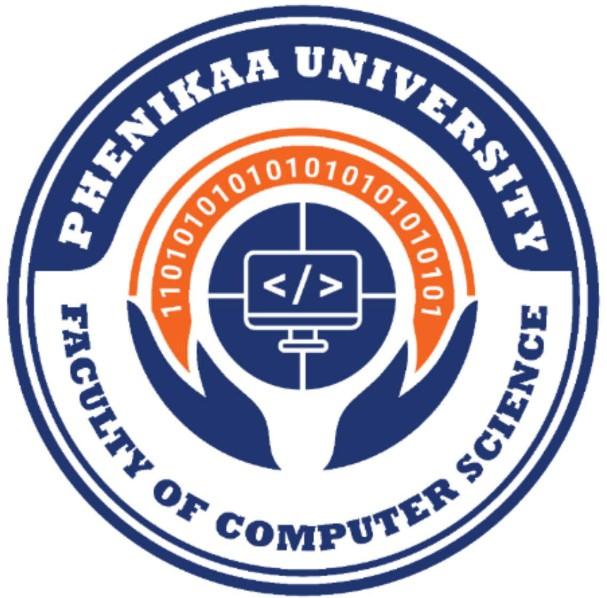


TRƯỜNG ĐẠI HỌC PHENIKAA KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO PHÂN TÍCH THIẾT KẾ PHẦN MỀM**

**PHÁT TRIỂN ỨNG DỤNG TÌM KIẾM VIỆC**

**JOB BRIDGE TRÊN MOBILE**

Sinh viên thực hiện: Vũ Tuấn Kiên - 22010373

Nguyễn Thế Hưng - 22010391

Đỗ Minh Nhật - 22010464

Vũ Hồng Đăng - 22010420

Nhóm 9

Khóa K16

Giáo Viên Hướng Dẫn ThS. Vũ Quang Dũng

**Hà Nội , Ngày 27 tháng 10 năm 2025**

**Mục lục**

[**Nhiệm vụ chính 6**](#_8ocj3bjvvvkx)

[1. Phát triển Frontend (Mobile Application Development) 6](#_6kuyazsgpil0)

[1.1 Xây dựng Tính năng Giao diện Người dùng 6](#_vufo2ttf21wy)

[1.2. Tối ưu hóa Hiệu suất và Bảo trì Mã nguồn 6](#_xp7nd7o20agj)

[2. Phát triển Backend (Server-Side Development) 6](#_5yjlns7a1r2g)

[2.1 Phát triển Hệ thống Lõi và Cơ sở Dữ liệu 6](#_qconffr5hiko)

[2.2 Xây dựng và Quản lý API 6](#_18bcqg8v8f86)

[**1.Giới thiệu 7**](#_xcu962dvf82r)

[1.1 Bối cảnh Thị trường Tuyển dụng và Kết nối Nhân sự 7](#_r1gj3gv7cq43)

[1.2 Vấn đề và Nhu cầu Cấp thiết 7](#_t8e5bbv5v2ak)

[1.3 Giải pháp Job Bridge 7](#_1vkynh1261r8)

[**2.Thiết kế và triển khai 7**](#_upws2e7ygltw)

[2.1 Các yêu cầu chức năng 7](#_btd9wdk6iwrd)

[2.1.1 Yêu cầu chức năng phía Frontend 7](#_jacc76y58gg1)

[2.1.2 Yêu cầu chức năng phía Backend 8](#_of8tx16llkcz)

[2.2 Các yêu cầu phi chức năng 8](#_t156fvy2uv5d)

[2.2.1 Các yêu cầu phi chức năng phía frontend 8](#_rkmm48apy2kh)

[2.2.2 Các yêu cầu phi chức năng liên quan đến Backend 9](#_g41suiypb2xe)

[**3.Các ràng buộc (Constraints) 11**](#_wno9u0sobehq)

[3.1 Ràng buộc về thời gian 11](#_pzfkj9o5k5k7)

[3.2 Ràng buộc công nghệ 11](#_274ggqf9zfva)

[3.3 Ràng buộc về nguồn lực 12](#_fbqdk1ruxzvb)

[3.4 Ràng buộc về người dùng 12](#_fyjzz371npyc)

[3.5 Ràng buộc về cạnh tranh 12](#_2uysi3uiu1mm)

[3.6 Các ràng buộc về triển khai 12](#_ckr4o0500hzs)

[3.6.1 Các ràng buộc kinh tế 12](#_s6yo16h4wotl)

[3.6.2 Các ràng buộc về đạo đức 14](#_r1tnzat6pzzs)

[**4.Mô hình hệ thống / Thiết kế giải pháp 16**](#_jrlmthfm2e34)

[4.1Các kịch bản của hệ thống (Use-cases) 16](#_12mlltdhsadt)

[4.1.1 Kịch bản dành cho người tìm việc (Candidates) 16](#_jiz9dbl2qnrk)

[4.1.2 Kịch bản dành cho nhà tuyển dụng (Employer) 18](#_y0eq4qg4yjv)

[4.1.3 Kịch bản dành cho trang admin 19](#_vwyyo2wu53d)

[4.2 Mô hình Use-case 19](#_o9jbbb7n87zx)

[4.2.1Mô hình Use-case liên quan đến Frontend 20](#_fzq0eapwf5fx)

[4.3 Mô hình lớp và đối tượng 29](#_b4vymap3fc5y)

[4.3.1 Mô hình Erd 29](#_ix5xfb5ekpxu)

[4.3.2 Mô hình Class 31](#_8nvez6mh12y7)

[4.4Các biểu đồ tuần tự 32](#_41gqjxhmsj4i)

[4.4.1Biểu đồ tuần tuần tự của người dùng 32](#_6qpsc7197qbo)

[**5.Các màn hình giao diện người dùng 38**](#_vn9n7qo3pcv3)

[**6. Công nghệ, thư viện, framework cho frontend 50**](#_su7xjf7jjxdm)

[6.1 Employer frontend 50](#_s63rv5ljqtbp)

[**7.Backend và cơ sở dữ liệu 52**](#_mvc85fu434x)

[7.1 Bộ công nghệ (TechStack) 52](#_f30clwes239q)

[7.2. Chi tiết Phát triển & Chức năng (Functionality Implementation) 52](#_viq00a8r3573)

[7.2.1. Module Xác thực và Ủy quyền (Authentication & Authorization) 52](#_yk53gsc8d13)

[7.2.2. Xử lý Dữ liệu và Mô hình (Models & Data Handling) 53](#_xh8fsummsmbd)

[7.2.3. Tích hợp Dịch vụ Bên ngoài (External Services Integration) 53](#_gkcvs2se3xqg)

[7.3. Hiệu năng và Bảo mật Hệ thống 53](#_7ghoe1okt0iw)

[7.3.1. Bảo mật API 53](#_3j3mc6rwmoah)

[7.3.2. Khả năng Mở rộng (Scalability) 53](#_rw5sskf2b175)

[**8.Một số thành phần khác của đồ án 53**](#_bz2q7ub8wn6y)

[8.1 Kế hoạch dự án 53](#_o6q70fvzj7bi)

[8.2 Đảm bảo thực hiện đúng làm việc nhóm 53](#_i6a5unig574j)

[8.3 Các vấn đề về đạo đức và làm việc chuyên nghiệp 54](#_x0ua89ysfnxf)

[**9.Kết luận 54**](#_jqvmcsjrzwxq)

[**Tài liệu tham khảo 55**](#_bunpx8lujsxn)

**DANH MỤC HÌNH ẢNH**

| STT | Hình ảnh | Trang |
| --- | --- | --- |
| 1 | Hình 4.2.1 Mô hình Use case tổng quát | 16 |
| 2 | Hình 4.3.1 Mô hình Erd | 21 |
| 3 | Hình 4.3.2 Mô hình Class | 22 |
| 4 | Hình 4.4.1 Biểu đồ tuần tự đăng nhập tài khoản | 23 |
| 5 | Hình 4.4.2 Biểu đồ tuần tự đăng ký tài khoản | 25 |
| 6 | Hình 4.4.3 Biểu đồ tuần tự nâng cấp tài khoản | 26 |
| 7 | Hình 4.4.4 Biểu đồ tuần tự đẩy thông báo | 27 |
| 8 | Hình 4.4.5 Biểu đồ tuần tự đăng tin tuyển dụng |  |
| 9 | Hình 4.4.6 Biểu đồ tuần tự duyệt ứng viên |  |
| 10 | Hình 4.4.7 Biểu đồ tuần tự gửi email cho ứng viên |  |
| 11 | Hình 4.4.8 Biểu đồ tuần tự tìm kiếm việc làm |  |
| 12 | Hình 4.4.9 Biểu đồ tuần tự ứng tuyển |  |
| 13 | Hình 4.4.10 Biểu đồ tuần tự luyện phỏng vấn |  |
| 14 | Hình 5.1 Màn hình đăng nhập của người dùng (User Login Screen) | 27 |
| 15 | Hình .5.2 Màn hình đăng ký của người dùng (User Register Screen) | 28 |
| 16 | Hình 5.3 Màn hình nhận thông báo của người dùng (User Notification screen) | 28 |
| 17 | Hình 5.4 Màn hình Trang chủ |  |
| 18 | Hình 5.5 Màn hình trang Tuyển dụng |  |
| 19 | Hình 5.6 Màn hình trang Ứng viên |  |
| 20 | Hình 5.7 Màn hình trang Tài khoản |  |
| 21 | Hình 5.8 Màn hình trang chủ Candidate |  |
| 22 | Hình 5.9 Màn hình tìm kiếm và danh sách công ty |  |
| 23 | Hình 5.10 Màn hình chi tiết công việc |  |
| 24 | Hình 5.11 Màn hình trang nhà tuyển dụng(công ty) |  |
| 25 | Hình 5.12 Màn hình trang luyện phỏng vấn |  |
| 26 | Hình 5.13 Một số màn hình khác của ứng viên |  |
| 27 | Hình 5.14 Giao diện dashboard của admin web |  |
| 28 | Hình 5.15 Giao diện quản lý người dùng của admin web |  |
| 29 | Hình 5.16 Giao diện danh sách công ty của admin web |  |
| 30 | Hình 5.17 Giao diện danh sách job của admin web |  |
| 31 | Hình 5.18 Giao diện thanh toán và nâng cấp tài khoản |  |

# Nhiệm vụ chính

## 1. Phát triển Frontend (Mobile Application Development)

### 1.1 Xây dựng Tính năng Giao diện Người dùng

* Ứng dụng Nhà tuyển dụng (Employer App): Phát triển các màn hình chức năng quan trọng như:
  + Trung tâm Thông báo (Notifications Hub): Thiết kế giao diện hiển thị các thông báo quan trọng (ứng viên nộp hồ sơ, tin nhắn mới, nhắc nhở).
  + Quản lý Tài khoản Cao cấp (Upgrade/Subscription Management): Xây dựng luồng mua/nâng cấp tài khoản, hiển thị các gói dịch vụ và quản lý trạng thái thanh toán.
* Ứng dụng Ứng viên (Candidate App): Phát triển các màn hình chức năng quan trọng như:
  + Trung tâm Thông báo (Notifications Hub): Thiết kế giao diện hiển thị các thông báo về trạng thái hồ sơ, phản hồi từ nhà tuyển dụng, gợi ý việc làm phù hợp.
  + Quản lý Tài khoản Cao cấp (Upgrade/Subscription Management): Xây dựng giao diện xem và mua các dịch vụ/tính năng cao cấp (nếu có).

