giữ thông tin.

1.2.7.6. Hook

Hooks chính thức được giới thiệu trong phiên bản React 16.8. Nó cho phép chúng ta sử dụng state và các tính năng khác của React mà không phải dùng đến Class

Điều này có nghĩa là từ phiên bản 16.8 trở đi, chúng ta đã có thể sử dụng state trong stateless (functional) component

Hook là 1 function có thể gắn (hook) vào 1 function component để giúp component có thể quản lý các state hay các tính năng vòng đời của 1 component

Hook không hoạt động được trong class component

Hàm useState nhận đầu vào là giá trị khởi tạo của 1 state và trả ra 1 mảng gồm có 2 phần tử, phần tử đầu tiên là state hiện tại, phần tử thứ 2 là 1 function dùng để update state.

Hàm useEffect dùng để xử lý các side effects. Nó thay thế các hàm cũ trong lifecycle.

Hàm useEffect nhận đầu vào gồm 2 tham số:

Tham số 1 là 1 hàm sẽ được chạy khi các giá trị trong chuỗi ở hàm thứ 2 thay đổi. trong hàm này nếu bạn trả về 1 hàm thì hàm trả về này sẽ được chạy khi component được update

Trong đó mỗi khi props.source thay đổi giá trị thì hàm ở tham số 1 sẽ chạy lại và trước khi hàm ở tham số 1 sẽ chạy lại hàm return sẽ chạy để dọn dẹp những dữ liệu của hàm ở tham số 1 để tránh tồn bộ nhớ

useMemo giúp ta kiểm soát việc được render dư thừa của các component con bằng cách truyền vào 1 tham số thứ 2 thì chỉ khi tham số này thay đổi thì thằng useMemo mới được thực thi.

Trong khi đó ở component UsingMemo soldoutProducts sẽ chỉ được tính toán lại nếu như products (tham số truyền vào ở tham số thứ 2 của hàm useMemo) thay đổi. Nó giúp ta loại bỏ được các trường hợp đầu vào không thay đổi, giúp giảm tải việc tính toán

useCallback có nhiệm vụ tương tự như useMemo nhưng khác ở chỗ function truyền vào useMemo bắt buộc phải ở trong quá trình render trong khi đối với useCallback đó lại là function callback của 1 event nào đó như là onClick chẳng hạn.

useContext sẽ giải quyết vấn đề là việc trên dòng dữ liệu trong react chỉ đi 1 chiều từ cha xuống con vậy làm sao để truyền props từ con lên cha hoặc giữa các component con với nhau. Bạn có thể tạo một context để thêm những shared state và sau đó component nào cần thì bạn gọi context đó ra xài, không cần phải truyền từ cha xuống con nữa. Nó cũng giống như biến global được chia sẻ và sử dụng ở nhiều nơi, nhưng context được quản lý tốt hơn để maintain code dễ hơn.

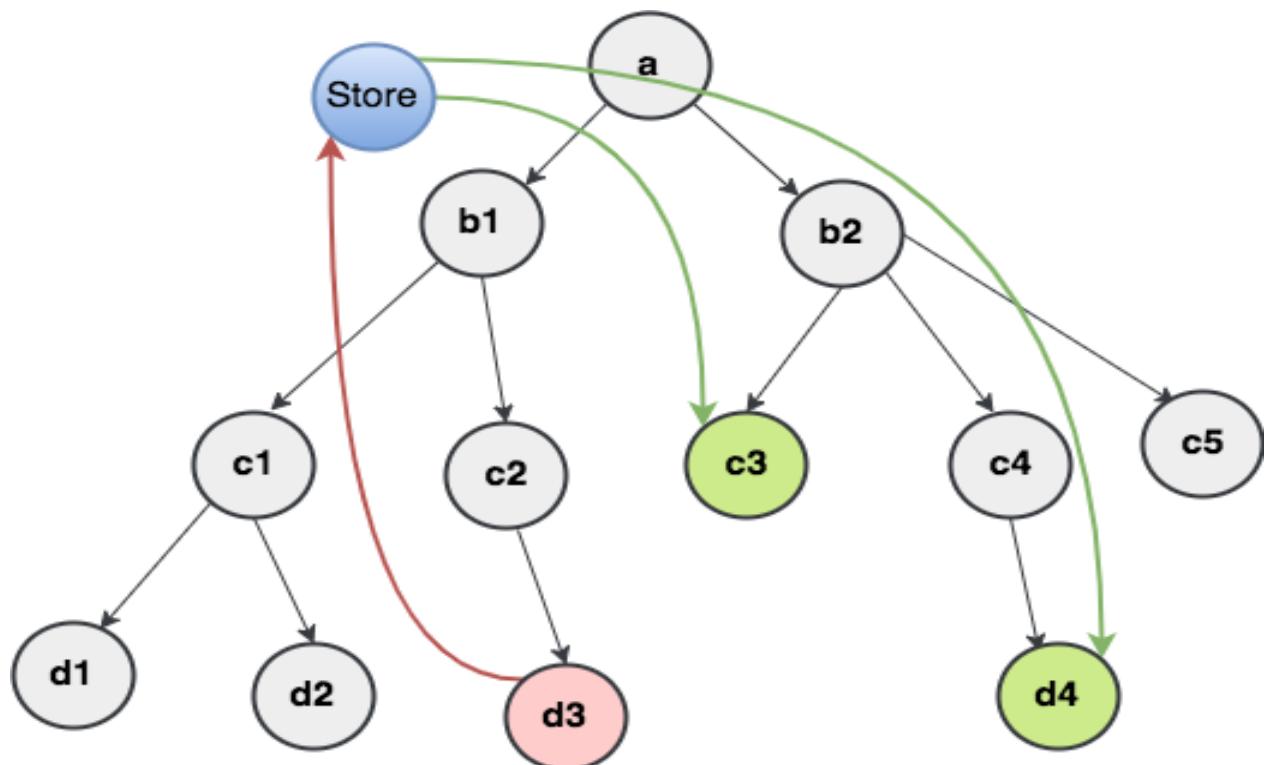
Ngoài ra chúng ta còn rất nhiều cái hook khác được định nghĩa sẵn bởi React mà các bạn có thể đọc thêm ở trên trang chủ của React hoặc chúng ta có thể tự viết hook cho mình trong trường hợp cái logic đó lặp lại ở các component khác nhau. Các hàm hook luôn được bắt đầu với use

Trong ví dụ trên ta tạo sử dụng hook useFetch để lấy dữ liệu từ data. Hàm useFetch nhận vào url và trả về data và sẽ gọi api mỗi khi url thay đổi (ứng dụng useEffect đã giới thiệu ở trên). Sau đó chúng ta có thể sử dụng useFetch như các hook khác

1.2.8. Redux

Redux là 1 thư viện Javascript để quản lý state của ứng dụng, thường được sử dụng với javascript framework như React.

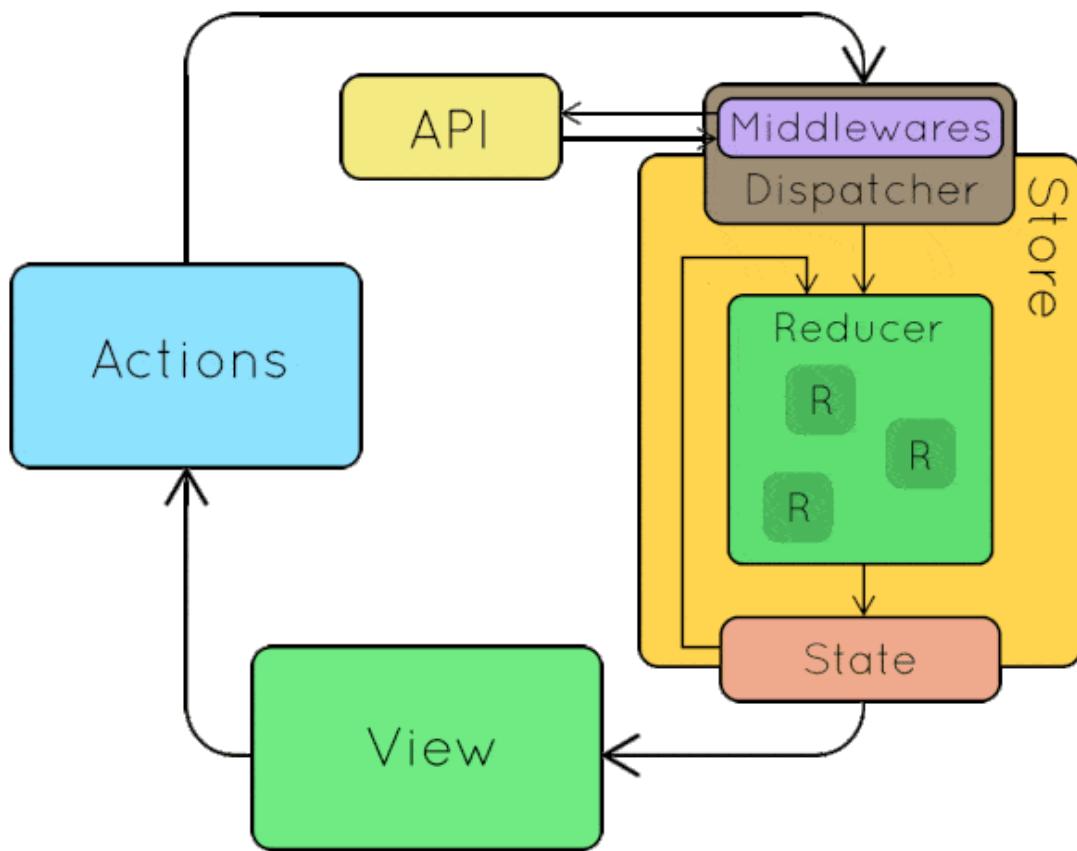
Như đã đề cập bên trên dữ liệu của react chỉ truyền 1 chiều từ cha cho con. Việc truyền dữ liệu giữa con lên cha hoặc giữ các component đồng cấp rất phức tạp. Chúng ta có thể dùng useContext để cập ở trên tuy nhiên useContext sẽ render lại tất cả các component nằm trong provider khi value thay đổi điều này ôn với các ứng dụng nhỏ tuy nhiên với các ứng dụng lớn việc này sẽ giảm hiệu xuất của trang web. Do đó redux là 1 giải pháp quản lý state được nhiều người sử dụng khi sử dụng React. Trái với useContext, khi sử dụng Redux các component chỉ render lại những component thay đổi props hoặc state tuy nhiên việc setup Redux đòi hỏi nhiều công sức hơn là useContext.



Hình 1.38: Mô hình hoạt động của Redux

Cách mà Redux hoạt động là khá đơn giản. Nó có 1 store lưu trữ toàn bộ state của app. Mỗi component có thể access trực tiếp đến state được lưu trữ thay vì phải truyền props từ component này đến component khác.

Cơ chế hoạt động của nó được tóm gọn trong 1 sơ đồ đơn giản:



Hình 1.39: Sơ đồ hoạt động của Redux

Khi người dùng tương tác với UI sẽ tạo ra 1 action action này sẽ được dispatcher gửi đến reducer (các action bắt đồng bộ ví dụ như gọi api sẽ được xử lý bởi các middleware tại bước này). Reducer sẽ nhận thông tin từ action và state hiện tại và xử lý sao đó trả về một state mới. Khi state thay đổi UI cũng sẽ cập nhật theo.

Thành phần chính của Redux gồm: Actions, Store, Reducers.

1.2.8.1. Action

Actions đơn giản là các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.

Actions được gửi bằng cách sử dụng store.dispatch() method, chúng phải có một type property để biểu lộ loại action để thực hiện. Chúng cũng phải có một payload chứa thông tin. Actions được tạo thông qua một action creator.

Trong đó setLoginStatus là tên của action (function) sẽ được gọi trong hàm dispatch. setLoginStatus trả về 1 object gồm type để reducer nhận biết phải thực hiện hành động nào và payload để truyền data xuống reducer

1.2.8.2. Reducers

Reducers là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store. 1 reducer tương đương với 1 state nhưng kèm theo các mô tả state sẽ thay đổi như thế nào khi các action khác nhau được gọi.

Mỗi reducer có 1 tên khác nhau. Reducer nhận vào 2 tham số là state với giá trị default là initialState (giá trị khởi tạo của state) và 1 action

Reducer sẽ dựa theo action type để thay đổi state với payload (nếu có) và trả về state mới. Lưu ý state mới phải là object mới nếu không Redux sẽ không thể cập nhật lại UI

1.2.8.3. Store

Store là nơi lưu trạng thái của ứng dụng và chỉ có duy nhất một Store trong bất kỳ một chương trình ứng dụng Redux nào. Nhiệm vụ của store chính là quản lý, access các state được lưu, update state thông qua dispatch, cho phép truy cập state thông qua Getstate và đăng ký hoặc hủy đăng ký các listeners thông qua helper methods.

1.2.8.4. Sử dụng Redux

Sử dụng useSelector của react-redux để lấy state counter từ store.

Sử dụng useDispatch để trả về function dispatch, truyền increment và decrement vào dispatch để gọi 2 action này

1.2.8.5. Redux-toolkit

Việc setup redux khá phức tạp và tốn thời gian. Redux-toolkit là một package được sinh ra nhằm giải quyết phần lớn những vấn đề kể trên, được phát triển bởi chính

chủ reduxjs team, giúp chúng ta viết code redux nhanh gọn, hoàn chỉnh theo một quy chuẩn thống nhất.

Hàm configureStore là 1 hàm của redux-toolkit giúp chúng ta đơn giản việc setup store. Như ví dụ ở trên, chúng ta khởi tạo một store bằng hàm createStore của redux với tham số nhận vào là một reducer, configureStore sẽ làm điều tương tự như vậy, chúng ta sẽ khởi tạo store theo phương pháp sau

Về cú pháp thì không khác nhau nhiều. Tuy nhiên thay vì chỉ khởi tạo một store đơn thuần, configureStore sẽ mặc định thiết lập cho phép sử dụng redux devtool để debug và theo dõi quá trình cập nhật state cũng như thiết lập sẵn một số middleware giúp thực hiện các action bất đồng bộ.

createSlice là hàm của redux-toolkit giúp chúng ta định nghĩa 1 reducer và các action liên quan đến nó

createSlice nhận vào 1 object gồm:

- name: Tên của reducer
- initialState: state khởi tạo của reducer
- reducer: là 1 object định nghĩa các hàm quy định reducer sẽ cập nhật state như thế nào. Bên cạnh đó như đã nhắc ở trên reducer phải trả về 1 object mới tuy nhiên redux-toolkit đã đơn giản hóa việc đó chúng ta có thể thực hiện tính toán ngay trên object cũ

createSlice trả về 1 object chứa actions và reducer. Chúng ta có thể export các action và reducer để sử dụng. Như vậy chỉ 1 hàm đơn giản đã giúp chúng ta tiết kiệm rất nhiều thời gian bằng cách gộp bước tạo action và tạo reducer vào làm 1 .