### 1.2. Tối ưu hóa Hiệu suất và Bảo trì Mã nguồn

* Refactoring và Chất lượng Mã: Tối ưu hóa, refactor mã nguồn hiện có để đảm bảo tính tường minh, dễ đọc, dễ hiểu theo các tiêu chuẩn lập trình di động (sử dụng kiến trúc mô hình MVC (Model-View-Controller) rõ ràng).
* Quản lý Tích hợp API: Đảm bảo việc gọi và xử lý dữ liệu từ API được thực hiện mượt mà, trơn tru, có khả năng xử lý lỗi (error handling) và hiển thị trạng thái tải (loading state) hiệu quả, mang lại trải nghiệm tốt nhất cho người dùng.

## 2. Phát triển Backend (Server-Side Development)

### 2.1 Phát triển Hệ thống Lõi và Cơ sở Dữ liệu

* Xây dựng Hệ thống Backend Toàn diện: Phát triển toàn bộ logic nghiệp vụ (Business Logic) và các dịch vụ (Services) phía backend để hỗ trợ tất cả các tính năng của ứng dụng di động.
* Thiết kế và Tối ưu hóa Cơ sở Dữ liệu (Database): Thiết kế mô hình dữ liệu (Database Schema) một cách chặt chẽ và có cấu trúc. Thực hiện các biện pháp tối ưu hóa database (ví dụ: tối ưu hóa truy vấn, đánh chỉ mục – indexing) để đảm bảo tốc độ và khả năng mở rộng (scalability) của hệ thống.

### 2.2 Xây dựng và Quản lý API

* Viết và Cấu hình Controllers/Routes: Phát triển các Controller (hoặc tương đương) với cấu trúc gọn gàng, tuân thủ nguyên tắc lập trình sạch (Clean Code/SOLID), đảm bảo quá trình xử lý logic và truy xuất dữ liệu ở phía backend diễn ra hiệu quả và có tổ chức.
* Thiết kế Bảo mật API và Hiệu năng: Triển khai các biện pháp bảo mật mạnh mẽ cho API (API Security) như:
  + Xác thực và Ủy quyền (Authentication & Authorization): Sử dụng các tiêu chuẩn bảo mật hiện đại (ví dụ: JWT, OAuth 2.0).
  + Giới hạn Tốc độ Yêu cầu (Rate Limiting): Cấu hình Rate Limiting để ngăn chặn lạm dụng và tấn công từ chối dịch vụ (DDoS) cơ bản, đảm bảo tính ổn định của hệ thống.
  + Kiểm tra đầu vào (Input Validation) và phòng chống lỗ hổng bảo mật phổ biến (ví dụ: SQL Injection, XSS).

# 1.Giới thiệu

## 1.1 Bối cảnh Thị trường Tuyển dụng và Kết nối Nhân sự

Thị trường lao động Việt Nam và toàn cầu đang chứng kiến sự dịch chuyển mạnh mẽ từ mô hình tìm kiếm việc làm truyền thống sang nền tảng số hóa. Với sự gia tăng của lực lượng lao động trẻ và nhu cầu tuyển dụng linh hoạt của doanh nghiệp, các nền tảng tuyển dụng trực tuyến đóng vai trò then chốt trong việc tối ưu hóa quy trình. Tuy nhiên, thị trường hiện tại vẫn tồn tại những **khoảng trống** đáng kể:

* **Thời gian phản hồi chậm:** Ứng viên thường phải chờ đợi lâu để nhận được phản hồi về hồ sơ, dẫn đến trải nghiệm tìm việc không hiệu quả.
* **Thiếu tương tác trực tiếp:** Phần lớn các nền tảng chỉ dừng lại ở việc đăng tải và nộp hồ sơ, thiếu kênh giao tiếp tức thời, minh bạch giữa hai bên.
* **Sự phân mảnh của dữ liệu:** Ứng viên phải sử dụng nhiều kênh để theo dõi công việc và quá trình ứng tuyển.

## 1.2 Vấn đề và Nhu cầu Cấp thiết

Vấn đề cốt lõi là sự mất kết nối (Disconnection) giữa ứng viên tiềm năng và nhà tuyển dụng. Nhà tuyển dụng gặp khó khăn trong việc nhanh chóng sàng lọc và tiếp cận nhân tài, trong khi ứng viên mất cơ hội vì quy trình nộp đơn quá dài và thiếu khả năng tương tác để làm rõ thắc mắc. Nhu cầu cấp thiết đặt ra là phải có một giải pháp di động toàn diện, không chỉ là nơi đăng tin mà còn là một "cầu nối" giao tiếp trực tiếp, tức thời.

## 1.3 Giải pháp Job Bridge

Trước bối cảnh đó, ứng dụng di động **Job Bridge** ra đời với sứ mệnh trở thành "cầu nối việc làm" thế hệ mới, giải quyết triệt để vấn đề mất kết nối. Job Bridge tập trung vào việc tạo ra một môi trường làm việc thông suốt và minh bạch:

* **Luyện tập phỏng vấn:** Cung cấp tính năng luyện tập câu hỏi phỏng vấn AI giúp ứng viên có thể luyện tập các câu hỏi liên quan ngành nghề, mức độ giúp ứng viên tăng tự tin khi phỏng vấn
* **Trải nghiệm người dùng di động tối ưu:** Tập trung vào sự tiện lợi, giúp ứng viên nộp hồ sơ chỉ với vài thao tác và nhà tuyển dụng quản lý hồ sơ mọi lúc, mọi nơi.
* **Minh bạch hóa quy trình:** Cập nhật trạng thái ứng tuyển liên tục, giúp cả hai bên theo dõi tiến trình một cách rõ ràng

Ngoài ứng dụng di động chính, giải pháp Job Bridge còn bao gồm trang quản trị (Admin Panel) - một web độc lập dành riêng cho quản trị viên. Trang này cho phép theo dõi số liệu, quản lý người dùng, công ty, tin tuyển dụng và đảm bảo hoạt động hiệu quả của toàn bộ nền tảng.

# 2.Thiết kế và triển khai

## 2.1 Các yêu cầu chức năng

### 2.1.1 Yêu cầu chức năng phía Frontend

Phần này mô tả các chức năng cốt lõi được thiết kế cho ứng dụng di động Job Bridge, nhằm phục vụ nhu cầu của hai nhóm người dùng chính là Ứng viên (Candidates) và Nhà tuyển dụng (Employers). Các chức năng được xây dựng nhằm đảm bảo quy trình tuyển dụng và tìm việc diễn ra thuận tiện, trực quan và kết nối hiệu quả.

Hệ thống được phân chia thành các nhóm chức năng chính như sau:

* 1. Chức năng Dành cho Ứng viên (Candidate Functionalities)

- Login/Sign up: tạo tài khoản dựa trên email của bạn và xác thực qua email, sau đó bạn có thể truy cập vào trang chủ của người tìm việc

- Trung tâm Thông báo: Gửi và hiển thị thông báo tức thì (Push Notifications) về tin nhắn mới, cập nhật trạng thái hồ sơ và gợi ý việc việc làm phù hợp.

-Nâng cấp tài khoản: Thanh toán qua thẻ tiện dụng để nâng cấp tài khoản người dùng. Lúc này có thể truy cập một số tính năng nâng cao như Luyện tập trò chuyện cùng AI

2. Chức năng Dành cho Nhà tuyển dụng (Employer Functionalities)

- Trung tâm Thông báo: Gửi và hiển thị thông báo tức thì (Push Notifications) về tin nhắn mới, cập nhật trạng thái hồ sơ và gợi ý việc làm phù hợp.

- Nâng cấp tài khoản: Nâng cấp tài khoàn để có thể sử dụng chức năng nâng cao như gợi ý ứng viên AI, giúp bạn dễ dàng tìm ra đâu ra ứng viên phù hợp

### 2.1.2 Yêu cầu chức năng phía Backend

Hệ thống Backend Job Bridge chịu trách nhiệm quản lý logic nghiệp vụ và đảm bảo tính bảo mật cho ứng dụng di động. Các chức năng cốt lõi được Backend phải xử lý bao gồm:

1. Quản lý Xác thực & Tài khoản (Authentication & Account)

* Xử lý Đăng ký (Sign Up) và Đăng nhập (Login) an toàn.
* Xác thực Tài khoản qua Email (Email Verification).
* Quản lý Phiên làm việc và Token (JWT).
* Phân quyền người dùng (Ứng viên/Nhà tuyển dụng).

2. Quản lý Thông báo (Notification Management)

* Tạo, lưu trữ và kích hoạt Thông báo Đẩy (Push Notifications) tức thời.
* Xử lý logic Thông báo Tin nhắn mới và Cập nhật Trạng thái Hồ sơ.
* Xử lý logic gợi ý và gửi Thông báo Việc làm/Ứng viên phù hợp.

3. Quản lý Dịch vụ Cao cấp & Thanh toán (Subscription & Billing)

* Giao tiếp và xử lý Giao dịch với Cổng Thanh toán (Payment Gateway).
* Cập nhật và lưu trữ Trạng thái Gói Dịch vụ đã nâng cấp.
* Kiểm tra Quyền truy cập (Authorization Check) đối với các tính năng cao cấp.

4. Hỗ trợ Tính năng AI

* Cung cấp API cho chức năng Luyện tập Trò chuyện cùng AI (cho Ứng viên).
* Cung cấp logic nghiệp vụ cho chức năng Gợi ý Ứng viên AI (cho Nhà tuyển dụng).

## Các yêu cầu phi chức năng

### 2.2.1 Các yêu cầu phi chức năng phía frontend

**a, Yêu cầu về Giao diện (UI/UX Requirements)**

Để đảm bảo ứng dụng di động Job Bridge mang lại trải nghiệm chuyên nghiệp, năng động và thân thiện, giao diện cần đáp ứng các tiêu chuẩn sau:

* **Tính Thẩm mỹ và Thương hiệu:** Giao diện phải có thiết kế hiện đại, năng động và phù hợp với chủ đề kết nối công việc. Tông màu chủ đạo phải là Xanh lá cây (Green) và Trắng (White). Màu xanh lá tượng trưng cho sự phát triển, hy vọng và tươi mới, tạo cảm giác tin cậy và chuyên nghiệp trong môi trường tuyển dụng.
* **Tính Nhất quán**: Các thành phần giao diện (UI Components) như nút bấm hành động (CTA buttons), thanh điều hướng, biểu tượng và phông chữ phải đồng bộ và nhất quán trên toàn bộ ứng dụng, áp dụng cho cả giao diện của Ứng viên và Nhà tuyển dụng. Điều này đảm bảo người dùng có thể dễ dàng chuyển đổi và sử dụng ứng dụng mà không bị bỡ ngỡ.
* **Tính Trực quan và Dễ sử dụng**: Các chức năng cốt lõi phải được đặt ở vị trí dễ nhìn và dễ truy cập.
  + **Đối với Ứng viên**: Thanh tìm kiếm việc làm, tab Hồ sơ cá nhân và Trung tâm thông báo phải luôn hiển thị rõ ràng.
  + **Đối với Nhà tuyển dụng**: Khu vực quản lý Hồ sơ ứng viên và nút Đăng tin tuyển dụng mới phải được sắp xếp logic, đảm bảo quy trình làm việc hiệu quả và nhanh chóng

**b, Yêu cầu về Hiệu năng (Perfomance Requirements)**

Ứng dụng di động Job Bridge phải đáp ứng các tiêu chuẩn hiệu năng nghiêm ngặt để đảm bảo trải nghiệm tìm việc và tuyển dụng diễn ra liên tục, không bị gián đoạn:

* **Thời gian tải**: Tất cả các màn hình chính (Trang chủ, Danh sách việc làm, Hồ sơ cá nhân/Công ty, Hộp thư chat) phải tải hoàn toàn và hiển thị dữ liệu trong vòng dưới 3 giây trên kết nối internet trung bình (4G hoặc Wi-Fi thông thường).
* **Tốc độ Phản hồi API**: Các yêu cầu API quan trọng (Tìm kiếm, Gửi hồ sơ, Gửi tin nhắn) phải có thời gian phản hồi (latency) tối thiểu, lý tưởng là dưới 500ms (mili giây).
* **Dữ liệu Ứng viên/Tin đăng**: Ảnh đại diện, logo công ty và các tài nguyên đa phương tiện khác phải được nén (compressed) hiệu quả và tối ưu hóa cho di động để giảm thiểu kích thước truyền tải, giúp giảm thời gian tải mà vẫn giữ được chất lượng hiển thị.
* **Bộ nhớ Cache**: Ứng dụng phải tận dụng cơ chế bộ nhớ đệm (Caching) ở phía client để lưu trữ dữ liệu tĩnh và giảm số lần gọi API lặp lại.
* **Tương tác UI**: Các hiệu ứng chuyển đổi màn hình (transitions), cuộn (scrolling) danh sách tin tuyển dụng/hồ sơ, và các tương tác chạm (touch interactions) phải hoạt động trơn tru và mượt mà, không gây hiện tượng giật, đứng (lagging), đặc biệt khi ứng dụng chạy trên các thiết bị di động có cấu hình trung bình hoặc thấp hơn.

### 2.2.2 Các yêu cầu phi chức năng liên quan đến Backend

#### **2.2.2.1 Yêu cầu về hiệu năng (xử lý dữ liệu)**

Hiệu năng của backend trong hệ thống thương mại điện tử MERN Stack là một yếu tố quan trọng để đảm bảo tốc độ phản hồi nhanh, khả năng mở rộng và trải nghiệm người dùng mượt mà. Các yêu cầu chính bao gồm:

**a, Thời gian phản hồi (Response Time)**

API phải có thời gian phản hồi trung bình < 200ms cho các truy vấn đọc (GET).

Các thao tác ghi (POST, PUT, DELETE) phải dưới 500ms trong điều kiện tải thông thường.

Đối với các tác vụ phức tạp như xử lý giỏ hàng, thanh toán, thời gian không được vượt quá 1 giây.

**b, Xử lý số lượng yêu cầu đồng thời (Concurrency Handling)**

Hệ thống phải hỗ trợ tối thiểu 1000 request/giây mà không làm chậm hệ thống.

Sử dụng Load Balancer (Nginx, HAProxy) để phân phối tải giữa các máy chủ.

Áp dụng caching (Redis, Memcached) để giảm tải cho cơ sở dữ liệu.

**c, Tối ưu hóa truy vấn cơ sở dữ liệu (Database Query Optimization)**

Indexing: Dữ liệu trong Supabase phải có chỉ mục (Indexes) cho các trường hay truy vấn như userId, orderId, productId.

Pagination & Limit: Khi lấy danh sách sản phẩm, sử dụng limit & skip để tránh truy vấn dữ liệu lớn.

Aggregation Pipelines: Sử dụng Aggregation Framework thay vì truy vấn nhiều lần để cải thiện hiệu suất.

**d, Kiểm soát tải và mở rộng hệ thống (Scalability & Load Management)**

Hỗ trợ Scale Horizontally bằng cách chạy nhiều instance Node.js với PM2 hoặc Docker.

Cơ sở dữ liệu Supabase có thể mở rộng theo chiều ngang (Sharding) khi số lượng dữ liệu lớn.

Sử dụng CDN (Cloudflare, AWS CloudFront) để giảm tải cho backend khi phân phối dữ liệu tĩnh.

**e, Quản lý bộ nhớ và tài nguyên (Memory & Resource Management)**

Hạn chế rò rỉ bộ nhớ bằng cách tối ưu Garbage Collection (GC) trong Node.js.

Giới hạn kích thước tải lên (upload limit) để tránh tắc nghẽn server.

Dùng worker threads hoặc message queues (RabbitMQ, Kafka) để xử lý các tác vụ nền như gửi email, cập nhật đơn hàng

#### **2.2.2.2 Yêu cầu về bảo mật**

Bảo mật backend là một trong những yếu tố quan trọng khi xây dựng hệ thống thương mại điện tử sử dụng SERN Stack. Một hệ thống bảo mật tốt giúp bảo vệ dữ liệu khách hàng, giao dịch tài chính, tránh các cuộc tấn công mạng và đảm bảo tính toàn vẹn của hệ thống.

**a, Bảo mật xác thực và ủy quyền (Authentication & Authorization)**

* + - * + Sử dụng JWT (JSON Web Token) cho cơ chế xác thực, có thời gian hết hạn hợp lý (ví dụ: 15-30 phút cho token truy cập, 7 ngày cho refresh token).
        + Mã hóa mật khẩu bằng bcrypt trước khi lưu vào database (không

bao giờ lưu mật khẩu dưới dạng plaintext).

* + - * + Áp dụng OAuth2.0 nếu cho phép đăng nhập qua bên thứ ba như Google, Facebook.
        + Sử dụng Role-Based Access Control (RBAC) để phân quyền người dùng (Admin, User, Seller…).

***b,* Bảo mật dữ liệu và cơ sở dữ liệu (Database Security)**

* + - * + Không bao giờ để lộ thông tin kết nối database trong code,thay vào đó lưu trong biến môi trường .env.
        + Hạn chế quyền truy cập cơ sở dữ liệu, chỉ cấp quyền cần thiết (ví dụ: tài khoản API chỉ có quyền đọc dữ liệu).
        + Sử dụng SSL/TLS để mã hóa kết nối giữa backend và database, đặc biệt khi sử dụng Supabase.
        + Sử dụng NoSQL Injection Prevention: Kiểm tra kỹ dữ liệu đầu vào để tránh truy vấn NoSQL Injection.

**c, Ngăn chặn các cuộc tấn công phổ biến**

* + SQL/NoSQL Injection

* + - Không bao giờ chèn dữ liệu đầu vào trực tiếp vào truy vấn mà phải sử dụng các phương pháp bảo vệ như Supabase Policies

* + - Kiểm tra và validate dữ liệu người dùng trước khi thực hiện truy vấn. Cross-Site Scripting (XSS)
    - Sanitize dữ liệu đầu vào để tránh kẻ tấn công chèn mã độc vào các trường như tên, bình luận, mô tả sản phẩm.
    - Dùng thư viện như xss-clean để lọc dữ liệu độc hại.

* + Cross-Site Request Forgery (CSRF)

* + - Dùng CSRF token (thư viện csurf trong Express) để xác thực các request quan trọng như thanh toán, thay đổi mật khẩu.

* + Brute Force Attack
    - Giới hạn số lần đăng nhập thất bại (dùng express-rate-limit).

* + - Sử dụng CAPTCHA khi phát hiện hoạt động đăng nhập bất thường.