1.2.9. *Typecript*

TypeScript là một phiên bản cao hơn của JavaScript, được thiết kế để xây dựng các ứng dụng lớn và phức tạp. Nó kế thừa nhiều khái niệm từ Java và C#, TypeScript là ngôn ngữ tĩnh (Static typed) có nghĩa là nó nghiêm ngặt và có trật tự trái ngược với free-type. Nó còn được bổ sung thêm lớp hướng đối tượng mà điều này không có ở Javascript. Với TypeScript, ta có thể bê nguyên xi code JavaScript vào trong cùng một file và chạy cùng

nhau bình thường, bởi vì TypeScript duy trì cú pháp của JavaScript và mở rộng nó bằng một loạt tính năng mới. Nhờ đó mà hiệu năng làm việc được tăng lên đáng kể.

1.2.9.1. Ưu điểm của Typescript:

Dễ dàng hơn trong phát triển các dự án lớn, được hỗ trợ bởi các Javascript Framework lớn.

Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override...v.v

Cách tổ chức code rõ ràng hơn, hỗ trợ cơ chế giúp kiến trúc hệ thống code hướng module, hỗ trợ namespace, giúp xây dựng các hệ thống lớn nơi mà nhiều lập trình viên có thể làm việc cùng nhau một cách dễ dàng hơn.

Hỗ trợ các tính năng mới nhất của Javascript. TypeScript luôn đảm bảo việc sử dụng đầy đủ các kỹ thuật mới nhất của Javascript, ví dụ như version hiện tại là ECMAScript 2015 (ES6).

Một lợi thế của Typescript nữa là mã nguồn mở vì vậy nó miễn phí và có cộng đồng hỗ trợ rất lớn.

Với static typing, code viết bằng TypeScript dễ dự đoán hơn, và dễ debug hơn.

1.2.9.2. Basic Types:

Trong TypeScript chia ra làm 7 loại cơ bản, bao gồm: boolean, number, string, array, enum, any, void.

- Boolean:
- String:
- Number:
- Array : có 2 kiểu khai báo tương đương với nhau trong TypeScript
- Enum: khi khai báo enum một cách thông thường các phần tử sẽ được đánh số từ 0 và tăng dần
- Khi muốn phần tử đầu tiên là 1 chữ không phải là 0 như mặc định thì cần khai báo như sau:

- Any: Any là một kiểu mà bạn có thể gán bất kỳ kiểu nào cho nó.
- Void: Cũng giống như any nhưng void được sử dụng là đầu ra của hàm.

1.2.9.3. Function

TypeScript có khai báo function giống như JavaScript. Nhưng khi khai báo function TypeScript còn hỗ trợ việc khai báo với các kiểu trả ra của function và cũng như kiểu đầu vào của dữ liệu. Không những thế khi sử dụng TypeScript ta có thể khai báo giá trị mặc định của đầu vào ngay khi khai báo function, điều mà JavaScript không có. Không dừng lại ở đó TypeScript còn hỗ trợ việc bỏ qua nhập một hoặc vài đầu vào.

1.2.9.4. Type aliases

Trong TypeScript, có rất nhiều types cơ bản như là number, string,... Ngoài ra, trong TypeScript chúng ta có các types nâng cao và các types nâng cao này được gọi là type aliases. Với type aliases, chúng ta có thể tạo tên mới cho một type nhưng chúng ta không define một type mới. Từ khóa type trong TypeScript là một cách cung cấp type aliases cho các variables, objects và functions của chúng ta. Các type aliases này mô tả dữ liệu của chúng ta trông như thế nào. Để mô tả loại dữ liệu của mình trông như thế nào chúng ta sử dụng các type cơ bản (string, number,...) hoặc bằng cách tạo tạo các type tùy chỉnh của chúng ta.

1.2.9.5. Interface

Interface trong typescript cho phép bạn định nghĩa thuộc tính là gì và phương thức là gì mà đối tượng cần để được implement. Nếu đối tượng tuân thủ đúng khuôn mẫu interface thì đối tượng đã implement interface ấy sẽ được thực thi đúng. Nếu interface không được thực thi đúng thì typescript sẽ phát sinh lỗi ngay lập tức.

1.2.9.6. Types vs Interfaces

Có thể merge interfaces, types thì không

Type aliases có thể sử dụng computed properties:

- Từ khóa in có thể được sử dụng để iterate tất cả các item bên trong một tập hợp keys. Chúng ta có thể sử dụng tính năng này để tạo mapped types.

Ví dụ sử dụng type aliases

Extend và implements:

Trong TypeScript, chúng ta dễ dàng extends và implements interfaces. Nhưng không thể với types. Với interface để kế thừa chúng ta sử dụng extends:

1.2.9.7. Intersection

Intersection cho phép chúng ta kết hợp nhiều types thành một type duy nhất. Để tạo một intersection type chúng ta dùng &:

1.2.9.8. Generics

Generic type trong TypeScript cho phép bạn viết các function, class và interface có thể tái sử dụng và tổng quát hóa. Trong hướng dẫn này, bạn sẽ tập trung vào việc tạo các hàm generic (generic function).

TypeScript cho phép bạn khai báo một tham số có kiểu dữ liệu bị ràng buộc bởi kiểu dữ liệu của một tham số khác.

1.2.10. Antd

1.2.10.1. Antd là gì

Ant là tập hợp các components của React được xây dựng theo chuẩn thiết kế của Ant UED Team. Tương tự như chuẩn Material Design, Ant cung cấp hầu hết các component thông dụng trong ứng dụng web hiện đại, như Layout, Button, Icon, DatePicket, v.v... Bên cạnh đó Ant cũng có những component riêng thú vị, như LocaleProvider cho phép bạn thay đổi ngôn ngữ trên toàn ứng dụng.

Có thể coi Ant Design cho React là tập hợp của hầu hết các thư viện về React. Nó đáp ứng được hầu hết các yêu cầu của project của bạn mà bạn không phải cài thêm bất cứ thư viện nào nữa. Dưới đây là danh sách các component mà nó cung cấp:

- General: Button, Icon
- Layout: Grid, Layout
- Navigation: Affix, Breadcrumb, Dropdown, Menu, Pagination, Steps

- Data Entry: AutoComplete, Checkbox, Cascader, DatePicker, Form, InputNumber, Input, Mention, Rate, Radio, Switch, Slider, Select, TreeSelect, Transfer, TimePicker, Upload
- Data Display: Avatar, Badge, Collapse, Carousel, Card, Calendar, List, Popover, Tree, Tooltip, Timeline, Tag, Tabs, Table
- Feedback: Alert, Drawer, Modal, Message, Notification, Progress, Popconfirm, Spin, Skeleton
- Other: Anchor, BackTop, Divider, LocaleProvider

1.3.Nội dung của chương này

Giới thiệu về đề tài về xây dựng trang ứng dụng đăng tin tuyển dụng cũng như ứng tuyển dành cho sinh viên trường Đại học Tôn Đức Thắng

- Mục đích của đề
- Xây dựng nên sơ đồ Class, sơ đồ Activity, Sơ đồ UseCase
- Mô tả của từng useCase các tính năng trong trang ứng dụng

Giới thiệu về khái niệm của các công nghệ sử dụng trong quá trình xây dựng trang ứng dụng này như

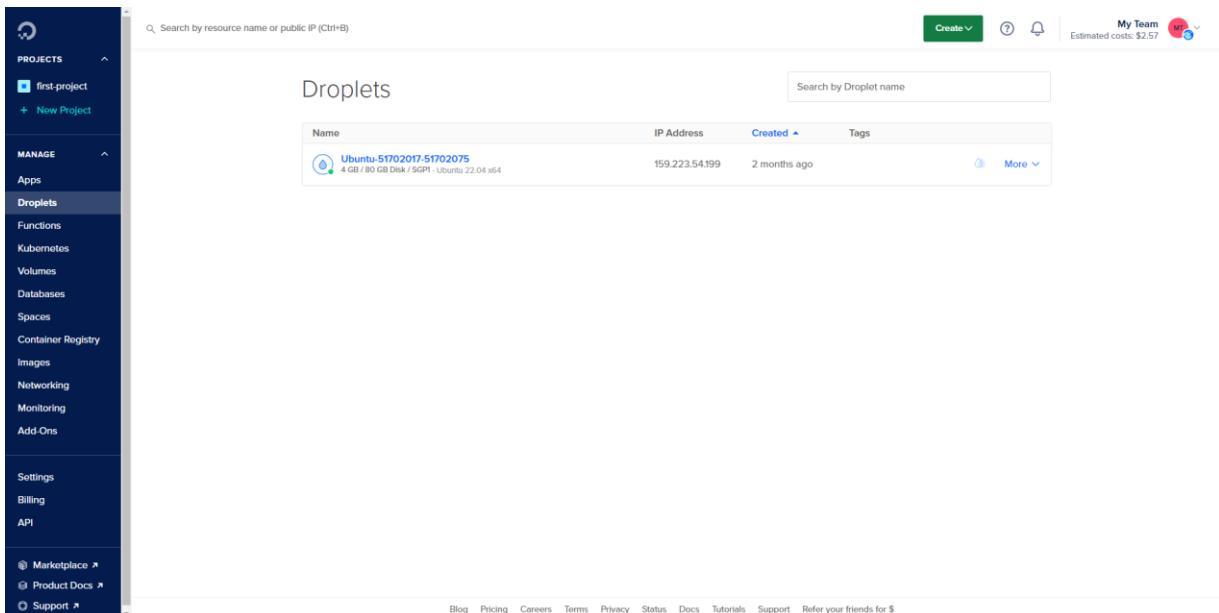
- Server : Docker , VPS , Nginx, CI/CD
- Back-End : NodeJS , Prisma , SQL Server
- Front-End : React , TypeScript, Antd, Redux

CHƯƠNG 2 – QUÁ TRÌNH THIẾT LẬP CÁC CÔNG CỤ

2.1. Quá trình thiết lập server

Để đưa source code lên server ta cần một server ở đây chúng em thuê 1 con server chạy hệ điều hành Ubuntu với 4GB ram và 80GB bộ nhớ của Digitalocean với chi phí duy trì hàng tháng

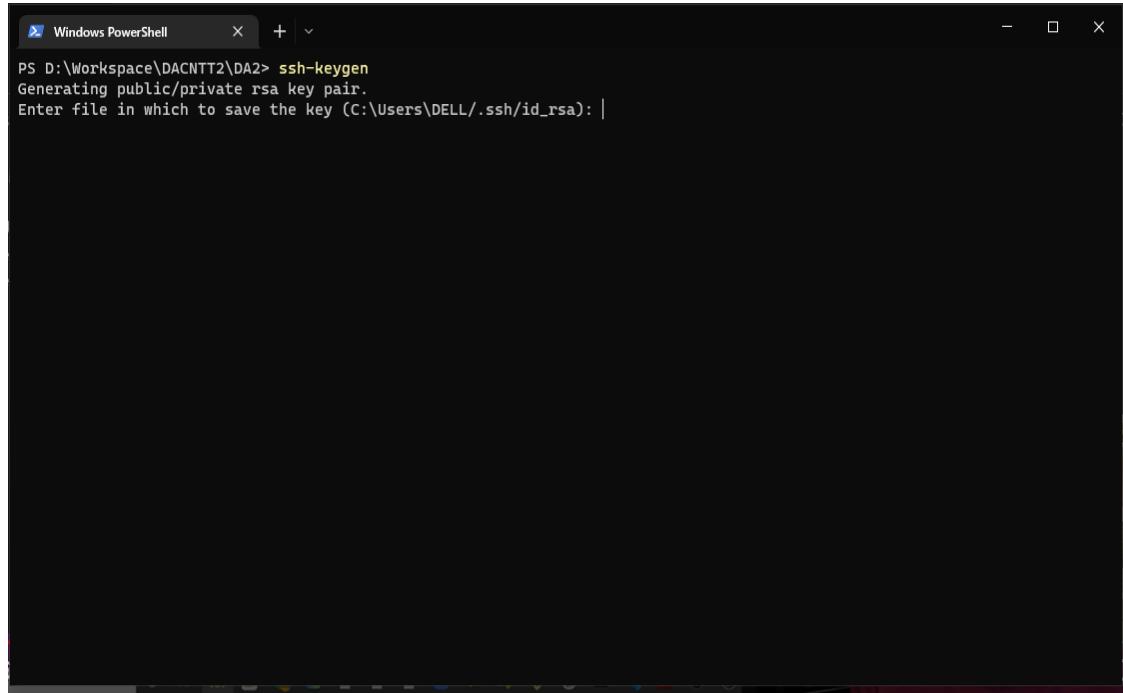
Quá trình thuê thì sẽ đưa thẻ visa của bản thân vào tài khoản để thuê server và mỗi tháng thì digitalocen sẽ trừ khoản tiền đó



Hình 2.1: Hình ảnh thuê server

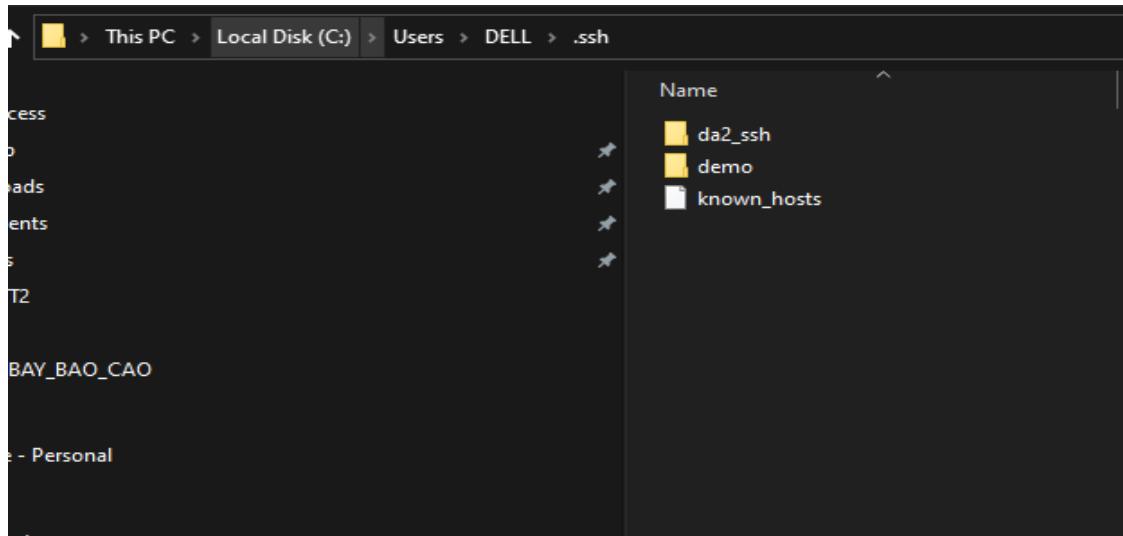
Khi đã có server thì bước tiếp theo thì sẽ cấu hình cho việc kết nối với server đó qua một số công cụ như cmd , Window Terminal,, và các bước như sau:

Bước 1 : ta sẽ sử dụng ssh-keygen để tạo ra 1 chuỗi các ký tự ngẫu nhiên được gọi là key để sau khi thiết lập mỗi lần muốn kết nối với server ta chỉ cần gõ lệnh ssh@root + tên server và nhập mật khẩu của server thì lúc này ta sẽ kết nối được với server của mình vừa thuê



Hình 2.2: Gõ ssh-keygen để tạo ra bộ mã cho riêng mình

Bước 2 ta vào thư mục C:/Users/(Tên của user)/.ssh và tạo ra 1 thư mục ví dụ ở đây máy cá nhân này có thư mục (demo) để mô phỏng lại quá trình tạo ssh-keygen và da2_ssh đã tạo trong quá trình thực hiện đề tài này

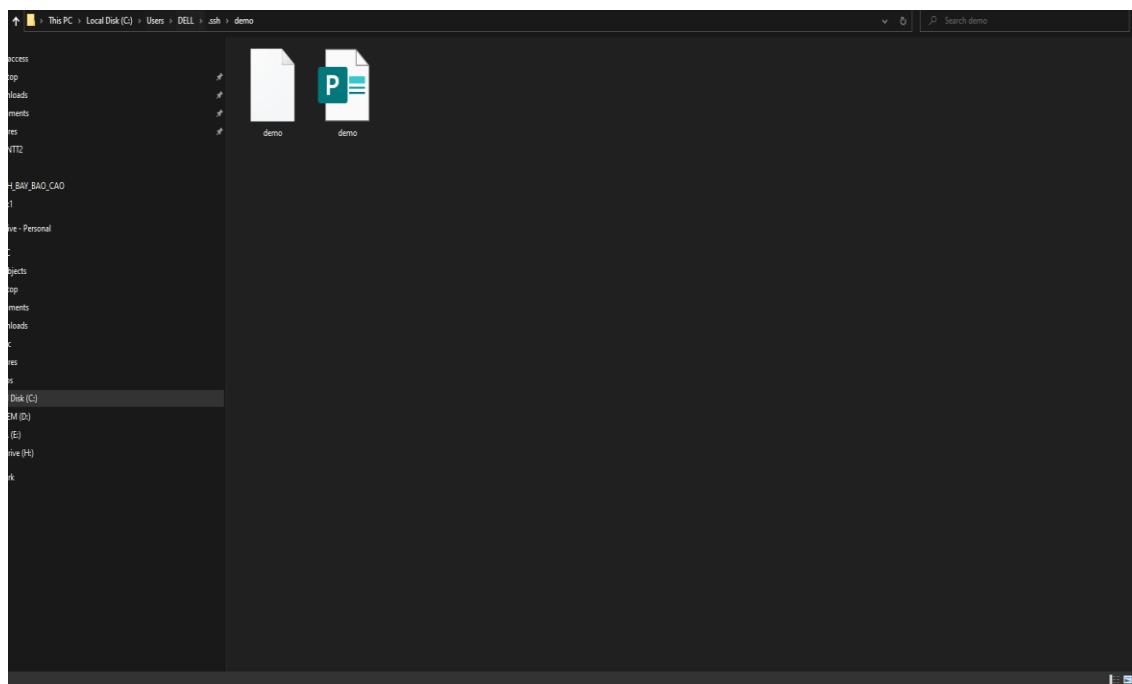


Hình 2.3: Thư mục .ssh chứa ssh-keygen

Tiếp tục ta ở Window Terminal ta gõ lệnh để với thư mục demo ta gõ lệnh chuyển trỏ nơi tạo file chưa public key và private key

```
PS D:\Workspace\DACNTT2\DA2> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:/Users/DELL/.ssh/id_rsa): C:/Users/DELL/.ssh/demo/demo
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:/Users/DELL/.ssh/demo/demo.
Your public key has been saved in C:/Users/DELL/.ssh/demo/demo.pub.
The key fingerprint is:
SHA256:p44unkCBt9axRF455Ni5r9uV7yuc2mfDrD2YyZC6b6Q dell@DESKTOP-TDOQ9QP
The key's randomart image is:
+---[RSA 3072]---+
| . |
| = o |
| o B |
| o . oS.. |
| o o + . +o . |
| + * + =.+oB |
| . o * E+ooOoB |
| . o**=.o*=+ |
+---[SHA256]---+
PS D:\Workspace\DACNTT2\DA2> |
```

Hình 2.4: Hình ảnh tạo ssh key trong thư mục demo



Hình 2.5: Hình ảnh thư mục demo chứa ssh key ta vừa tạo với demo

- Trong thư mục ta vừa tạo thì sẽ có 2 tập tin bao gồm 1 file demo thường và 1 file demo có đuôi là pub tức nghĩa là public key còn file còn lại là secret key
- Sau khi khởi tạo ssh key ở máy cá nhân ta truy cập với server thông qua ssh bằng dòng lệnh

```

PS D:\Workspace\DACNTT2\DA2> ssh root@159.223.54.199
root@159.223.54.199's password:
Permission denied, please try again.
root@159.223.54.199's password:
Permission denied, please try again.
root@159.223.54.199's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Dec  4 08:36:48 UTC 2022

System load:          0.0
Usage of /:           13.6% of 77.35GB
Memory usage:         54%
Swap usage:           0%
Processes:            130
Users logged in:     0
IPv4 address for br-dae8ae5879b: 172.19.0.1
IPv4 address for docker0:        172.17.0.1
IPv4 address for eth0:          159.223.54.199
IPv4 address for eth0:          10.15.0.5
IPv4 address for eth1:          10.104.0.2

22 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Thu Dec  1 12:58:36 2022 from 14.169.46.238
root@Ubuntu-51702017-51702075:~#

```

Hình 2.6: Kết nối với server thuê được bằng ssh

- Ta sẽ sử dụng ssh root@(IP) mà nhà cung cấp dịch vụ vps cung cấp cho ta để kết nối và nhập mật khẩu của server để truy cập vào server
- Khi này ta sẽ có quyền thao tác với server lúc này ta sẽ add ssh được sinh ra ở máy cá nhân

```

*** System restart required ***
Last login: Thu Dec  1 12:58:36 2022 from 14.169.46.238
root@Ubuntu-51702017-51702075:~# cd .ssh
root@Ubuntu-51702017-51702075:~/ssh# ls
authorized_keys  known_hosts
root@Ubuntu-51702017-51702075:~/ssh# vi authorized_keys
root@Ubuntu-51702017-51702075:~/ssh# |

```

Hình 2.7: Thêm ssh key vào máy chủ

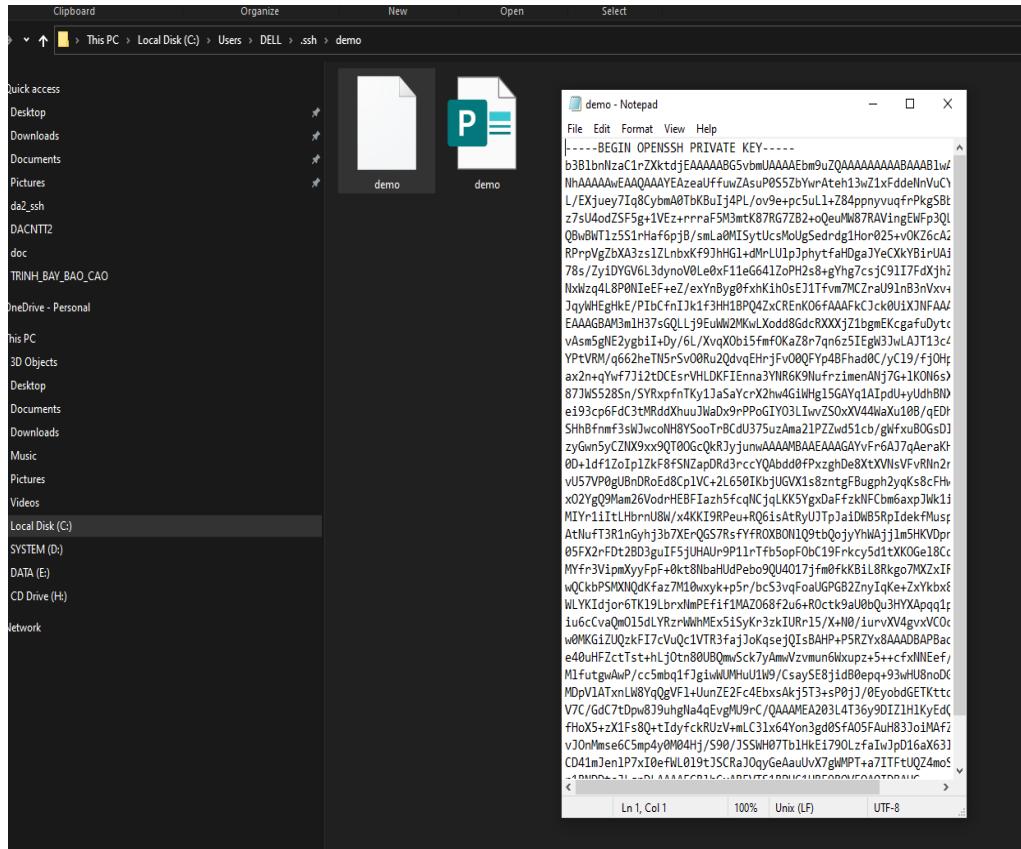
- Khi này ta thực hiện chuyển thư mục bằng `cd .ssh` và `ls` để check xem đã có file `authorized_keys` hay chưa nên chia sẽ tạo file này bằng sử dụng câu lệnh vừa `cd` vừa tạo bằng `sudo nano authorized_keys` để chuyển đến thư mục `.ssh` và có quyền như root sử dụng lệnh viết `nano` trên tập tin `authorized_keys`
- Ở đây máy chủ đã có tập tin `authorized_keys` nên em sẽ dùng `vim (vi)` để xem nội dung thư mục

```
root@Ubuntu-51702017-51702 ~ + 
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDANJ2MoREOZi385axdpqbaVH4cpCyCptDoIu3/xh01ejv6JhIicqSp1bSS3NQ
zRfR/qS8BuVpSdQ53x7jTS8bRNvT01dM2PSob06ygh9bf3keMk9Unt6DtW31WLMDoLfhBbyIUjk1kCXJj57y83wkvud6Qg2wLBMTB
LNutJHXU+gDzKgM+nKLjAXkRyRGMhsdika9A0lAu2qJUDLcj3ve+7goVj9QI3NyqRtJMIs7HaaB5AgN9ZNvv85wfnnqs8q4mtFenk
Jjcr0Vc8XjoWoEraHht5GQMmvp5cQs1mb5j6tLsAb9+gBs97tpiUsECHOp7TLnz6LkzhQPY7obNbl dell@DESKTOP-TDOQQQP ]
```

"authorized_keys" 2L, 403B 1,1 All

Hình 2.8: Nội dung tập tin `authorized_keys`

- Nếu là lần đầu thiết lập tập tin này không có gì và ta sẽ thực hiện thao tác nhân nút I trên bàn phím để vào chế độ viết và ta quay lại máy cá nhân thư mục .ssh ta tạo có 2 file public và file thường



Hình 2.9: Ta sẽ copy key trong file private key

Ta sẽ past private key này vào file ở server thì đây là demo sẽ có nội dung khác với key đã được thêm vào server trước đó

Sau khi thêm private của máy cá nhân vào server thì ta nhấn :x để thoát và lưu lại nội dung đã ghi với vim

Sau khi đã lưu lại private key ta tiến hành update và cài đặt một số thư viện với các dòng lệnh như sau:

- Sudo apt update : Để Ubuntu sẽ tiến hành việc tải các tài nguyên cần thiết để cấu hình server

- Sudo apt upgrade : Để nâng cấp phiên bản nếu tài nguyên sau khi update còn quá cũ
- Sudo apt-get install ca-certificates : Cài đặt tài nguyên này để cho việc chứng thực số cho server này có thể hoạt động trong môi trường mạng
- Sudo apt-get install curl : Để cài đặt 1 công cụ (tool) có thể cài đặt một số thư viện khác ở đây để tải 1 số thư viện
- Sudo apt-get install gnupg : Để cài đặt một phần mềm bảo mật cho server
- Sudo apt-get install lsd-realase : Để cài đặt phiên bản linux tiêu chuẩn nhất giành cho máy chủ nếu phiên bản linux của máy chủ quá cũ
- Ta sử dụng 1 câu lệnh như sau :

Curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor - /usr/share/keyrings/docker-archive-keyring.gpg Để thêm khóa mã hóa cho docker (GPG – GNU Private Guard)

- sudo apt-get install docker-ce docker-ce-cli containerd.io : Để cài đặt các tài nguyên của Docker

Để sử dụng database mà cụ thể là SQL SERVER (2019) bằng docker thì Microsoft đã có images của sql ở docker hub và ta sẽ thực hiện lệnh docker pull để lấy images sql đó để chạy ở server ở trang chủ của docker : https://hub.docker.com/_/microsoft-mssql-server

Ta sẽ thực hiện các lệnh như sau :

- sudo docker pull mcr.microsoft.com/mssql/server:phiên bản-latest thì ở đây máy chủ này đang sử dụng phiên bản 2019 thì sẽ thay phiên bản bằng 2019
- Tiếp theo ta sẽ chạy 1 lệnh nữa để chạy cái images đó thành 1 container trên docker
 - o docker run -e "ACCEPT_EULA =Y" -e "SA_PASSWORD = MatKhau" - p 1433:1433 –name ten --hostname ten -d <https://mcr.microsoft.com/mssql/server:2019-latest>

Để kiểm tra container có hoạt động không ta gõ lệnh docker ps để xem máy chủ có những container nào đang hoạt động

```
root@Ubuntu-51782075:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
292c73566fd7 da2_frontend "/docker-entrypoint.s..." 2 days ago Up 2 days 0.0.0.0:80->80/tcp, ::0->80/tcp frontda2
8f64uf5913f8 da2_backend "/docker-entrypoint.s..." 2 days ago Up 2 days 0.0.0.0:3002->3002/tcp, ::0->3002/tcp cdhd2
922ba2001dcd mcr.microsoft.com/mssql/server:2019-latest "/opt/mssql/bin/pow..." 4 weeks ago Up 2 days 0.0.0.0:1433->1433/tcp, ::0->1433/tcp backendda2
root@Ubuntu-51782075:~#
```