***d,* Bảo vệ API và Giao Tiếp Giữa Các Thành Phần**

* + - * + Sử dụng HTTPS thay vì HTTP để mã hóa toàn bộ dữ liệu truyền tải.

* + - * + Chỉ cho phép CORS (Cross-Origin Resource Sharing) đối với các domain tin cậy.

* + - * + Bảo vệ API với API Gateway (AWS API Gateway, Nginx, Kong API Gateway) để kiểm soát truy cập.

* + - * + Sử dụng Web Application Firewall (WAF) để ngăn chặn các cuộc tấn công tự động.

**e, Bảo vệ giao dịch thanh toán**

* + - * + Không lưu trữ thông tin thẻ tín dụng trên hệ thống – thay vào đó sử dụng bên thứ ba như Stripe, PayPal, VNPay.

* + - * + Mọi request thanh toán cần có xác thực hai yếu tố (2FA) hoặc OTP để tăng độ an toàn.

* + - * + Sử dụng Webhook bảo mật để xử lý giao dịch thay vì tin vào dữ liệu từ frontend

**g, Kiểm soát phiên và bảo vệ tài khoản người dùng**

* + - * + Hạn chế session hijacking bằng cách đặt Secure & HttpOnly Cookie.
        + Tự động đăng xuất người dùng sau một khoảng thời gian không hoạt động.
        + Yêu cầu xác thực lại (re-authentication) trước khi thực hiện các hành động nhạy cảm như đổi mật khẩu, rút tiền.

**h, Giám sát và phát hiện xâm nhập**

* + - * + Ghi log tất cả các hoạt động quan trọng (đăng nhập, thay đổi thông tin, thanh toán).
        + Dùng hệ thống giám sát bảo mật như Wazuh, Fail2Ban, hoặc CloudTrail để phát hiện tấn công.
        + Thông báo cho admin khi phát hiện hành vi đáng ngờ (ví dụ: nhiều lần đăng nhập thất bại, giao dịch bất thường).

# 3.Các ràng buộc (Constraints)

## 3.1 Ràng buộc về thời gian

* **Thời gian phát triển**: Hệ thống cần được hoàn thiện trong một khung thời gian nhất định để đáp ứng nhu cầu thị trường hoặc tận dụng các cơ hội tuyển dụng (ví dụ: ra mắt trước cuối năm vì cuối năm có nhiều việc cần tuyển).
* **Thời gian phản hồi của hệ thống:** App cần đảm bảo tốc độ tải trang nhanh

(dưới 3 giây) để tránh mất khách hàng do trải nghiệm người dùng kém.

* **Thời gian cập nhật:** Các tính năng mới hoặc sửa lỗi cần được triển khai nhanh chóng để duy trì tính cạnh tranh và đáp ứng phản hồi từ người dùng.

## 3.2 Ràng buộc công nghệ

* **Tương thích đa nền tảng**: Hệ thống phải hoạt động tốt trên nhiều hệ điều hành như Android, IOS,… để có thể đa dạng người dùng.
* **Hạn chế công nghệ hiện có**: Nếu sử dụng SERN Stack (Supabase, Express.js, React Native, Node.js), hệ thống bị giới hạn bởi các đặc điểm của công nghệ này, như khả năng xử lý đồng thời hoặc yêu cầu tối ưu hóa cho các truy vấn phức tạp.
* **Phụ thuộc vào bên thứ ba**: Hệ thống có thể bị ảnh hưởng bởi độ tin cậy của các dịch vụ bên ngoài (API thanh toán, vận chuyển, xác thực), dẫn đến rủi ro nếu các dịch vụ này gặp sự cố.

## 3.3 Ràng buộc về nguồn lực

* **Nhân sự**: Số lượng và kỹ năng của đội ngũ phát triển, vận hành, và hỗ trợ khách hàng có thể giới hạn khả năng xây dựng và duy trì hệ thống.
* **Ngân sách**: Tổng chi phí cho dự án (bao gồm phát triển, vận hành, marketing) phải nằm trong giới hạn tài chính của doanh nghiệp, ảnh hưởng đến quy mô và chất lượng của giải pháp.
* **Dữ liệu ban đầu**: Hệ thống cần dữ liệu các công việc, khách hàng, và nhà tuyển dụng để

khởi động, nhưng việc thu thập và nhập liệu ban đầu có thể gặp khó khăn nếu không có nguồn sẵn có.

## 3.4 Ràng buộc về người dùng

* **Trải nghiệm người dùng (UX)**: Hệ thống phải đơn giản, dễ sử dụng cho cả khách hàng không rành công nghệ, đồng thời cung cấp các tính năng nâng cao cho người dùng chuyên nghiệp.
* **Số lượng người dùng tối đa**: Hệ thống cần được thiết kế để xử lý một lượng truy

cập nhất định (ví dụ: 10.000 người dùng đồng thời) mà không bị quá tải.

* **Đa dạng ngôn ngữ và văn hóa**: Nếu nhắm đến thị trường quốc tế, hệ thống phải hỗ trợ nhiều ngôn ngữ và phù hợp với thói quen mua sắm của từng khu vực.

## 3.5 Ràng buộc về cạnh tranh

* **Đối thủ cạnh tranh**: Hệ thống cần có các tính năng nổi bật hoặc ưu điểm vượt trội so với các nền tảng ứng dụng khác (TopCV, Linkedin) để thu hút người dùng.
* **Thị hiếu thị trường**: Công việc và mức offer phải phù hợp với xu hướng tìm việc

hiện tại, đòi hỏi hệ thống phải linh hoạt để thay đổi danh mục hoặc chiến lược kinh doanh.

* **Thời gian tồn tại**: Trong môi trường cạnh tranh cao, hệ thống cần duy trì sự hấp

dẫn lâu dài để tránh bị lãng quên hoặc thay thế bởi các đối thủ mới.

## Các ràng buộc về triển khai

### 3.6.1 Các ràng buộc kinh tế

#### **3.6.1.1Ràng buộc về chi phí phát triển**

* + Chi phí thuê lập trình viên: Nếu bạn không tự code, việc thuê lập trình viên MERN Stack có thể tốn kém.
  + Chi phí thiết kế UI/UX: Website thương mại điện tử cần giao diện chuyên nghiệp, điều này có thể phát sinh chi phí cho designer.
  + Chi phí tích hợp API: Một số dịch vụ như thanh toán trực tuyến (Stripe, PayPal, VNPay), vận chuyển, hoặc xác thực người dùng có thể yêu cầu phí sử dụng.
  + Chi phí phần mềm và công cụ: Nếu sử dụng các dịch vụ như hosting cao
  + cấp, CDN, hoặc hệ thống kiểm thử, chi phí có thể tăng.

#### **3.6.1.2 Ràng buộc về hạ tầng và vận hành**

* + Chi phí lưu trữ và cơ sở dữ liệu:
    - Nếu sử dụng Supabase bạn có thể phải trả tiền cho các gói nâng cao khi dữ liệu tăng.
    - Nếu tự triển khai Supabase trên VPS hoặc server riêng, cần tính đến chi phí bảo trì.
  + Chi phí máy chủ (Hosting/VPS):
    - Dùng Heroku, Vercel, Render có thể miễn phí ở mức cơ bản, nhưng hiệu suất có giới hạn.
    - Nếu chọn AWS, DigitalOcean, Linode thì có phí duy trì theo tài nguyên sử dụng.
    - Băng thông và tốc độ tải trang: Nếu trang có nhiều hình ảnh hoặc video sản phẩm, cần tối ưu để tránh tốn chi phí băng thông.

#### **3.6.1.3 Ràng buộc về bảo mật và tuân thủ pháp lý**

* Chi phí bảo mật dữ liệu:
  + Phải đầu tư vào chứng chỉ SSL, bảo vệ API và mã hóa thông tin khách hàng.
  + Bảo mật token đăng nhập (JWT, OAuth) để tránh rủi ro bị hack.
* Tuân thủ quy định pháp lý:
  + Nếu tuyển dụng, phải tuân thủ Visa .
  + Nếu hoạt động tại Việt Nam, phải đăng ký giấy phép kinh doanh và tuân thủ các quy định về công ty.
* Chi phí xử lý gian lận:
  + Cần hệ thống phát hiện giao dịch bất thường, tránh bị chargeback hoặc gian lận thanh toán.

#### **3.6.1.4 Ràng buộc về marketing và tăng trưởng**

* Chi phí quảng cáo và SEO:
  + Chạy quảng cáo trên Google Ads, Facebook Ads, TikTok Ads tốn kém nếu không có chiến lược tối ưu.
  + SEO cần đầu tư vào content chất lượng, nghiên cứu từ khóa và backlink.
* Chi phí chăm sóc khách hàng:
  + Nếu có live chat hoặc chatbot, cần duy trì server hoặc tích hợp dịch vụ như Tidio, Intercom.
  + Nếu thuê nhân viên CSKH, chi phí có thể tăng theo số lượng khách hàng.
* Chi phí duy trì và nâng cấp:
  + Một app tuyên dụng không chỉ dừng lại ở việc phát triển ban đầu mà còn cần cập nhật tính năng, sửa lỗi và cải thiện hiệu suất.

#### **3.6.1.5 Ràng buộc về quy mô và mở rộng**

 **Khả năng Mở rộng Hệ thống (Scalability):** Hệ thống Backend phải được thiết kế theo kiến trúc phi tập trung (Microservices hoặc tương đương) để dễ dàng mở rộng theo chiều ngang (Horizontal Scaling). Cụ thể, khi **lượng người dùng (Ứng viên và Nhà tuyển dụng) tăng đột biến**, hệ thống phải có khả năng tự động điều chỉnh tài nguyên (ví dụ: Auto-Scaling các Server API, Database Read Replicas) mà **không gây gián đoạn** dịch vụ hoặc ảnh hưởng đến hiệu năng truy cập của người dùng hiện tại.

 **Tính Sẵn sàng và Độ Ổn định (Availability & Reliability):** Ứng dụng phải duy trì **thời gian hoạt động (Uptime)** cao (ví dụ: 99.9% trở lên). Hệ thống cần có cơ chế dự phòng (Failover/Redundancy) cho các dịch vụ quan trọng (Xác thực, Chat, Database) để đảm bảo Job Bridge luôn sẵn sàng phục vụ người dùng 24/7.

### 3.6.2 Các ràng buộc về đạo đức

#### **3.6.2.1 Bảo vệ quyền riêng tư và dữ liệu khách hàng**

* Thu thập và xử lý dữ liệu minh bạch:
  + Chỉ thu thập thông tin cần thiết, không lạm dụng dữ liệu cá nhân.
  + Cung cấp chính sách bảo mật rõ ràng để khách hàng biết dữ liệu của họ được sử dụng như thế nào.
* Bảo mật thông tin người dùng:
  + Mã hóa thông tin nhạy cảm (mật khẩu, thông tin thanh toán) để tránh bị đánh cắp.
  + Không chia sẻ dữ liệu khách hàng với bên thứ ba mà không có sự đồng ý.
* Quyền được xóa dữ liệu:
  + Cho phép khách hàng yêu cầu xóa tài khoản và dữ liệu cá nhân khi không còn sử dụng dịch vụ.
  + Ví dụ sai phạm: Facebook từng bị chỉ trích vì vụ Cambridge Analytica, nơi dữ liệu người dùng bị lạm dụng cho mục đích chính trị mà không có sự đồng ý.

#### **3.6.2.2 Minh bạch trong kinh doanh và quảng cáo**

* **Chất lượng và Trung thực của Thông tin (Chống thông tin sai lệch):**
  + - * **Tin đăng trung thực:** Các Nhà tuyển dụng phải cung cấp mô tả công việc, yêu cầu kinh nghiệm, và mức lương **trung thực**, không thổi phồng cơ hội nghề nghiệp hoặc đưa ra các thông tin gây hiểu lầm cho Ứng viên.
      * **Hồ sơ xác thực:** Ứng viên phải đảm bảo thông tin cá nhân và kinh nghiệm trong hồ sơ là **đúng thực tế**, không giả mạo bằng cấp hay kinh nghiệm làm việc.
* **Minh bạch về Quá trình Tuyển dụng:**
  + - * **Trạng thái rõ ràng:** Hệ thống phải đảm bảo hiển thị **trạng thái ứng tuyển** một cách minh bạch, tránh gây cảm giác chờ đợi vô vọng cho Ứng viên.
      * **Chính sách giá dịch vụ rõ ràng:** Đối với các gói Nâng cấp tài khoản (cho cả hai bên), **tổng chi phí** và **lợi ích đi kèm** phải được hiển thị rõ ràng, tránh các chiêu trò nâng giá rồi giảm giá ảo để tạo cảm giác khuyến mãi không có thật.
* **Chống Spam và Lạm dụng Nền tảng:**
  + - * **Kiểm soát giao tiếp:** Cấm Nhà tuyển dụng sử dụng nền tảng để **spam** tin nhắn hoặc email quảng cáo quá mức, không liên quan đến cơ hội việc làm.
      * **Chống lừa đảo:** Triển khai các cơ chế kiểm duyệt và báo cáo để ngăn chặn các chiêu trò **lừa đảo việc làm** (Scam job postings) hoặc các hành vi "clickbait" (sử dụng tiêu đề gây hiểu lầm để thu hút hồ sơ) trên nền tảng.
      * Ví dụ sai phạm: Một số trang thương mại điện tử tạo đơn hàng ảo để đánh giá 5 sao hoặc cố tình ẩn thông tin phí vận chuyển, khiến khách hàng hiểu lầm.

#### ***3.6.2.3 Trách nhiệm với xã hội và cộng đồng***

Job Bridge cam kết vận hành nền tảng một cách có trách nhiệm, đóng góp tích cực vào thị trường lao động và đảm bảo môi trường tương tác an toàn, công bằng cho tất cả người dùng.

* **Hoạt động Hợp pháp và Tuân thủ Đạo đức:**
  + - * **Chống tin tuyển dụng vi phạm:** Tuyệt đối cấm đăng tải các tin tuyển dụng liên quan đến hoạt động **bất hợp pháp, lừa đảo, hoặc vi phạm nghiêm trọng thuần phong mỹ tục** (ví dụ: cấm phân biệt chủng tộc, giới tính, tôn giáo trong yêu cầu tuyển dụng).
      * **Tuân thủ pháp luật lao động:** Hệ thống phải có cơ chế kiểm duyệt sơ bộ để ngăn chặn các tin đăng có dấu hiệu vi phạm luật lao động hiện hành (ví dụ: tuyển dụng lao động trẻ em, yêu cầu quá mức về điều kiện cá nhân).
* **Hỗ trợ Người dùng và Giải quyết Khiếu nại Công bằng:**
  + - * **Chính sách sử dụng rõ ràng:** Cung cấp các điều khoản và chính sách sử dụng (Terms of Service) rõ ràng cho cả Ứng viên và Nhà tuyển dụng, đặc biệt là quy định về bảo mật dữ liệu và giải quyết tranh chấp.
      * **Xử lý khiếu nại công bằng:** Thiết lập một quy trình xử lý khiếu nại (ví dụ: báo cáo tin đăng lừa đảo, báo cáo hành vi quấy rối) một cách **nhanh chóng và công bằng**, tránh lạm dụng điều khoản để gây bất lợi cho người dùng (Ứng viên bị từ chối không rõ lý do, Nhà tuyển dụng bị đánh giá sai).
* **Đóng góp Cộng đồng và Phát triển Bền vững:**
  + - * **Hỗ trợ việc làm bền vững:** Khuyến khích các Nhà tuyển dụng đăng tin về các công việc có trách nhiệm xã hội, thân thiện với môi trường (Green Jobs).
      * **Cơ hội giáo dục:** Cung cấp miễn phí các tài nguyên/tính năng cơ bản cho sinh viên hoặc người lao động mới bắt đầu, giúp họ nâng cao kỹ năng tìm việc và hòa nhập thị trường lao động.
* **Ví dụ Sai phạm Cần Tránh:** Một số nền tảng tuyển dụng không kiểm soát chặt chẽ, dẫn đến tình trạng các công ty ma, lừa đảo đăng tin tuyển dụng giả mạo, gây mất thời gian và rủi ro cho Ứng viên. Job Bridge phải ngăn chặn triệt để tình trạng này.

#### **3.6.2.4 Trách nhiệm với nhân viên và đối tác**

* Đối xử công bằng với nhân viên:
  + - * Tránh bóc lột lao động, đảm bảo chế độ lương thưởng hợp lý.
      * Nếu có đội ngũ hỗ trợ khách hàng, cần đảm bảo họ được làm việc trong môi trường tốt.
* Hợp tác công bằng với nhà cung cấp:
  + - * Thanh toán đúng hạn, không ép giá bất hợp lý.
      * Tránh việc ưu tiên hiển thị sản phẩm của một số nhà cung cấp nhất định nếu không có lý do minh bạch.
      * Ví dụ sai phạm: Một số sàn thương mại điện tử lớn bị chỉ trích vì ưu tiên sản phẩm của chính họ, khiến các nhà bán hàng nhỏ lẻ gặp khó khăn.

#### **3.6.2.5 Kiểm soát nội dung và bảo vệ người dùng**

* **Ngăn chặn Nội dung Độc hại và Lạm dụng:**
  + - * **Kiểm duyệt Giao tiếp:** Cần kiểm duyệt và chặn các tin nhắn, bình luận hoặc phản hồi giữa Nhà tuyển dụng và Ứng viên mang tính **kích động, thù địch, quấy rối (harassment)** hoặc sử dụng ngôn ngữ không phù hợp.
      * **Chính sách Kiểm duyệt:** Áp dụng chính sách kiểm duyệt rõ ràng cho tin đăng, hồ sơ và mọi hình thức tương tác công khai/riêng tư để duy trì môi trường giao tiếp chuyên nghiệp.
      * **Cơ chế Báo cáo:** Cung cấp chức năng cho người dùng dễ dàng **báo cáo** các tin đăng, hồ sơ hoặc hành vi nhắn tin vi phạm.
* **Bảo vệ Dữ liệu và Người dùng Dễ tổn thương:**
  + - * **Bảo vệ Thông tin Cá nhân:** Đảm bảo dữ liệu cá nhân của Ứng viên và Nhà tuyển dụng được bảo mật tuyệt đối, tuân thủ các quy tắc về quyền riêng tư.
      * **Kiểm soát Tuyển dụng không phù hợp:** Ngăn chặn các tin tuyển dụng nhắm mục tiêu vào những nhóm người dùng dễ bị tổn thương, đặc biệt là các tin đăng yêu cầu lao động **dưới tuổi vị thành niên** hoặc yêu cầu các điều kiện lao động phi pháp, thiếu đạo đức.
      * **Xác thực độ tuổi/giới hạn:** Thiết lập các biện pháp xác thực cơ bản để đảm bảo người dùng tham gia tuyển dụng/tìm việc phù hợp với quy định pháp luật về tuổi lao động.

# 4.Mô hình hệ thống / Thiết kế giải pháp

## 4.1Các kịch bản của hệ thống (Use-cases)

### 4.1.1 Kịch bản dành cho người tìm việc (Candidates)

#### **a, Đăng nhập vào hệ thống**

* Mục tiêu: Người dùng truy cập tài khoản cá nhân để tìm việc
* Điều kiện tiên quyết: Người dùng đã có tài khoản đăng ký.Các bước:
* Người dùng truy cập screen đăng nhập trên app.
* Nhập email và mật khẩu vào các trường tương ứng. Sau đó chọn Role
* Nhấn nút "Đăng nhập".
* Hệ thống kiểm tra thông tin trong cơ sở dữ liệu.
* Nếu thông tin đúng, người dùng được chuyển hướng đến trang chính. Nếu sai, hiển thị thông báo lỗi "Email hoặc mật khẩu không chính xác".
* Kết quả: Người dùng đăng nhập thành công và có thể sử dụng các chức năng cá nhân hóa.