Hình 2.10: Hình ảnh thẻ hiện máy chủ có các container đang hoạt động

2.2.Cách cài đặt gitlab CI/CD vào source code

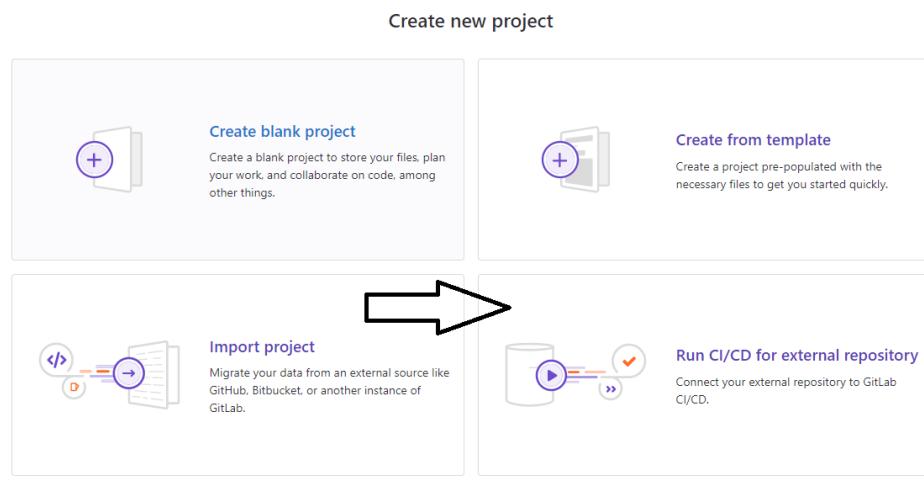
Để thiết lập gitlab CI/CD vào trong mã nguồn của dự án ta cần thực hiện chuyển đổi mã nguồn dự án từ GitHub sang GitLab vì trong Gitlab có tích hợp sẵn và có nhiều tài liệu về việc cấu hình CI/CD trên này ta sẽ thực hiện như sau



Hình 2.11: Bước một tiến hành tạo dự án mới trên GitLab

Ta vào gitlab.com và đăng nhập vào gitlab

Ở màn hình dự án ta chọn và tạo mới 1 dự án (New Project)



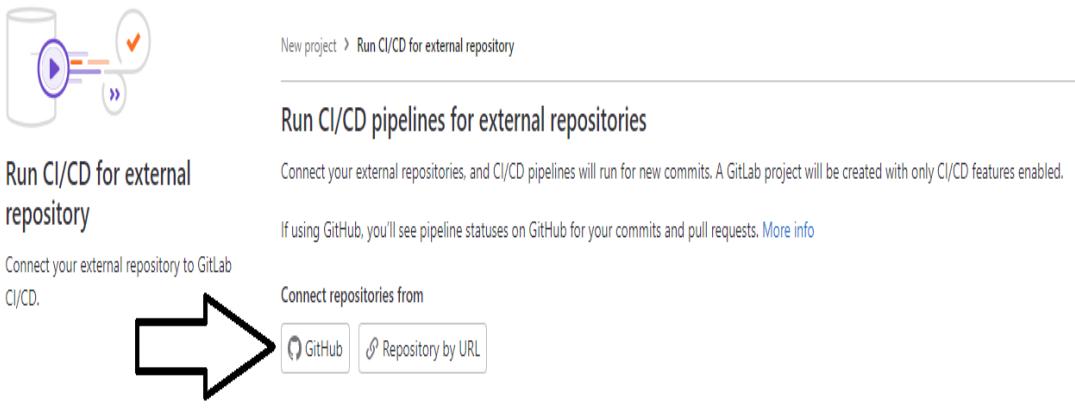
Hình 2.12: Bước hai chọn chạy dự án mở rộng sử dụng CI/CD

Bước tiếp theo sau khi chọn tạo mới dự án ta sẽ có các lựa chọn như sau :

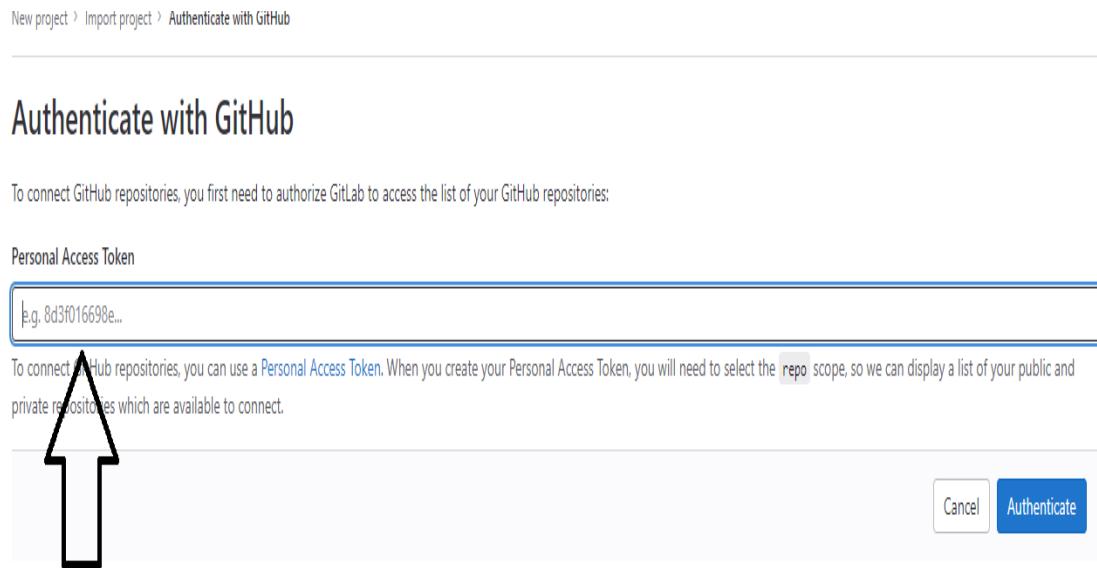
- Tạo ra 1 dự án trống

- Tạo 1 dự án trống tạo từ mẫu có sẵn
- Thêm dự án bằng câu lệnh
- Chạy dự án mở rộng thêm CI/CD

Với dự án hiện tại ta sẽ chọn lựa chọn là chạy dự án mở rộng thêm CI/CD



Hình 2.13 : Bước ba chọn sử dụng dự án từ nền tảng GitHub



Hình 2.14: Màn hình yêu cầu ta nhập token ủy quyền

Sau khi chọn vào lựa chọn liên kết với github ta sẽ được chuyển đến màn hình yêu cầu mã token ủy quyền cho gitlab có thể liên kết với github

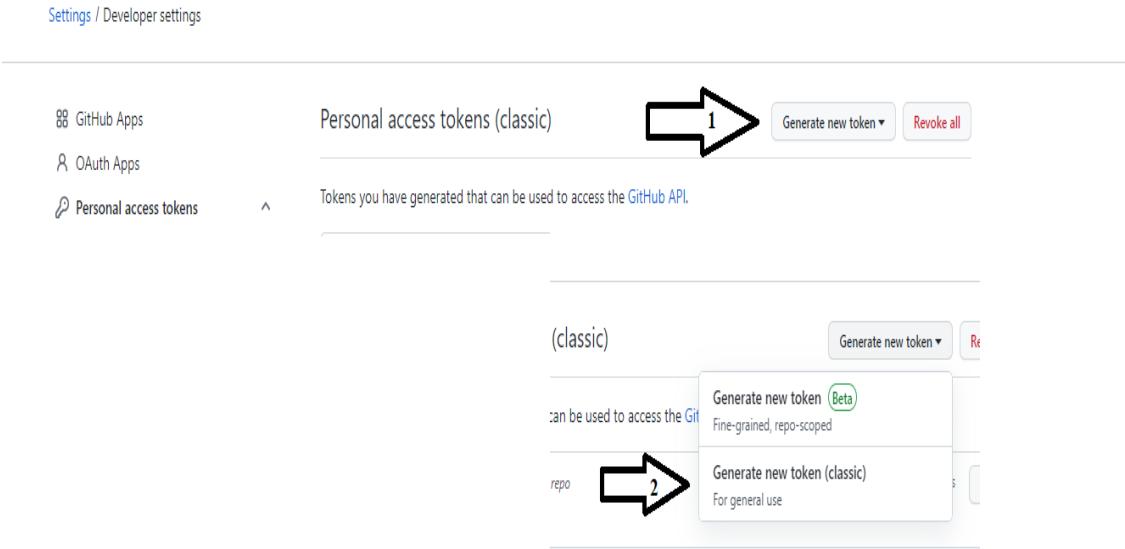
The screenshot shows a GitHub repository page for a user named 'datnguyenkien'. The repository has 2 branches and 0 tags. The commit history is listed, showing various commits from 'Back-End' and other files like '.gitignore', 'README.md', and 'docker-compose.yml'. On the right side, there's a sidebar with user information and a 'Settings' button highlighted with a large black arrow pointing to it. Other options in the sidebar include 'About', 'Readme', 'Stars', 'Watching', 'Forks', 'Releases', 'Packages', and 'Sign out'.

Hình 2.15: Bước bốn quay lại github để cài đặt

Ta sẽ quay lại github chừa dự án và thực hiện chọn vào settings

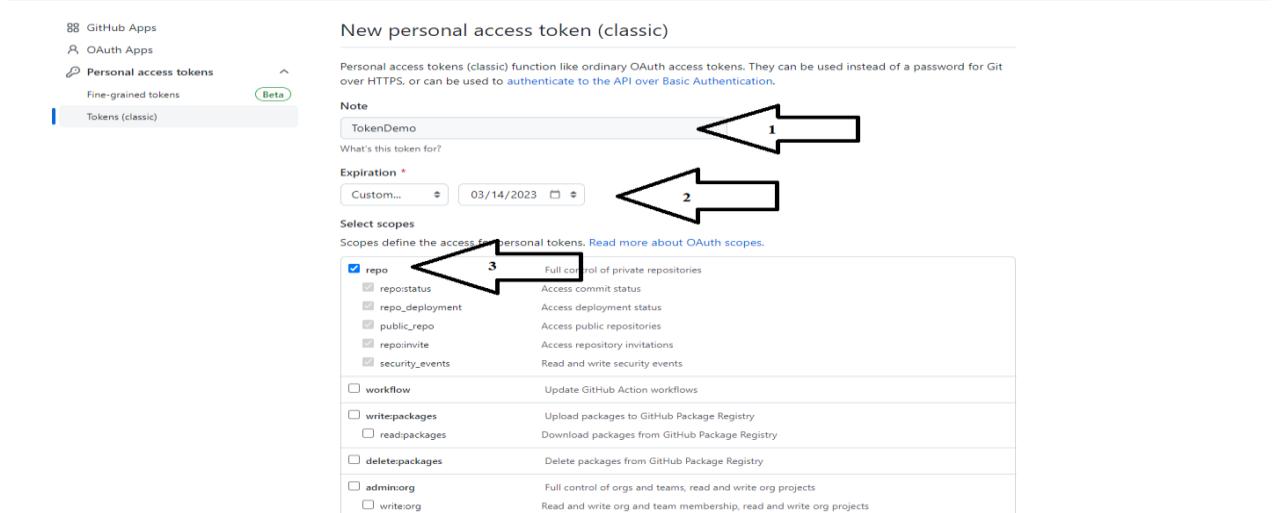
The screenshot shows the 'Settings' page on GitHub. It includes sections for 'Security', 'Code security and analysis', 'Integrations', 'Applications', 'Scheduled reminders', 'Archives', 'Security log', and 'Sponsorship log'. A large black arrow points to the 'Developer settings' link at the bottom of the left sidebar. The main content area is titled 'Contributions & Activity' and contains options for making the profile private, hiding activity, and including private contributions. There is also an 'Update preferences' button.

Hình 2.16: Bước năm chọn vào mục Developer settings



Hình 2.17: Tạo token ủy quyền

Tại đây ta thực hiện chọn “Generate new token” để tạo mã ủy quyền cho gitlab (1),
Chọn tiếp lựa chọn Generate new token (classic) (2)



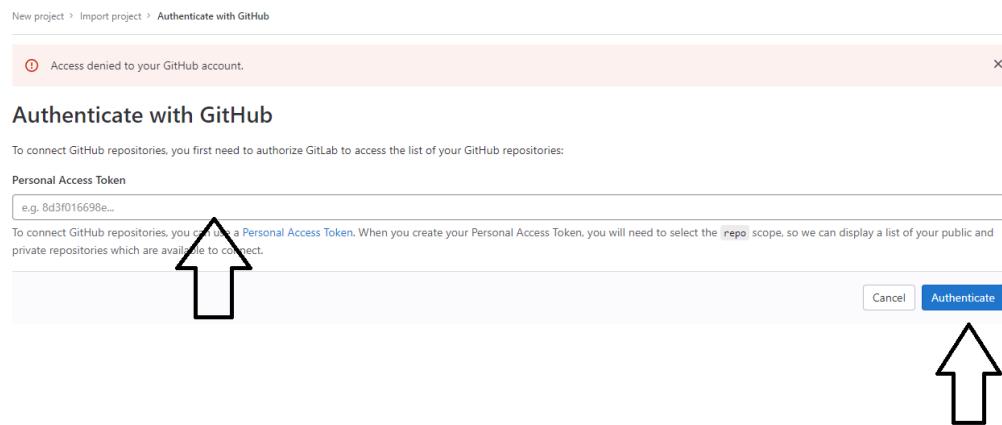
Hình 2.18: Bước tiếp theo điền các thông tin khi tạo token ủy quyền

Ta sẽ đặt tên cho Token (1), Chọn thời hạn hết hạn của Token này (2) và cuối cùng là lựa chọn các quyền thao tác với dự án của Token này (3)

[Settings / Developer settings](#)

The screenshot shows the GitHub Developer Settings interface. On the left, there's a sidebar with options like GitHub Apps, OAuth Apps, Personal access tokens (selected), and Fine-grained tokens (Beta). The main area is titled "Personal access tokens (classic)" with buttons for "Generate new token" and "Revoke all". Below this, it says "Tokens you have generated that can be used to access the GitHub API." A note at the top of the token list says "Make sure to copy your personal access token now. You won't be able to see it again!" A large black arrow points to the first token listed, which is "ghp_ixlMBezdMcYogBD2ZTGUzYxVN80r3G0ZPB7t".