#### **b, Đăng ký tài khoản mới**

* Mục tiêu: Người dùng tạo tài khoản để bắt đầu tìm kiếm công việc
* Điều kiện tiên quyết: Người dùng chưa có tài khoản.
* Các bước:
  + Người dùng nhấp vào liên kết "Đăng ký" trên trang chủ.
  + Điền thông tin (email, mật khẩu,username,role) vào biểu mẫu.
  + Nhấn nút "Đăng ký".
  + Hệ thống kiểm tra email có trùng lặp không. Nếu không, gửi email xác nhận chứa liên kết kích hoạt.
  + Người dùng nhấp vào liên kết trong email để kích hoạt tài khoản.
  + Kết quả: Tài khoản được tạo và người dùng có thể đăng nhập.

#### **c, Khôi phục mật khẩu**

* Mục tiêu: Người dùng lấy lại quyền truy cập khi quên mật khẩu.
* Điều kiện tiên quyết: Người dùng đã có tài khoản nhưng quên mật khẩu.
* Các bước:
  + Người dùng nhấp vào "Quên mật khẩu" trên trang đăng nhập.
  + Nhập email đã đăng ký và nhấn "Gửi".
  + Hệ thống gửi email chứa mã OTP hoặc liên kết đặt lại mật khẩu.
  + Người dùng nhập mã OTP hoặc nhấp liên kết, sau đó tạo mật khẩu mới.
  + Kết quả: Mật khẩu được cập nhật và người dùng có thể đăng nhập lại.

#### **d.Nâng cấp tài khoản**

* Mục tiêu: Người dùng thanh toán để nâng cấp tài khoản lên VIP
* Điều kiện tiên quyết: Người dùng đã có tài khoản và thẻ tín dụng
* Các bước
  + Người dùng nhấn ô nâng cấp tài khoản
  + Chọn thẻ tín dụng và ngân hàng
  + Nhập đầy đủ thông tin cần thiết và ấn thanh toán
  + Kết quả: Tài khoản có quyền truy cập các tính năng cao cấp hơn

#### **e. Notification**

* + Mục tiêu: Nhận thông báo tự động
  + Các bước:
    - Nhấn vào tab notification
    - Kết quả:Nhận các thông báo như người tuyển dụng đăng công việc hay người tuyển dụng đã xem cv…

#### **f. Quản lý thông tin nghề nghiệp (CV)**

* + Mục tiêu: Ứng viên tải lên, cập nhật và quản lý CV của mình.
  + Điều kiện tiên quyết: Ứng viên đã đăng nhập vào hệ thống.
  + Các bước:
    - Ứng viên truy cập tab “Thông tin nghề nghiệp” trong trang hồ sơ cá nhân.
    - Chọn “Tải CV lên” và chọn file từ thiết bị (định dạng PDF, DOCX,...).
    - Hệ thống kiểm tra tính hợp lệ của file (dung lượng, định dạng, virus).
    - Nếu hợp lệ, hệ thống lưu CV vào cơ sở dữ liệu.
    - Ứng viên có thể chọn “Xem CV” hoặc “Cập nhật CV mới”.
    - **Kết quả:** CV được lưu trữ thành công và được sử dụng cho các lần ứng tuyển sau.

#### **g. Tìm kiếm và khám phá việc làm**

* + Mục tiêu: Ứng viên tìm kiếm các công việc phù hợp với kỹ năng và sở thích.
  + Điều kiện tiên quyết: Ứng viên đã đăng nhập và hoàn thiện hồ sơ cá nhân.
  + Các bước:
    - Ứng viên truy cập mục “Tìm việc”.
    - Nhập từ khóa theo **tên công việc**, **công ty**, hoặc **địa điểm**.
    - Hệ thống hiển thị danh sách kết quả tương ứng.
    - Ứng viên có thể lọc theo mức lương, loại hình làm việc hoặc cấp bậc.
    - Hệ thống AI tự động gợi ý việc làm phù hợp dựa trên nội dung CV và lịch sử tìm kiếm.
    - **Kết quả:** Ứng viên khám phá và chọn được công việc phù hợp để ứng tuyển.

#### **h. Ứng tuyển và theo dõi trạng thái**

* + Mục tiêu: Ứng viên nộp đơn cho công việc mong muốn và theo dõi kết quả ứng tuyển.
  + Điều kiện tiên quyết: Ứng viên đã có CV hợp lệ và đăng nhập vào hệ thống.
  + Các bước:
    - Ứng viên truy cập trang chi tiết công việc mong muốn.
    - Nhấn nút “Ứng tuyển ngay”.  
      Hệ thống xác nhận CV sẽ sử dụng (CV đã tải lên trước đó).
    - Hệ thống gửi hồ sơ đến nhà tuyển dụng và ghi nhận trạng thái ứng tuyển.
    - Ứng viên vào mục “Công việc đã ứng tuyển” trong hồ sơ cá nhân để theo dõi tiến trình.
    - **Kết quả:** Hồ sơ được gửi đi thành công; Ứng viên có thể xem trạng thái (Đang xem / Đã duyệt / Từ chối).

#### **i. Luyện phỏng vấn**

* + Mục tiêu: Ứng viên luyện tập kỹ năng phỏng vấn và nhận đánh giá từ AI.
  + Điều kiện tiên quyết: Ứng viên đã đăng nhập và có thông tin nghề nghiệp trong hệ thống.
  + Các bước:
    - Ứng viên truy cập mục “Luyện phỏng vấn”.
    - Chọn cấp độ phỏng vấn (Intern, Junior, Senior, v.v.).
    - Chọn nguồn câu hỏi: từ **ngân hàng câu hỏi** hoặc **AI tạo mới**.
    - Ứng viên trả lời câu hỏi bằng giọng nói, hệ thống ghi âm và xử lý âm thanh.
    - AI phân tích nội dung, giọng nói, độ tự tin → chấm điểm và hiển thị nhận xét.
    - **Kết quả:** Ứng viên nhận được đánh giá chi tiết, điểm số.

### 4.1.2 Kịch bản dành cho nhà tuyển dụng (Employer)

#### **a, Đăng nhập vào trang nhà tuyển dụng**

* Mục tiêu: Nhà tuyển dụng truy cập vào trang người tuyển dụng
* Điều kiện tiên quyết: Nhà tuyển dụng có tài khoản của nhà tuyển dụng
* Các bước**:**
  + - * Người tuyển dụng truy cập trang đăng nhập
      * Nhập thông tin tài khoản và chọn role nhà tuyển dụng
      * Hệ thống xác thực role và chuyển hướng đến trang tuyển dụng
      * Kết quả**:** Nhà tuyển dụng vào được tuyển dụng để quản lý công việc

#### **b, Đăng ký tài khoản mới**

* Mục tiêu: Người tuyển dụng tạo tài khoản để bắt đầu tìm kiếm công việc
* Điều kiện tiên quyết: Người dùng chưa có tài khoản.
* Các bước:
  + Người dùng nhấp vào liên kết "Đăng ký" trên trang chủ.
  + Điền thông tin (email, mật khẩu,username,role) vào biểu mẫu.
  + Nhấn nút "Đăng ký".
  + Hệ thống kiểm tra email có trùng lặp không. Nếu không, gửi email xác nhận chứa liên kết kích hoạt.
  + Người dùng nhấp vào liên kết trong email để kích hoạt tài khoản.
  + Kết quả: Tài khoản được tạo và người dùng có thể đăng nhập.

#### **c, Khôi phục mật khẩu**

* Mục tiêu: Người dùng lấy lại quyền truy cập khi quên mật khẩu.
* Điều kiện tiên quyết: Người dùng đã có tài khoản nhưng quên mật khẩu.
* Các bước:
  + Người dùng nhấp vào "Quên mật khẩu" trên trang đăng nhập.
  + Nhập email đã đăng ký và nhấn "Gửi".
  + Hệ thống gửi email chứa mã OTP hoặc liên kết đặt lại mật khẩu.
  + Người dùng nhập mã OTP hoặc nhấp liên kết, sau đó tạo mật khẩu mới.
  + Kết quả: Mật khẩu được cập nhật và người dùng có thể đăng nhập lại.

#### **d.Nâng cấp tài khoản**

* Mục tiêu: Người dùng thanh toán để nâng cấp tài khoản lên VIP
* Điều kiện tiên quyết: Người dùng đã có tài khoản và thẻ tín dụng
* Các bước
  + Người dùng nhấn ô nâng cấp tài khoản
  + Chọn thẻ tín dụng và ngân hàng
  + Nhập đầy đủ thông tin cần thiết và ấn thanh toán
  + Kết quả: Tài khoản có quyền truy cập các tính năng cao cấp hơn

#### **e. Notification**

* + Mục tiêu: Nhận thông báo tự động
  + Các bước:
    - Nhấn vào tab notification
    - Kết quả:Nhận các thông báo như người ứng viên đã ứng tuyển hay ứng viên đã lưu thông tin công việc…

### 4.1.3 Kịch bản dành cho trang admin

#### **a, Xem dashboard tổng quan**

* Mục tiêu: Hiển thị số liệu chung và biểu đồ.
* Các bước**:** vào Dashboard → component dùng useQuery gọi getDashboardStats() và getAnalyticsData() → dữ liệu trả về hiển thị ở các <Statistic> và chart.

#### **b, Xem danh sách Users (liệt kê, tìm kiếm, lọc, phân trang)**

* Mục tiêu: Quản lý danh sách người dùng.
* Các bước: Các bước: mở Users → component gọi getUsers(filters) (React Query) → service query Supabase với .select(..., {count:'exact'}).range(...) → kết quả hiển thị trong Ant Table (pageSize 10/20, search, filter).