Hình 2.19 : Sau khi khởi tạo token ta sẽ nhận được 1 dãy chữ cái ngẫu nhiên
Ta quay lại màn hình ở gitlab và điền token ủy quyền vào ô và chọn ủy quyền
(Authenticate)



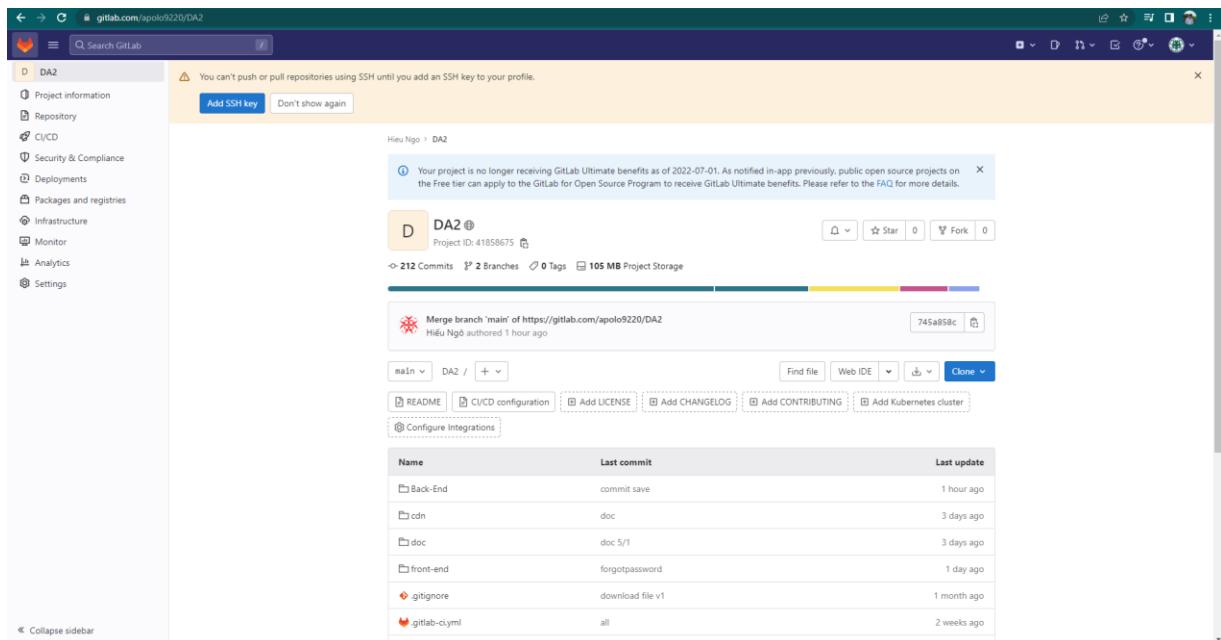
Hình 2.20: Điền token ủy quyền cho gitlab

The screenshot shows the "Import repositories from GitHub" dialog in GitLab. It lists 15 repositories being imported. At the top, there's a message: "Cancelling project import failed: The import cannot be canceled because it is finished". Below this, there's a section for "Import 15 repositories" with a "Filter by name" search bar. The list includes repositories like "Phoenix22121999/DA2" (status: Complete, Go to project), "NolanNgo/Tuhoc1" (status: Complete, Go to project), "NolanNgo/mytraning" (status: Not started, Connect), and "NolanNgo/project-". Three arrows point to specific elements: 1 points to the "Import 15 repositories" button; 2 points to the "Connect" button for the "mytraning" repository; 3 points to the "Go to project" button for the "DA2" repository.

Hình 2.21: Màn hình sau khi ủy quyền

Sau khi ủy quyền xong thì sẽ hiện ra danh sách các dự án mà tài khoản github tạo Access Token có

Nếu có nhiều dự án ta sẽ sử dụng bộ lọc để lấy được dự án ta cần dùng CI/CD (1), Nếu có dự án ta sẽ chọn Connect và đợi khoảng vài phút để Gitlab sẽ lấy dự án từ Github về (2) còn nếu đã kết nối dự án rồi ta sẽ chọn vào Go to project để chuyển hướng đến dự án của ta cần cấu hình CI/CD (3)



Hình 2.22: Dự án từ Github đã được đưa sang GitLab

Như thế ta hoàn thành xong bước đầu tiên là lấy dự án từ Github sang Gitlab sau đó ta thực hiện việc cấu hình chạy CI/CD trên dự án này với nền tảng Gitlab

Sau khi ta có được dự án ở nền tảng Gitlab ta sẽ tiến hành thiết lập công cụ CI/CD vào trong dự án

Ta sẽ tạo 1 tập tin với tên là .gitlab-ci.yml : Đây sẽ là tập tin chứa mã nguồn được viết theo chuẩn của CI/CD để mỗi lần có sự thay đổi về mã nguồn của dự án mà cảng cứ vào đó mà CI/CD dựa vào tập tin này để build cũng như deploy

CHƯƠNG 3 – KẾT QUẢ VÀ KẾT LUẬN

3.1. Kết quả

3.1.1. Giao diện

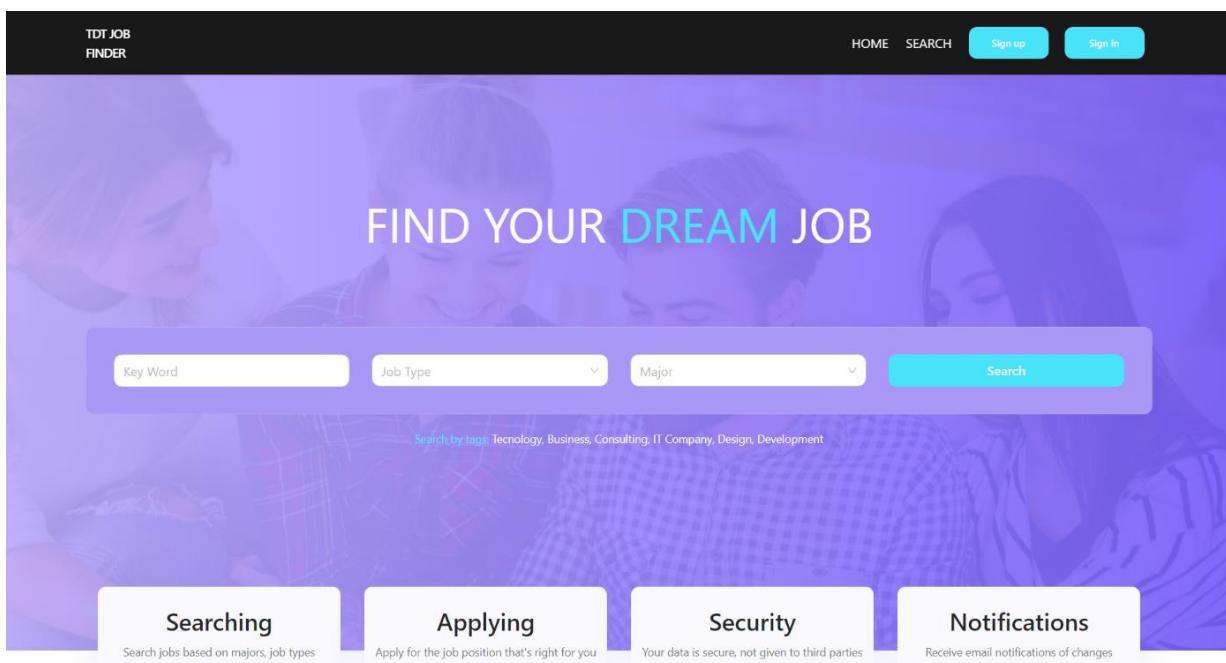
3.1.1.1. Trang chủ

- Banner

Trên cùng của trang chủ là banner, ở giữa banner là thanh tìm kiếm, sau khi chọn các thông tin mà mình muốn và bấm search, người dùng sẽ chuyển đến trang tìm kiếm với thông tin đã chọn.

Bên dưới là các tính năng của trang web bao gồm:

- Searching
- Applying
- Security
- Notification



Hình 3.40 Home Page - Banner

- Top Major

Phần tiếp theo là Top Major sẽ hiện 8 chuyên ngành mới nhất, nếu người dùng bấm vào 1 chuyên ngành sẽ được chuyển đến trang tìm kiếm với filter có chuyên ngành đã chọn. Nếu người dùng chọn All Major thì sẽ chuyển tới trang tìm kiếm với filter rỗng



Hình 3.41 Home Page – Top Major

- Feature Job

Phần feature job sẽ hiện 5 bài đăng mới nhất với thông tin chi tiết. Khi người dùng bấm vào view job sẽ được chuyển đến trang Job Detail

Nếu người dùng bấm vào All Job sẽ chuyển đến trang tìm kiếm với filter rỗng.

Featured Job

The screenshot displays five job listings from a recruitment website:

- 1** Hồ Chí Minh Full Time Language Male
Công ty Áo
- 2** Hà Nội Part Time IT Female
Công ty Áo
- 3** Công ty Áo
- 1** Công ty Áo
- 2** Công ty Áo
- 3** Công ty Áo

Each listing includes a small logo, the job title, company name, a 'View Job' button, and filters for Major (IT), Job Type (Full Time or Part Time), Location (various districts/cities), and Salary (0đ - 10.000.000đ to 0đ - 26.500.000đ).

All Job

Hình 3.42: Home Page – Featured Job

- Recruitment

Phần này sẽ khuyến khích các người dùng đăng ký tài khoản và sử dụng các chứng năng của trang web. Phần recruitment gồm 2 phần nhỏ:

- Phần cho candidate.
- Phần cho recruiter.

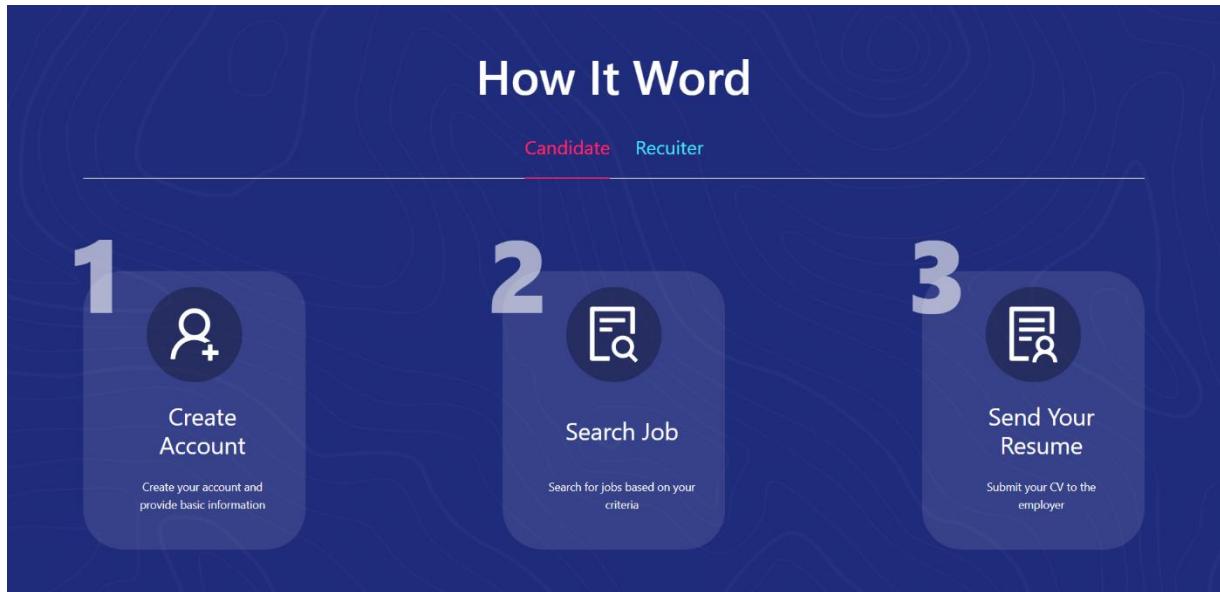
Khi người dùng bấm vào Upload CV hoặc Write your post sẽ được chuyển sang đăng nhập.



Hình 3.43: Home Page – Recruitement

- How it word

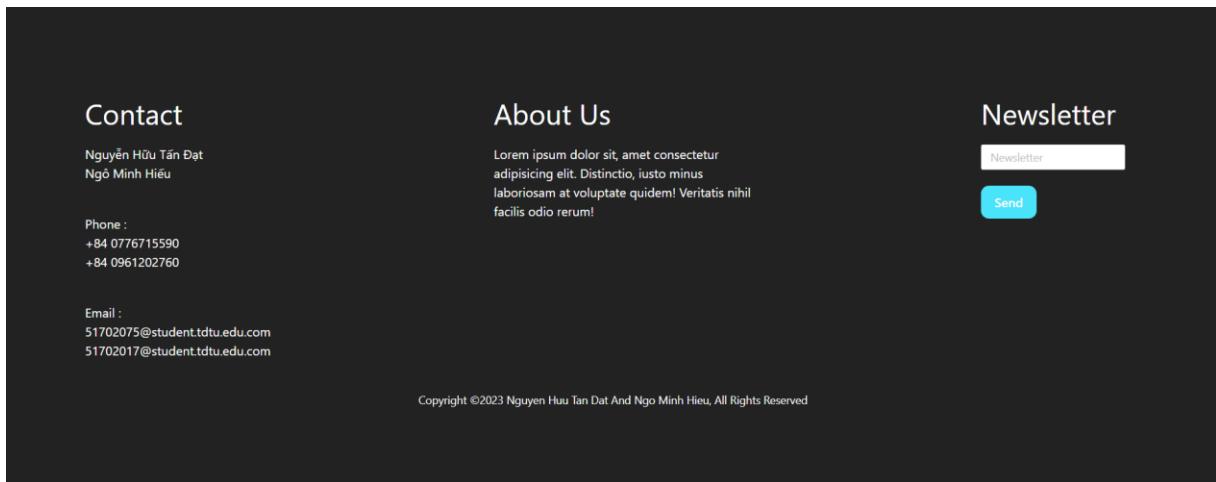
Phần cuối cùng của trang home là How it word, phần này sẽ hiện các bước sử dụng trang web cho candidate và recruiter



Hình 3.44: Home Page – How It Work

3.1.1.2. Footer

Phần footer sẽ hiển thị thông tin của Nguyễn Hữu Tân Đạt và Ngô Minh Hiếu



Hình 3.45: Footer

3.1.1.3. Header

Phần header gồm logo của trang web nằm bên trái, bên phải là thanh điều hướng. Thanh điều hướng sẽ thay đổi dựa trên loại người dùng đã đăng nhập. Nếu đăng nhập sẽ xuất hiện các trang tương ứng với loại người dùng, nếu chưa đăng nhập sẽ chỉ có 4 trang là home, search, đăng ký, đăng nhập



Hình 3.46: Header - UnAuth



Hình 3.47: Header - Candidate



Hình 3.48: Header – Recruiter

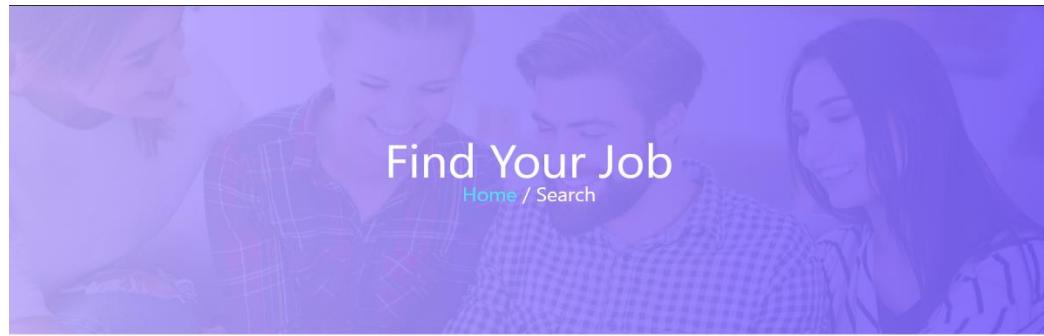


Hình 3.149: Header - Admin

3.1.1.4. Search

Trang search gồm 2 phần bên trái là thanh search filter, bên phải là danh sách các bài post được phân 10 bài 1 trang. Khi người dùng thay đổi thông tin bên thanh search filter thì kết quả sẽ trả về bên phải. Người dùng có thể search theo các thuộc tính sau:

- Từ khóa: sẽ trả về các bài đăng có tiêu đề chưa từ khóa mà người dùng nhập
- Vị trí: sẽ trả về các bài đăng có thành phố, quận/huyện, phường/xã mà người dùng nhập
- Chuyên ngành: sẽ trả về các bài đăng có chuyên ngành bao gồm các chuyên ngành mà người dùng chọn, người dùng có thể chọn được nhiều chuyên ngành 1 lúc
- Loại công việc: sẽ trả về các bài đăng có loại là loại công việc là người dùng chọn
- Ngày đăng: sẽ trả về các bài đăng được đăng trong khoảng mà người dùng chọn
- Khoản lương: sẽ trả về các bài post có mức lương nằm trong khoảng mà người dùng chọn
- Người dùng có thể kết hợp các filter với nhau để tìm kiếm



Key Words

Location

Province:

District:

Ward:

Major

Job Type

Full Time

Part Time

Date Post

All

Last 24 Hours

Last 7 Days

Last 14 Days

Last 30 Days

Salary Range

From:

To:

Showing 1-10 of 12 posts

Create Date ASC ▾

Rank	Job Title	Company	Action
1	Hồ Chí Minh Full Time Language Male	Công ty Áo	View Job
2	Hà Nội Part Time IT Female	Công ty Áo	View Job
3	1	Công ty Áo	View Job
4	2	Công ty Áo	View Job
5	3	Công ty Áo	View Job
6	4	Công ty Áo	View Job
7	5	Công ty Áo	View Job
8	6	Công ty Áo	View Job
9	7	Công ty Áo	View Job
10	8	Công ty Áo	View Job

< 1 2 >

Hình 3.11: Search page

Showing 1-10 of 12 posts

Create Date ASC ▾

Người dùng có thể sắp xếp các bài viết theo thứ tự ngày đăng.

Nếu người dùng đăng nhập với tư cách là ứng viên sẽ có thêm chức năng bookmark bài viết. Khi người dùng click vào bài đăng, bài đăng sẽ mở rộng ra, cho thấy các thông tin chi tiết: chuyên ngành, loại công việc, vị trí, mức lương.

Nếu click vào nút “View Job” thì sẽ chuyển qua trang job detail

3.1.1.5. Job Detail

Trang Job Detail sẽ hiển thị tất cả thông tin của bài đăng đó, bên trên là banner của bài đăng, bên trái sẽ có nút apply này sẽ thành applied nếu người dùng là ứng viên và đã ứng tuyển vào nút sẽ bị vô hiệu. Nếu người dùng chưa đăng nhập sẽ hiện thông báo yêu cầu đăng nhập với tài khoản ứng viên

Trong trường hợp người dùng đã đăng nhập với tài khoản ứng tuyển thì sẽ cho chọn cv để apply

Bên dưới là thông tin của bài đăng, bên trái là nội dung của bài đăng mà nhà tuyển dụng đã viết, bên phải là thông tin gồm:

- Của bài đăng:
 - Ngày đăng
 - Vị trí
 - Loại công việc
 - Chuyên ngành
 - Mức lương
 - Giới tính
- Của công ty đăng bài:
 - Tên công ty
 - Số điện thoại
 - Email
 - Vị trí

Hà Nội Part Time IT Female
Công ty Áo

Apply

Quasi vero, inquit, perpetua oratio rhetorum solum, non etiam philosophorum sit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc omni virtuti vitium contrario nomine opponitur. Num igitur eum postea censes anxio animo aut sollicito fuisse? Sed quid attinet de rebus tam aperitis plura requiri? *Duo Reges: constructio interrete.* Etenim semper illud extra est, quod arte comprehenditur.

Quod autem principium officii quaerunt, melius quam Pyrrho: Age, inquies, ista parva sunt. Hanc ergo intuens debet institutum illud quasi signum absolvere. Et quod est munus, quod opus sapientiae?

Sic tantum modo ad indicia veteris memoriae cognoscenda, curiosorum. Vitae autem degendae ratio maxime quidem illis placuit quieta. Recte dicens: *Omnia contraria, quos etiam insanos esse vultis.*

Restatis igitur vos;

Qui enim existimabat posse se miserum esse beatus non erit.

Praeteritis, inquit, gaudeo.

Ita fit beatae vitae domina fortuna, quam Epicurus ait exiguum intervenire sapienti.

Magna laus.

Si longus, levis dictata sunt.

Scrupulum, inquam, abeunt;

Summum enim bonum exposuit vacuitatem doloris;

Quo modo?

Serpere angulculos, nare anaticulas, evolare merulas, cornibus uti videmus boves, nepas aculeis.

Age sane, inquam.

Iubet igitur nos Pythius Apollo noscere nosmet ipsos.

1. Quasi ego id curem, quid ille aiat aut neget.
2. Quodsi ipsam honestatem undique perfectam atque absolutam.
3. Quid est igitur, cur ita semper deum appellet Epicurus beatum et aeternum?
4. Sed haec ab Antiocho, familiari nostro, dicuntur multo melius et fortius, quam a Stasea dicebantur.
5. Iam id ipsum absurdum, maximum malum neglegi.
6. Te ipsum, dignissimum maioribus tuis, voluptasne induxit, ut adolescentulus eriperes P.
7. Vide, quantum, inquam, fallare, Törquate.

- Animi enim quoque dolores percipiet omnibus partibus maiores quam corporis.
- Euro, inquit adridens; inquam, hac quidem de re;
- Non enim, si omnia non sequebatur, idcirco non erat ortus illinc.
- Quorum sine causa fieri nihil putandum est.
- Et nunc quidem quod eam tuetur, ut de vite potissimum loquar, est id extrinsecus;
- Sed tu istuc dixi bene Latine, parum plane.

Atqui si, ut convenire debet inter nos, est quaedam appetitio naturalis ea, quae secundum naturam sunt, appetens, eorum omnium est aliquae summa facienda.

Tollit definitiones, nihil de dividendo ac partiendo docet, non quo modo efficiatur concludaturque ratio tradit, non qua via captiosa solvantur ambigua distinguantur ostendit;

Nisi enim id faceret, cum Plato Aegyptum peragravit, ut a sacerdotibus barbaris numeros et caelestia acciperet?

Ab his oratores, ab his imperatores ac rerum publicarum principes exierunt. *Mihī enim erit isdem istis fortasse iam utendum.* Aut unde est hoc contritum vetustate proverbiū: quicum in tenebris? Neque enim disputari sine reprehensione nec cum iracundia aut pertinacia recte disputari potest. *Sed ego in hoc resisto;* Quod autem ratione actum est, id officium appellamus. Omnes enim iucundum motum, quo sensus hilaretur. *Invidiosum nomen est, infame, suspectum.* *Primum quid tu dics breve?* Non queritur autem quid naturae tuae consentaneum sit, sed quid disciplinae. Hoc ipsum elegantius ponи meliusque potuit. Quae in controversiam veniunt, de iis, si placet, disseramus.

Job Overview

Date Posted:
30/11/2022

Location:
141/80 Tâm Danh, Phường Phúc Xá, Quận Ba Đình, Thành phố Hà Nội

Job Type:
Part Time

Job Major:
IT

Salary:
0 ₫-21.000.000 ₫

Gender:
2

Company Overview

Name: Công ty Áo

Phone: 0776715590

Email: nghuutandat@gmail.com

Location: 141/80 Tâm Danh

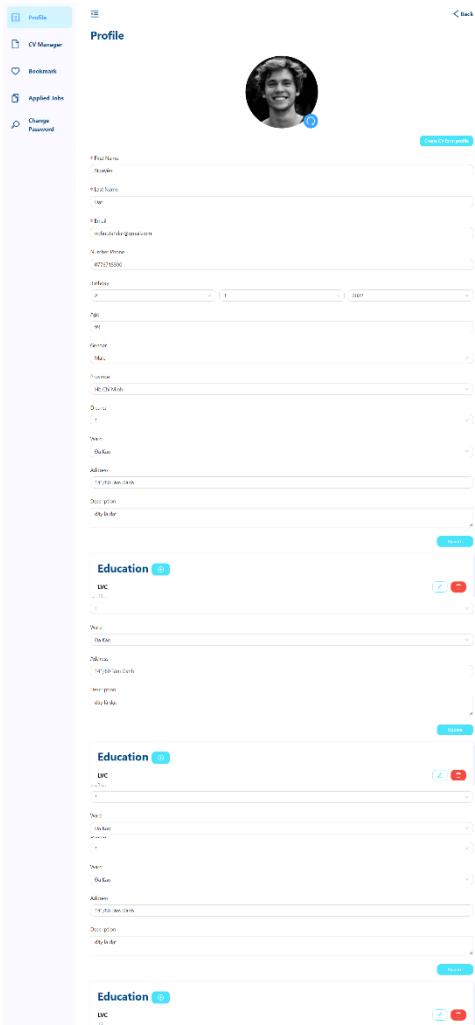
Hình 3.50: Job Detail

3.1.1.6. Candidate

Bên trái là sidebar dùng để di chuyển giữa các trang con trong trang Candidate

- Profile

Mặc định khi vào trang Candidate sẽ là trang Profile ở trang này sẽ hiện lên các thông tin của người dùng, người dùng có thể thay đổi avatar hoặc các thông tin khác.



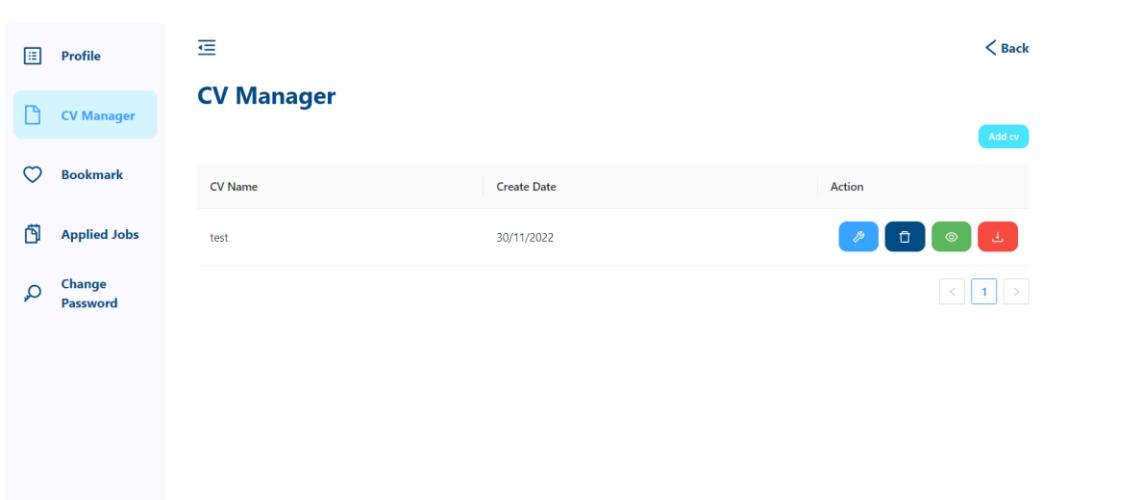
Hình 3.51: Candidate – Profile

Khi bấm vào thay đổi avatar sẽ hiện lên modal, khi người dùng click vào avatar hiện tại sẽ được chọn 1 file hình để upload lên, và chỉ được upload hình. Sau khi upload hình người dùng sẽ được đưa đến trang chỉnh sửa, người dùng có thể cắt hoặc xoay hình, sau đó người dùng có thể tiếp tục bấm Ok để tiếp tục lưu avatar

Sau khi chỉnh sửa các thông tin người dùng có thể bấm update để cập nhật thông tin.

Người dùng còn có thể tạo cv từ thông tin của profile

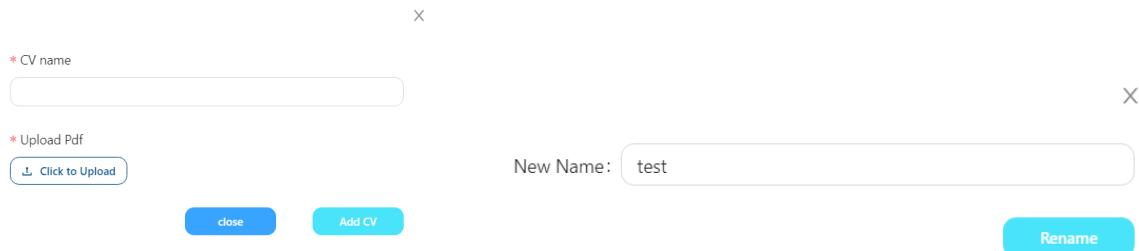
- CV manager



Hình 3.52: Candidate – CV Manager

Ở trang CV manager người dùng có thể thêm, đổi tên, xóa, xem, tải xuống các CV

Khi bấm “Add cv” sẽ hiện modal cho phép người dùng nhập tên, và chọn file pdf để tạo 1 cv



Nếu người dùng chọn Rename, sẽ hiện một modal để người dùng nhập tên mới

Nếu người dùng chọn View, sẽ hiện lên modal với nội dung của file pdf đó

Nếu người dùng chọn delete thì sẽ xóa CV

Nếu người dùng chọn Download, file sẽ được tự động download về máy của người dùng

- Bookmarks

Trang bookmarks sẽ thể hiện các post mà người dùng đã đánh dấu. Người dùng có thể xóa bookmark hoặc xem bài viết, khi bấm vào xem bài viết người dù sẽ được đưa đến trang post detail của bài đăng đó

Post Title	Create Date	Action
2	21/12/2022	
3	21/12/2022	
4	21/12/2022	
5	21/12/2022	
6	3/1/2023	
7	3/1/2023	

Hình 3.53: Candidate – Bookmarks

- Applied Jobs

Post	CV	Create Date	Action
Hồ Chí Minh Full Time Language Male	test	30/11/2022	

Hình 3.54: Candidate – Applied Job

Ở trang này sẽ hiển thị những bài post mà người dùng đã apply và cv họ đã apply, họ có thể xóa để hủy apply vào bài đăng đó

- Change passwords

Người dùng có thể thay đổi passwords ở đây, sau khi change passwords sẽ gửi email thông báo về mail của user

Change Password

* Old Password

* Password

* Confirm Password

Submit

Hình 3.55: Candidate – Change passwords

3.1.1.7. Recruiter

- Profile

Profile

* Company Name
Công Ty CP Viễn Thông FPT

* Email
fptdu@gmail.com

Numberphone
0961202763

Founded Date

Province

District

Ward

Address

Description

Là thành viên thuộc Tập đoàn công nghệ hàng đầu Việt Nam FPT, Công ty Cổ phần Viễn thông FPT (tên gọi tắt là FPT Telecom) hiện là một trong những nhà cung cấp dịch vụ viễn thông và Internet có uy tín và được khách hàng yêu mến tại Việt Nam và khu vực.