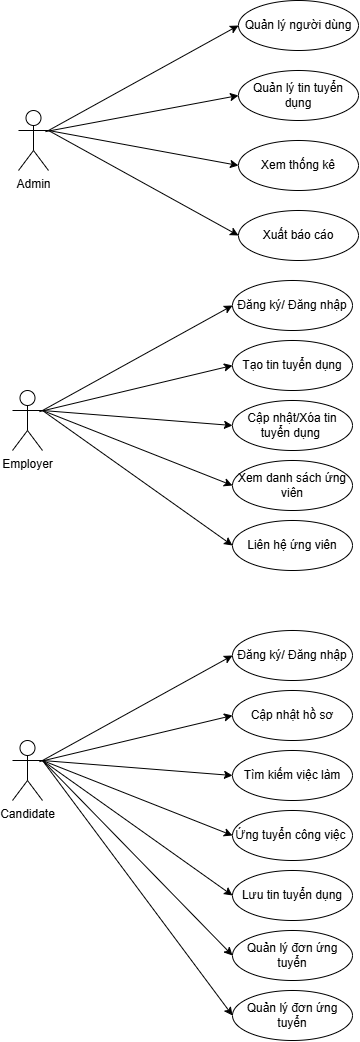
#### **c, Quản lý Companies (liệt kê, duyệt/từ chối)**

* Mục tiêu: Kiểm duyệt hồ sơ doanh nghiệp và thống kê công ty.
* Các bước: mở Companies → getCompanies() lấy danh sách + trạng thái → admin chọn Approve/Reject → gọi API update status → refetch danh sách.

#### **d, Quản lý Jobs (liệt kê, tìm kiếm, thao tác)**

* Mục tiêu: Quản lý các bài đăng tin tuyển dụng.
* Điều kiện tiên quyết: Người dùng đã có tài khoản và thẻ tín dụng
* Các bước: vào Jobs → component gọi getJobs(filters) → hiển thị bảng jobs (title, company, salary, created\_at, ứng tuyển). Thực hiện action (edit/delete/toggle) gọi service tương ứng và refetch.

## 4.2 Mô hình Use-case

Mô hình Use case tổng quát:

Hình 4.2.1 Mô hình Use case tổng quát

### 4.2.1Mô hình Use-case liên quan đến Frontend

#### **4.2.1.1 Use-case đăng nhập bằng email**

| **Thuộc tính** | **Chi tiết** |
| --- | --- |
| **Usecase Id** | UC\_001 |
| **Usecase Name** | Đăng nhập bằng Email (Email Login) |
| **Description** | Người dùng truy cập hệ thống bằng cách nhập Email và Mật khẩu đã đăng ký. |
| **Actor** | Người dùng |
| **Priority** | Cao |
| **Trigger** | Người dùng chọn tùy chọn “Đăng nhập” trên màn hình khởi động ứng dụng. |
| **Pre-condition** | Người dùng đã có tài khoản trên hệ thống Job Bridge. |
| **Post-condition** | Người dùng được xác thực thành công và truy cập vào Trang chủ (Homepage) tương ứng với vai trò của họ. |
| **Basic Flow** | 1. Người dùng chọn vai trò đăng nhập ["Nhà tuyển dụng" \*\* |
|  | 2. Người dùng nhập thông tin: Email và Mật khẩu. |
|  | 3. Hệ thống kiểm tra tính hợp lệ của dữ liệu (định dạng Email). |
|  | 4. Hệ thống kiểm tra Email có tồn tại trong cơ sở dữ liệu và **kiểm tra Mật khẩu** có khớp với thông tin lưu trữ hay không. |
|  | 5. Hệ thống kiểm tra **Vai trò đã chọn** (bước 1) có khớp với vai trò đã đăng ký của Email đó hay không. |
|  | 6. Nếu thông tin đăng nhập hợp lệ và Vai trò khớp, hệ thống tạo token phiên làm việc (session token). |
|  | 7. Hệ thống chuyển hướng Người dùng đến Trang chủ (Homepage) tương ứng với vai trò của họ. |
| **Alternative Flow** | **A. Sai thông tin Đăng nhập (Email/Mật khẩu)** |
|  | 1. Trong bước 4, Hệ thống phát hiện Email chưa tồn tại **hoặc** Mật khẩu không chính xác. |
|  | 2. Hệ thống hiển thị thông báo **"Sai Email hoặc Mật khẩu. Vui lòng kiểm tra lại."** (Tránh nói rõ Email hay Mật khẩu sai để tăng bảo mật). |
|  | 3. Người dùng nhập lại thông tin đăng nhập. |
| **Alternative Flow** | **B. Sai Vai trò Đăng nhập (Role Mismatch)** |
|  | 1. Trong bước 5, Hệ thống phát hiện vai trò đã chọn **không khớp** với vai trò đã đăng ký (Ví dụ: Email đăng ký là Ứng viên nhưng lại chọn đăng nhập với vai trò Nhà tuyển dụng). |
|  | 2. Hệ thống hiển thị thông báo **"Bạn đã đăng ký với vai trò [Vai trò đã đăng ký]. Vui lòng chọn lại vai trò đăng nhập."** |
|  | 3. Người dùng chọn lại vai trò đăng nhập chính xác hoặc quay lại trang khởi động. |
| **Alternative Flow** | **C. Tài khoản chưa được Xác thực** |
|  | 1. Trong bước 4, tài khoản tồn tại nhưng chưa hoàn tất xác thực Email. |
|  | 2. Hệ thống hiển thị thông báo **"Tài khoản của bạn chưa được kích hoạt. Vui lòng kiểm tra Email để xác thực."** |
|  | 3. Hệ thống có thể cung cấp tùy chọn gửi lại Email xác thực. |

#### **4.2.1.2 Use-case đăng ký tài khoản**

| **Thuộc tính** | **Chi tiết** |
| --- | --- |
| **Usecase Id** | UC\_002 |
| **Usecase Name** | Đăng ký Tài khoản (Sign Up) |
| **Description** | Người dùng tạo tài khoản mới trên hệ thống Job Bridge để bắt đầu sử dụng dịch vụ. |
| **Actor** | Người dùng |
| **Priority** | Cao |
| **Trigger** | Người dùng chọn tùy chọn “Đăng ký” trên màn hình khởi động ứng dụng. |
| **Pre-condition** | Người dùng chưa có tài khoản trên hệ thống Job Bridge. |
| **Post-condition** | Người dùng tạo thành công tài khoản, thông tin được lưu trữ trong Database và sẵn sàng đăng nhập. |
| **Basic Flow** | 1. Người dùng chọn vai trò ["Nhà tuyển dụng","Ứng viên"] |
|  | 2. Người dùng nhập các thông tin bắt buộc: Email, Mật khẩu (Password), Tên người dùng (Username), và Xác nhận Mật khẩu (Recheck Password). |
|  | 3. Hệ thống kiểm tra tính hợp lệ của dữ liệu (định dạng Email, độ mạnh Mật khẩu, Mật khẩu khớp nhau). |
|  | 4. Hệ thống kiểm tra Email nhập vào đã **tồn tại** trong cơ sở dữ liệu hay chưa. |
|  | 5. Nếu tất cả thông tin hợp lệ và Email chưa tồn tại, hệ thống tiến hành tạo tài khoản mới và lưu trữ dữ liệu. |
|  | 6. Hệ thống hiển thị thông báo "Đăng ký tài khoản thành công" và chuyển hướng người dùng đến màn hình xác thực Email (hoặc trang đăng nhập). |
| **Alternative Flow** | **A. Email đã Tồn tại** |
|  | 1. Trong bước 4, Hệ thống phát hiện Email đã được sử dụng. |
|  | 2. Hệ thống hiển thị thông báo **"Tài khoản đã tồn tại. Vui lòng sử dụng Email khác hoặc chọn Đăng nhập."** |
|  | 3. Người dùng quay lại bước 2 để nhập lại thông tin hoặc chuyển sang trang đăng nhập. |
| **Alternative Flow** | **B. Dữ liệu không Hợp lệ** |
|  | 1. Trong bước 3, Hệ thống phát hiện dữ liệu nhập vào không hợp lệ (ví dụ: định dạng Email sai, Mật khẩu quá ngắn). |
|  | 2. Hệ thống hiển thị thông báo lỗi tương ứng bên dưới trường nhập liệu (ví dụ: "Mật khẩu phải có ít nhất 8 ký tự"). |
|  | 3. Người dùng chỉnh sửa lại thông tin và hệ thống tiếp tục kiểm tra. |

#### **4.2.1.3 Use-case nâng cấp tài khoản**

| Thuộc tính | Chi tiết |
| --- | --- |
| Usecase Id | UC\_003 |
| Usecase Name | Nâng cấp Tài khoản |
| Description | Người dùng (Ứng viên hoặc Nhà tuyển dụng) tiến hành mua và kích hoạt gói dịch vụ cao cấp. |
| Actor | Người dùng (Ứng viên/Nhà tuyển dụng) |
| Priority | Cao |
| Trigger | Người dùng chọn tùy chọn "Nâng cấp Tài khoản" hoặc chọn một gói dịch vụ cao cấp. |
| Pre-condition | Người dùng đã đăng nhập vào hệ thống và đang ở tài khoản Cơ bản. |
| Post-condition | Tài khoản của Người dùng được nâng cấp thành công lên gói dịch vụ cao cấp đã chọn và kích hoạt các tính năng liên quan. |
| Basic Flow | 1. Người dùng truy cập trang "Nâng cấp Tài khoản" hoặc "Quản lý Gói dịch vụ". |
|  | 2. Hệ thống hiển thị danh sách các gói dịch vụ cao cấp (tính năng, thời hạn, chi phí). |
|  | 3. Người dùng chọn một gói dịch vụ mong muốn và chọn thanh toán. |
|  | 4. Hệ thống chuyển hướng Người dùng đến trang/giao diện thanh toán an toàn (Payment Gateway). |
|  | 5. Người dùng cung cấp thông tin thanh toán (ví dụ: thông tin thẻ) và xác nhận giao dịch. |
|  | 6. Hệ thống Backend giao tiếp với Payment Gateway để xử lý và xác nhận giao dịch thành công. |
|  | 7. Hệ thống cập nhật trạng thái tài khoản của Người dùng lên gói dịch vụ cao cấp đã chọn. |
|  | 8. Hệ thống hiển thị thông báo "Nâng cấp Tài khoản thành công" và cho phép truy cập các tính năng nâng cao. |
| Alternative Flow | Thanh toán Thất bại |
|  | 1. Trong bước 6, Payment Gateway trả về trạng thái giao dịch thất bại (ví dụ: thẻ hết hạn, không đủ tiền). |
|  | 2. Hệ thống hiển thị thông báo lỗi "Thanh toán không thành công. Vui lòng thử lại" kèm lý do (nếu có). |
|  | 3. Người dùng quay lại bước 5 để nhập lại thông tin thanh toán hoặc chọn phương thức khác. |
|  | Gói đã được Kích hoạt |
|  | 1. Người dùng cố gắng nâng cấp lên gói dịch vụ mà họ đã đang sử dụng. |
|  | 2. Hệ thống hiển thị thông báo "Bạn đang sử dụng gói dịch vụ này. Vui lòng chọn gia hạn hoặc gói cao hơn." |
|  | 3. Người dùng chọn gia hạn hoặc thoát khỏi quy trình. |