Thành lập ngày 31/01/1997, khởi nguồn từ Trung tâm Dịch vụ Trực tuyến do 4 thành viên sáng lập cùng sản phẩm mạng Intranet đầu tiên của Việt Nam mang tên "Trí tuệ Việt Nam - TTVN", sản phẩm được coi là đột nón móng cho sự phát triển của Internet tại Việt Nam. Sau 20 năm hoạt động, FPT Telecom đã lớn mạnh vượt bậc với hơn 7.000 nhân viên chính thức, gần 200 văn phòng điểm giao dịch thuộc hơn 80 chi nhánh tại 59 tỉnh thành trên toàn quốc, bên cạnh đó, Công ty đã và đang đặt dấu ấn trên trường quốc tế bằng 8 chi nhánh trải dài khắp Campuchia, cũng như việc được cấp giấy phép kinh doanh dịch vụ tại Myanmar.

Update

Hình 3.56: Recruiter - Profile

Ở đây người dùng có thể thay đổi thông tin của công ty và logo của công ty

- Post Manager

The screenshot shows a 'Post Manager' section within a recruitment software. On the left, there's a sidebar with 'Profile', 'Post Manager' (which is selected and highlighted in blue), 'All Applicants', and 'Change Password'. The main area has a header 'Post Manager' with a back arrow and a 'Back' button. Below the header is a 'Add new job' button. A table lists five job posts:

	Post Title	Create Date	Active	Action
1	Hồ Chí Minh Full Time Language Male	30/11/2022	<input checked="" type="checkbox"/>	
2	Hà Nội Part Time IT Female	30/11/2022	<input checked="" type="checkbox"/>	
3	1	1/12/2022	<input checked="" type="checkbox"/>	
4	2	1/12/2022	<input checked="" type="checkbox"/>	
5	3	1/12/2022	<input checked="" type="checkbox"/>	

Hình 3.57: Recruiter – Post Manager

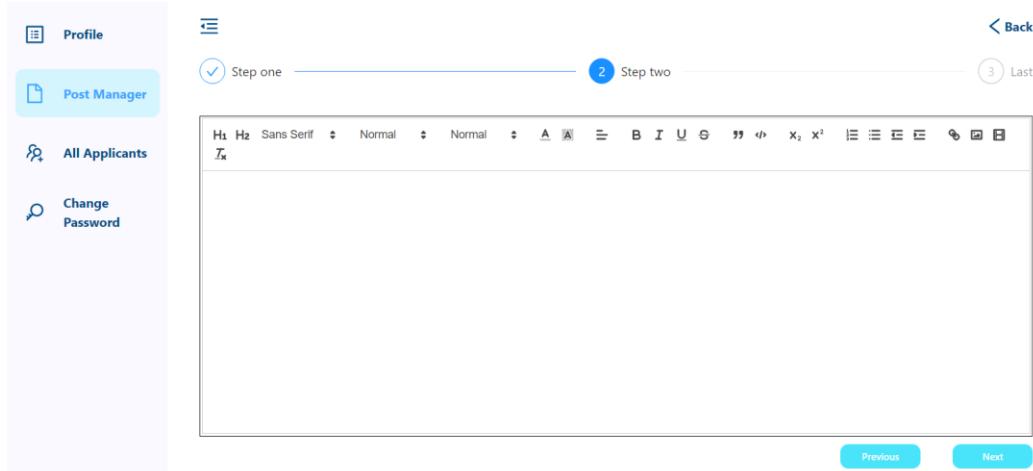
Ở trang post manager người dùng có thể xem các bài đăng hoặc thêm xóa sửa các bài đăng. Khi bấm vào add new job người dùng sẽ được đưa đến trang thêm bài đăng mới.

The screenshot shows the first step of creating a new job post. The top navigation bar includes 'Profile', 'Post Manager' (selected), 'All Applicants', and 'Change Password'. The main area is titled 'Step one' with a progress bar showing 'Step two' and 'Last'. The form fields are:

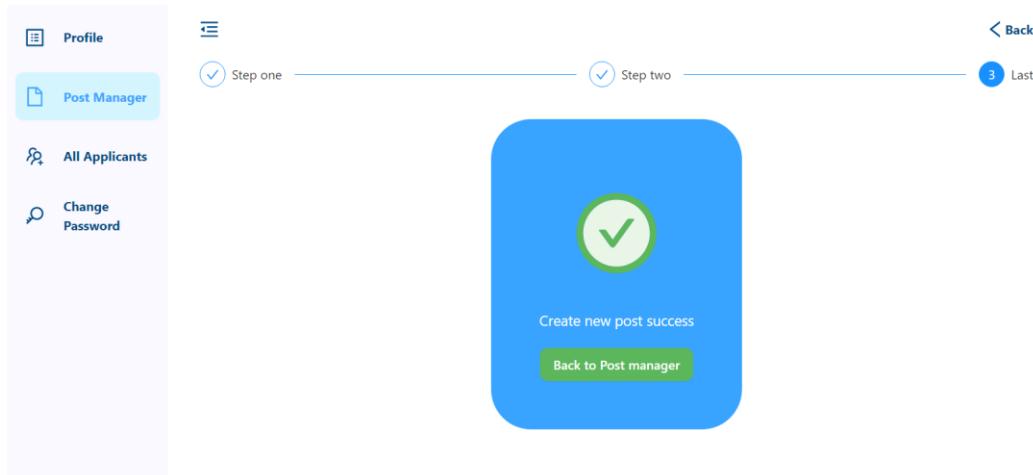
- Title:** (Input field)
- Job Type:** (Input field with placeholder 'Please select job type')
- Major:** (Input field with placeholder 'Please select majors')
- Gender:** (Input field with placeholder 'Please select gender')
- Salary:** (A range slider with markers at 0đ, 10.000.000đ, 25.000.000đ, 40.000.000đ, and 50.000.000đ)
- Province:** (Input field with placeholder 'Province')
- District:** (Input field with placeholder 'District')
- Ward:** (Input field with placeholder 'Ward')
- Address:** (Input field)

A 'Next' button is located at the bottom right of the form.

Hình 3.58: Recruiter – New post step one



Hình 3.59: Recruiter – New post step two

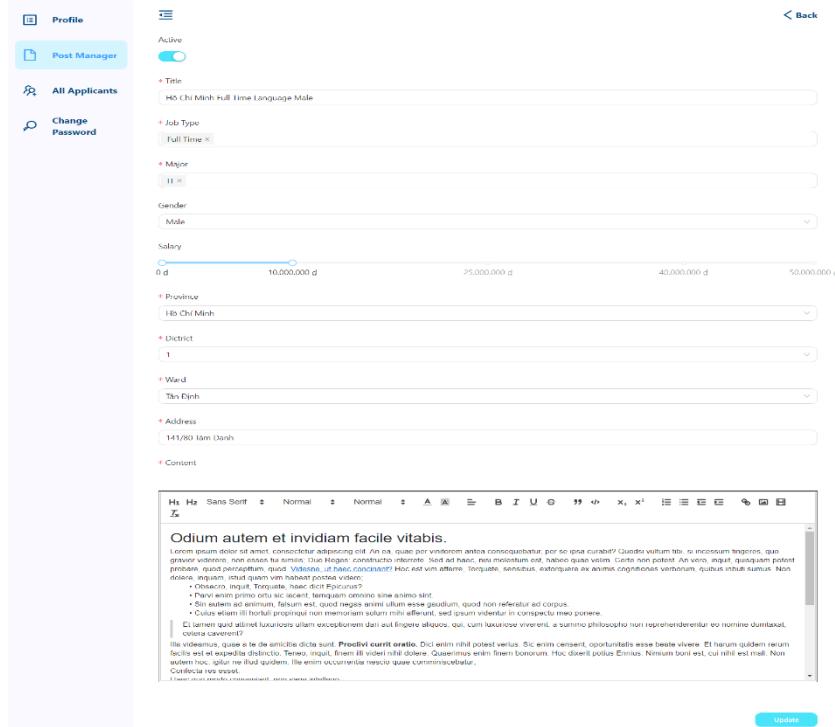


Hình 3.60: Recruiter – New post step three

Trang tạo bài đăng mới có 3 bước:

- Bước 1: Điền các thông tin thuộc tính của bài đăng
- Bước 2: Thêm nội dung của bài đăng, có hỗ trợ rich text editor
- Bước 3: Chờ kết quả trả về khi đã thành công hay không

Người dùng còn có thể chỉnh sửa các bài post đã đăng bằng cách bấm vào nút edit



Hình 3.61: Recruiter – Edit Post

Sau khi người dùng sửa đổi các thông tin của bài post người dùng có thể nhấn nút update để cập nhật lại các bài đăng. Người dùng còn có thể ẩn bài đăng khỏi việc tìm kiếm hoặc xóa bài đăng

- All Applicant

Post	CV	Create Date	Action
Hồ Chí Minh Full Time Language Male	test	30/11/2022	

Hình 3.62: Recruiter – All Applicant

Trang All Applicant sẽ hiện danh sách các ứng viên đã ứng tuyển vào các bài đăng. Người dùng có thể xem hoặc tải CV được ứng tuyển về

- Change Password

The screenshot shows a 'Change Password' form. On the left is a sidebar with 'Profile', 'Post Manager', 'All Applicants', and 'Change Password' (which is highlighted in blue). The main area has a title 'Change Password' and three input fields: 'Old Password', 'Password', and 'Confirm Password', each with a small circular icon. At the bottom right is a 'Submit' button.

Hình 3.63: Recruiter – Change Password

Người dùng có thể đổi passwords và sẽ có email thông báo sao khi đổi passwords

3.1.1.8. Admin

- Account Manager

Ở trang này Admin có thể thấy danh sách các tài khoản của người dùng, loại người dùng cũng như ngày tạo. Khi bấm vào View thì admin sẽ được chuyển sang trang account detail. Tùy theo loại người dùng trang account detail sẽ hiện các thông tin khác nhau

The screenshot shows an 'Account Manager' page. On the left is a sidebar with 'Dashboard', 'Account Manager' (highlighted in blue), 'Request Manager', 'All Post', 'User Type', 'Job Type', and 'Majors'. The main area has a title 'Account Manager', a 'Total Account: 21' message, and a search bar. Below is a table with columns: Username, User Type, Create Date, and Action. The table lists several user entries with their details. At the bottom right are navigation arrows and page numbers 1, 2, 3.

Username	User Type	Create Date	Action
taikhoantes4	Nhóm ứng viên	29/11/2022	
adminitest	Nhóm Quản Lý	29/11/2022	
administrator1	Nhóm Quản Lý	30/11/2022	
administrator2	Nhóm Quản Lý	2/12/2022	
administrator4	Nhóm ứng viên	2/12/2022	
51702017@student.tdtu.edu.vn	Nhóm ứng viên	2/12/2022	
administrator5	Nhóm ứng viên	30/12/2022	
51702075@student.tdtu.edu.vn	Nhóm ứng viên	27/1/2023	
candidate1	Nhóm ứng viên	7/2/2023	
admin	Nhóm Quản Lý	7/2/2023	

Hình 3.64: Admin – Account Manager

Trang Account Detail bao gồm 2 phần:

Phần đầu sẽ hiện các thông tin cơ bản của tài khoản.

- Nếu loại tài khoản là candidate, sẽ hiện các thông tin của người dùng như tên, tuổi, giới tính,
- Nếu loại là người dùng là recruiter thì sẽ hiện các thông tin của công ty.

Phần hai của trang account sẽ bao gồm các data của người dùng đó:

- Loại người dùng là candidate thì sẽ hiện các bảng thông tin: CV, Applied Jobs, Bookmark, admin không có quyền chỉnh sửa những thông tin này.
- Loại người dùng là recruiter thì sẽ hiện các bảng thông tin: các bài đăng mà tài khoản này đã tạo, các ứng viên applied vào các bài đăng của tài khoản này. admin không có quyền chỉnh sửa những thông tin này.

Nếu loại người dùng là Admin sẽ không thể xem chi tiết account

The screenshot displays the 'Account Manager' interface for a recruiter account. On the left, there's a sidebar with filters for 'All Post', 'User Type', 'Job Type', and 'Majors'. The main area is divided into two tabs: 'Information' and 'Data'.

Information Tab:

ID:	3
Username:	recruiter
Age:	
First Name:	
Last Name:	
Email:	nghuutandat@gmail.com
Gender:	
Numberphone:	0776715590
Province:	
District:	
Ward:	

Data Tab:

Post Section:

Post Title	Create Date	Active
Hồ Chí Minh Full Time Language Male	30/11/2022	<input checked="" type="checkbox"/>
Hà Nội Part Time IT Female	30/11/2022	<input checked="" type="checkbox"/>

All Applicant Section:

Post	CV	Create Date
Hồ Chí Minh Full Time Language Male	test	30/11/2022

Hình 3.265: Admin - Account Detail – Recruiter

The screenshot shows the 'Account Manager' section for a candidate account. The 'Information' tab is active, displaying details such as Id (2), Username (candidate), Age (7), First Name (Nguyễn), Last Name (Đạt), Email (nghiautandat@gmail.com), Gender (Male), Numberphone (0776715590), Province, District, and Ward. Below this is the 'Data' tab, which contains three sections: 'CV', 'Applied Jobs', and 'Bookmark Post'. The 'CV' section lists two entries: 'test' (Create Date: 30/11/2022) and 'Tmp' (Create Date: 4/1/2023). The 'Applied Jobs' section lists one entry: 'Hồ Chí Minh Full Time Language Male' (CV: test, Create Date: 30/11/2022). The 'Bookmark Post' section lists two entries: 'Hà Nội Part Time IT Female' (Create Date: 3/1/2023) and 'Hồ Chí Minh Full Time Language Male' (Create Date: 3/1/2023).

Hình 3.66: Admin - Account Detail – Candidate

- All Post

Trang All Post sẽ hiện tất cả các bài post, kể cả các bài đã được ẩn đi, khi admin nhấp vào view sẽ dẫn tới trang detail tail post hiện các thông tin chi tiết của bài post

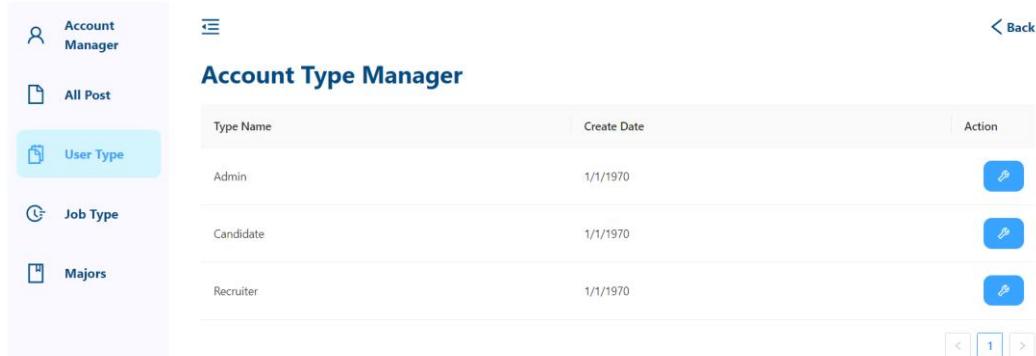
The screenshot shows the 'All Post' section of the Admin interface. It displays a table with columns: Post Title, Company, Create Date, and Action. There are five rows of data:

Post Title	Company	Create Date	Action
Hồ Chí Minh Full Time Language Male	Công ty Áo	30/11/2022	
Hà Nội Part Time IT Female	Công ty Áo	30/11/2022	
1	Công ty Áo	1/12/2022	
2	Công ty Áo	1/12/2022	
3	Công ty Áo	1/12/2022	
4	Công ty Áo	1/12/2022	

Hình 3.67: Admin – All Post

- Account Type Manager

Trang account type manager hiện 3 loại người dùng, admin có thể rename các loại người dùng nhưng không thể thêm hoặc xóa.

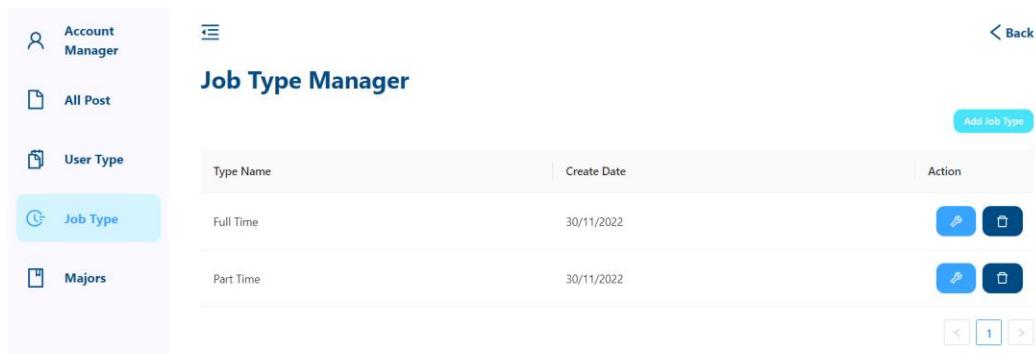


Type Name	Create Date	Action
Admin	1/1/1970	 
Candidate	1/1/1970	 
Recruiter	1/1/1970	 

Hình 3.68: Admin – Account Type Manager

- Job Type Manager

Trang job type manager hiện danh sách các loại công việc, admin có thể thêm, xóa, hoặc chỉnh sửa.



Type Name	Create Date	Action
Full Time	30/11/2022	 
Part Time	30/11/2022	 

Hình 3.69: Admin - Job Type Manager

- Major Manager

Trang major manager sẽ hiện danh sách các chuyên ngành, admin có thể thêm, xóa chỉnh sửa các chuyên ngành

Major Manager

Type Name	Create Date	Action
IT	30/11/2022	
Language	30/11/2022	
Pharmacy	30/11/2022	

Add Major

Hình 3.32: Admin – Major Manager

TDT JOB FINDER

Dashboard

- Account Manager
- Request Manager
- All Post
- User Type
- Job Type
- Majors

Candidate Account
10

Recruiter Account
2

Number Of Posts
10

Number Of Cvs
20

HOME **SEARCH** **ADMIN** admin **Sign out**

Hình 3.33: Admin – Admin – Thông kê của trang web

Request Manager

Total Post: 2

Username	User Type	Create Date	Action
2	Recruiter	8/2/2023	
3	Recruiter	8/2/2023	

Search by title

Dashboard

Account Manager

Request Manager

All Post

User Type

Job Type

Majors

Hình 3.34: Admin – Quản lý yêu cầu đăng ký

3.2.Kết luận

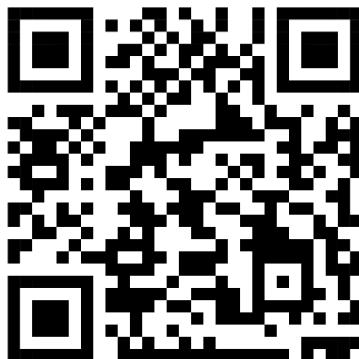
Sau quá trình tìm hiểu cũng như ứng dụng một số công nghệ vào việc thực hiện đề tài xây dựng trang web đăng tin tuyển dụng này, chúng em rút ra cho mình một số bài học như sau:

- Tìm hiểu quá trình hoạt động của một trang web bao gồm cả phía giao tiếp với cơ sở dữ liệu (API) và phía giành cho người dùng phổ thông sử dụng là (UI)
- Hiểu thêm 1 phần nhỏ về ngôn ngữ mà mình sử dụng .
- Tìm hiểu về thêm về ánh xạ đối tượng vào cơ sở dữ liệu thay cho việc truy suất dữ liệu thông qua việc viết các câu truy vấn trên nền SQL.
- Tìm hiểu các kiến thức cơ bản về một trong các công cụ hỗ trợ các nhà phát triển website có thể hoạt động ở nhiều môi trường khác nhau là docker.
- Hiểu thêm về như thế nào là một Images , Container , hay các câu lệnh thao tác với docker để chạy một trang ứng dụng có đầy đủ các tài nguyên mà không cần cài đặt quá nhiều.
- Tìm hiểu và xây dựng một bộ CI/CD tự động build mã nguồn và deploy chúng lên server thật.
- Tìm hiểu và thao tác với máy tính server với hệ điều hành Linux bằng các dòng lệnh ở cmd.
- Biết được một số kiến thức cơ bản về việc đưa dự án lên một con server được thuê và cũng như thêm tên miền gắn với trang web của nhóm.
- Sử dụng các công cụ hỗ trợ UI như antd ,.... Đọc tài liệu của các thư viện và ứng dụng nó vào trong dự án.
- Tiếp xúc với một ngôn ngữ mới là Typescript là một dạng nâng cao của JS và đòi hỏi phải viết code một cách chặt chẽ hạn chế lỗi xảy ra.
- Sử dụng React Lazy để tách các component, giúp cho trang web được load nhanh hơn cải thiện trải nghiệm người dùng.
- Sử dụng Redux để quản lý các state của React. Sử dụng react-router-dom để định tuyến phía client.

- Sử dụng google developer console để đăng ký client id dùng trong việc đăng nhập với google.

Tuy nhiên bên cách đó còn một số vấn đề và hạn chế như:

- Do thuê máy chủ server trên mạng nên sẽ phụ thuộc rất nhiều vào tốc độ mạng của server cũng như mạng của người dùng.
- Chỉ mới có một số kiến thức cơ bản về server nên sẽ dễ dẫn đến các tình huống có quá nhiều yêu cầu truy cập từ nhiều . Cân bằng tải cho server .
- Chưa áp dụng được kĩ thuật CI/CD runner vào trong dự án để không phụ thuộc vào docker hub.
- Đối với công khai web thì không đủ kinh phí để mua chứng chỉ SSL để nâng trang web lên https.
- Do chưa có kinh nghiệm về thiết kế nền UX (User Experience) và UI (User Interface) không được tốt.
- Chưa tối ưu hiệu xuất cho Front End.
- Đối với việc sử dụng CV thì chưa thể tự tạo CV theo template có sẵn mà phải sử dụng cv ngoài để upload vào trang web rồi mới dùng được chức năng ứng tuyển
Chúng em rất mong nhận được sự góp ý của quý Thầy , Cô và các bạn để có thể xem xét lại cũng như trau dồi thêm, tiếp tục tìm hiểu ứng dụng và hoàn thiện những nội dung chưa được tốt của đề tài này.



jobportaltdtu.com

Hình 3.35 Link đến Website <http://jobportaltdtu.com/>

DANH MỤC TÀI LIỆU THAM KHẢO

Tài liệu bao gồm :

- (n.d.). Retrieved from reactjs: <https://reactjs.org/>
- (n.d.). Retrieved from w3schools: https://www.w3schools.com/react/react_jsx.asp
- @nghiadd, N. (2017, 07 26). *CI, CD và ... DevOps ???* Retrieved from <https://viblo.asia/p/ci-cd-va-devops-07LKXYXDZV4>
- Ant Design of React.* (n.d.). Retrieved from ant.design: <https://4x.ant.design/docs/react/introduce>
- Blog, T. (n.d.). *Docker là gì? Tìm hiểu về Docker.* Retrieved from <https://topdev.vn/blog/docker-la-gi/>
- Blog, T. (n.d.). *Expressjs là gì? Tắt tân tật về Express.js.* (TopDev) Retrieved from <https://topdev.vn/blog/express-js-la-gi/>
- Blog, T. (n.d.). *NGINX là gì? Tổng quan về NGINX.* Retrieved from <https://topdev.vn/blog/nginx-la-gi/>
- Blog, T. (n.d.). *Triển khai CI/CD với Gitlab.* Retrieved from <https://topdev.vn/blog/trien-khai-ci-cd-voi-gitlab/>
- Chung, T. Q. (2021, 02 20). *"ORM" LÀ GÌ MÀ NHÀ NHÀ SỬ DỤNG?* Retrieved from <https://www.goccuachung.com/orm-la-gi-ma-nha-nha-su-dung/>
- Dan, B. N. (2020, 1 16). *Type vs Interface trong typescript.* Retrieved from viblo: <https://viblo.asia/p/type-vs-interface-trong-typescript-gJ599Gp5X2>
- Data, P. (n.d.). *Prisma schema.* Retrieved from <https://www.prisma.io/docs/concepts/components/prisma-schema>
- Deploy node application whit docker and nginx.* (2022, 07 17). Retrieved from <https://www.digitalocean.com/community/questions/deploy-node-application-whit-docker-and-nginx>
- DigitalOcean), J. E. (2016, 05 20). *How To Set Up Nginx Server Blocks (Virtual Hosts) on Ubuntu 16.04.* Retrieved from <https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04>
- DigitalOcean), K. Y. (2022, 10 28). *How To Run Nginx in a Docker Container on Ubuntu 22.04.* Retrieved from <https://www.digitalocean.com/community/tutorials/how-to-run-nginx-in-a-docker-container-on-ubuntu-22-04>
- DigitalOcean), S. (. (2022, 11 19). *How To Deploy a Go Web Application with Docker and Nginx on Ubuntu 22.04.* Retrieved from <https://www.digitalocean.com/community/tutorials/how-to-deploy-a-go-web-application-with-docker-and-nginx-on-ubuntu-22-04>
- Dinh, N. M. (2017, 02 26). *Node.js Tutorial: Phần 4: Express framework.* Retrieved from <https://viblo.asia/p/nodejs-tutorial-phan-4-express-framework-924IJXpNKPM>
- Docker. (n.d.). *Continuous integration with Docker.* Retrieved from <https://docs.docker.com/build/ci/>
- Docker. (n.d.). *Docker overview.* Retrieved from <https://docs.docker.com/get-started/overview/#docker-architecture>
- Erin Glass (Senior Manager, D. (2020, 04 25). *How To Install Nginx on Ubuntu 20.04.* Retrieved from <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04>

- G, H. (2022, 08 29). *VPS là gì? Tất cả các điều cần biết về Virtual Private Server*. Retrieved from <https://www.hostinger.vn/huong-dan/vps-la-gi-tat-ca-cac-dieu-can-biet-ve-may-chu-ao>
- Hoang, M. (2021, 07 30). *Tìm hiểu tổng quan về Nginx*. Retrieved from <https://viblo.asia/p/tim-hieu-tong-quan-ve-nginx-63vKjOExZ2R>
- Hung, N. (2021, 02 19). *NGINX là gì? NGINX có thể làm được gì?* Retrieved from <https://vietnix.vn/nginx-la-gi/>
- Hung, N. (2022, 04 25). *NodeJS là gì? Tổng quan kiến thức về NodeJS*. (vietnix) Retrieved from <https://vietnix.vn/nodejs-la-gi/>
- Hung, N. (2022, 01 14). *VPS là gì? Virtual Private Server dùng để làm gì?* Retrieved from <https://vietnix.vn/vps-la-gi/>
- ITNavi. (2021, 07 20). *CI/CD là gì? Những lợi ích mà mô hình CI/CD mang lại*. Retrieved from <https://itnavi.com.vn/blog/ci-cd-la-gi>
- Nam, L. (n.d.). *SQL Server là gì? cài đặt SQL server ra sao?* Retrieved from <https://longvan.net/sql-server-la-gi.html>
- ORM là gì? Tổng quan về ORM Framework.* (2017, 11 21). Retrieved from <https://stackjava.com/uncategorized/orm-la-gi-tong-quan-ve-orm-framework.html>
- ORM là gì? Tổng quan về ORM Framework.* (2021, 08 09). Retrieved from <https://hocspringboot.net/2021/08/09/orm-la-gi-tong-quan-ve-orm-framework-2/>
- Teky. (2022, 05 26). *CI/CD là gì? Thông tin chi tiết về chương trình CI và CD*. Retrieved from <https://teky.edu.vn/blog/ci-cd-la-gi/>
- TEL4VN. (2021, 01 01). *GIOI THIEU TỔNG QUAN VỀ KIẾN TRÚC CỦA DOCKER*. Retrieved from <https://tel4vn.edu.vn/blog/gioi-thieu-tong-quan-ve-kien-truc-cua-docker/>
- tenten.vn. (2022, 04 07). *VPS là gì? 6 điều cần biết về Virtual Private Server*. Retrieved from <https://tenten.vn/tin-tuc/vps-la-gi/>
- thanh, n. c. (2017, 6 28). *Tổng quan về JSX*. Retrieved from viblo: <https://viblo.asia/p/tong-quan-ve-jsx-Qbq5QqBL5D8>
- Toan, D. K. (2017, 5 30). Retrieved from viblo: <https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-icac-khai-niem-co-ban-V3m5WzjblO7>
- TypeScript Documentation*. (n.d.). Retrieved from typescriptlang: <https://www.typescriptlang.org/docs/>
- Usage Guide*. (n.d.). Retrieved from redux-toolkit: <https://redux-toolkit.js.org/usage/usage-guide>
- Usage Guides*. (n.d.). Retrieved from redux: <https://redux.js.org/usage/>
- VPS LÀ GÌ? VPS ĐƯỢC DÙNG ĐỂ LÀM GÌ?* (2016). Retrieved from <http://fit.tdc.edu.vn/blog/2016/04/vps-la-gi-vps-c-dung--lam-gi>