#### **4.2.1.4 Use-case nhận thông báo**

| **Thuộc tính** | **Chi tiết** |
| --- | --- |
| **Usecase Id** | UC\_004 |
| **Usecase Name** | Nhận và Hiển thị Thông báo |
| **Description** | Người dùng nhận và xem các thông báo tức thời (Push Notifications) và danh sách thông báo trong ứng dụng. |
| **Actor** | Người dùng (Ứng viên/Nhà tuyển dụng) |
| **Priority** | Cao |
| **Trigger** | Hệ thống có sự kiện mới liên quan đến người dùng (ví dụ: Nhà tuyển dụng nhận hồ sơ mới, Ứng viên nhận tin nhắn chat). |
| **Pre-condition** | Người dùng đã đăng nhập và cho phép ứng dụng gửi thông báo (Push Notification). |
| **Post-condition** | Thông báo đã được tạo, gửi đến thiết bị, hiển thị cho người dùng và được lưu trữ trong lịch sử thông báo. |
| **Basic Flow** | 1. **Sự kiện Kích hoạt:** Một sự kiện xảy ra trong hệ thống (ví dụ: Ứng viên gửi hồ sơ). |
|  | 2. **Backend Tạo Thông báo:** Hệ thống Backend xác định người dùng nhận (Nhà tuyển dụng) và tạo nội dung thông báo tương ứng. |
|  | 3. **Gửi Push Notification:** Hệ thống gửi thông báo đẩy (Push Notification) đến thiết bị di động của người dùng qua dịch vụ phù hợp (Firebase Cloud Messaging/APNs). |
|  | 4. **Lưu trữ:** Thông báo được lưu trữ vào cơ sở dữ liệu để xem lại trong Trung tâm Thông báo. |
|  | 5. **Hiển thị:** Ứng dụng hiển thị thông báo ngay lập tức trên màn hình thiết bị (nếu ứng dụng đang chạy nền) và cập nhật số lượng thông báo chưa đọc. |
|  | 6. **Người dùng Truy cập:** Người dùng chọn biểu tượng Trung tâm Thông báo trên ứng dụng. |
|  | 7. **Hiển thị Danh sách:** Ứng dụng tải và hiển thị danh sách tất cả các thông báo. |
|  | 8. **Đánh dấu Đã đọc:** Người dùng chạm vào một thông báo, hệ thống đánh dấu thông báo đó là "Đã đọc" và chuyển hướng người dùng đến màn hình chi tiết (ví dụ: Hộp thư Chat hoặc Hồ sơ ứng viên). |
| **Alternative Flow** | **A. Thất bại Gửi Push Notification (Thiết bị Offline)** |
|  | 1. Trong bước 3, thiết bị người dùng đang ngoại tuyến (offline). |
|  | 2. Dịch vụ Push Notification giữ thông báo trong hàng đợi. |
|  | 3. Khi thiết bị kết nối lại, thông báo sẽ được gửi. Thông báo vẫn được lưu trữ và hiển thị đầy đủ trong Trung tâm Thông báo (bước 7). |
|  | **B. Không có Quyền Gửi Thông báo** |
|  | 1. Người dùng đã tắt quyền nhận thông báo của ứng dụng trên thiết bị. |
|  | 2. Trong bước 3, Push Notification không được hiển thị trên màn hình. |
|  | 3. Thông báo vẫn được lưu trữ trong Trung tâm Thông báo (bước 4) và người dùng có thể xem khi truy cập trực tiếp (bước 6). |

#### **4.2.1.5 Use-case đăng tin tuyển dụng**

| **Usecase Id** | UC\_005 |
| --- | --- |
| **Usecase Name** | Đăng tin tuyển dụng |
| **Description** | Employer tạo tin tuyển dụng mới bằng cách nhập thông tin và gửi lên hệ thống. |
| **Actor** | Employer |
| **Priority** | Cao |
| **Trigger** | Người dùng chọn chức năng 'Tạo job mới'. |
| **Pre-condition** | Người dùng đã đăng nhập vào hệ thống. |
| **Post-condition** | Job mới được tạo thành công và hiển thị trong danh sách job. |
| **Basic Flow** | 1. Người dùng chọn 'Đăng tin tuyển dụng'.  2. Nhập thông tin chi tiết về job (tên, mô tả, yêu cầu, hạn nộp...).  3. Nhấn nút 'Tạo job'.  4. Hệ thống gửi API để lưu job và phản hồi kết quả.  5. Giao diện cập nhật danh sách job mới. |
| **Alternative Flow** | 1. Thiếu thông tin bắt buộc: Hiển thị cảnh báo yêu cầu điền đủ.  2. Lỗi mạng/API: Hiển thị thông báo lỗi và cho phép thử lại. |

#### **4.2.1.6 Use-case duyệt ứng viên**

| **Usecase Id** | UC\_006 |
| --- | --- |
| **Usecase Name** | Duyệt ứng viên |
| **Description** | Employer duyệt hoặc từ chối ứng viên cho một job. |
| **Actor** | Employer |
| **Priority** | Cao |
| **Trigger** | Người dùng chọn ứng viên và chọn hành động duyệt/từ chối. |
| **Pre-condition** | Job có ứng viên đang chờ duyệt. |
| **Post-condition** | Trạng thái ứng viên được cập nhật thành công. |
| **Basic Flow** | 1. Người dùng mở danh sách ứng viên.  2. Chọn ứng viên cần duyệt/từ chối.  3. Nhấn nút tương ứng để gửi yêu cầu duyệt/từ chối.  4. Hệ thống cập nhật trạng thái ứng viên qua API. |
| **Alternative Flow** | 1. Lỗi API: Hiển thị thông báo lỗi và giữ nguyên trạng thái. |

#### **4.2.1.7 Use-case giử mail cho ứng viên**

| **Usecase Id** | UC\_007 |
| --- | --- |
| **Usecase Name** | Gửi email cho ứng viên |
| **Description** | Employer gửi email đến ứng viên dựa trên các template có sẵn. |
| **Actor** | Employer |
| **Priority** | Trung bình |
| **Trigger** | Người dùng chọn 'Gửi email' trong trang ứng viên. |
| **Pre-condition** | Ứng viên có địa chỉ email hợp lệ. |
| **Post-condition** | Email được gửi thành công đến ứng viên. |
| **Basic Flow** | 1. Người dùng chọn ứng viên cần gửi email.  2. Chọn template hoặc nhập nội dung mới.  3. Nhấn nút 'Gửi email'.  4. Hệ thống gửi email qua API và hiển thị thông báo thành công. |
| **Alternative Flow** | 1. Lỗi gửi email: Hiển thị thông báo và cho phép thử lại. |

#### 

#### **4.2.1.8 Use-case Tìm kiếm và khám phá việc làm**

| **Thuộc tính** | **Chi tiết** |
| --- | --- |
| **Usecase Id** | UC\_008 |
| **Usecase Name** | Tìm kiếm và khám phá việc làm (Job Search & Discovery) |
| **Description** | Ứng viên có thể tìm kiếm công việc theo tên, công ty, địa điểm hoặc được hệ thống AI gợi ý việc làm phù hợp dựa trên nội dung CV và hồ sơ nghề nghiệp của họ. |
| **Actor** | Ứng viên |
| **Priority** | Cao |
| **Trigger** | Ứng viên truy cập ô tìm kiếm ở trang home. |
| **Pre-condition** | Ứng viên đã đăng nhập và có CV được lưu trong hệ thống. |
| **Post-condition** | Danh sách công việc phù hợp được hiển thị, cho phép ứng viên xem chi tiết hoặc lưu/ứng tuyển công việc mong muốn. |
| **Basic Flow** | 1. Ứng viên truy cập ô tìm kiếm ở trang home. |
|  | 2. Ứng viên nhập **từ khóa** tìm kiếm (tên việc làm, công ty, hoặc địa điểm). |
|  | 3. Hệ thống truy vấn cơ sở dữ liệu việc làm theo từ khóa và bộ lọc đã chọn. |
|  | 4. Hệ thống hiển thị danh sách việc làm phù hợp kèm thông tin cơ bản (tên, công ty, địa điểm, mức lương, thời gian đăng). |
|  | 5. Ứng viên chọn một việc làm để xem chi tiết (lưu, ứng tuyển). |
|  | 6. Đồng thời, hệ thống AI phân tích nội dung CV và hồ sơ nghề nghiệp của ứng viên để **gợi ý các việc làm tương tự hoặc phù hợp nhất**. |
| **Alternative Flow** | **A. Không tìm thấy kết quả** |
|  | 1. (Từ bước 4) Nếu không có việc làm nào phù hợp với từ khóa hoặc bộ lọc, hệ thống hiển thị thông báo “Không tìm thấy công việc phù hợp.” và gợi ý ứng viên mở rộng phạm vi tìm kiếm. |
| **Alternative Flow** | **B. Gợi ý AI không khả dụng** |
|  | 1. Nếu module AI tạm thời không hoạt động hoặc chưa có đủ dữ liệu từ CV, hệ thống hiển thị thông báo “Chưa đủ thông tin để gợi ý việc làm. Hãy cập nhật CV của bạn để nhận gợi ý chính xác hơn.” |

#### **4.2.1.9 Use-case Ứng tuyển và theo dõi trạng thái**

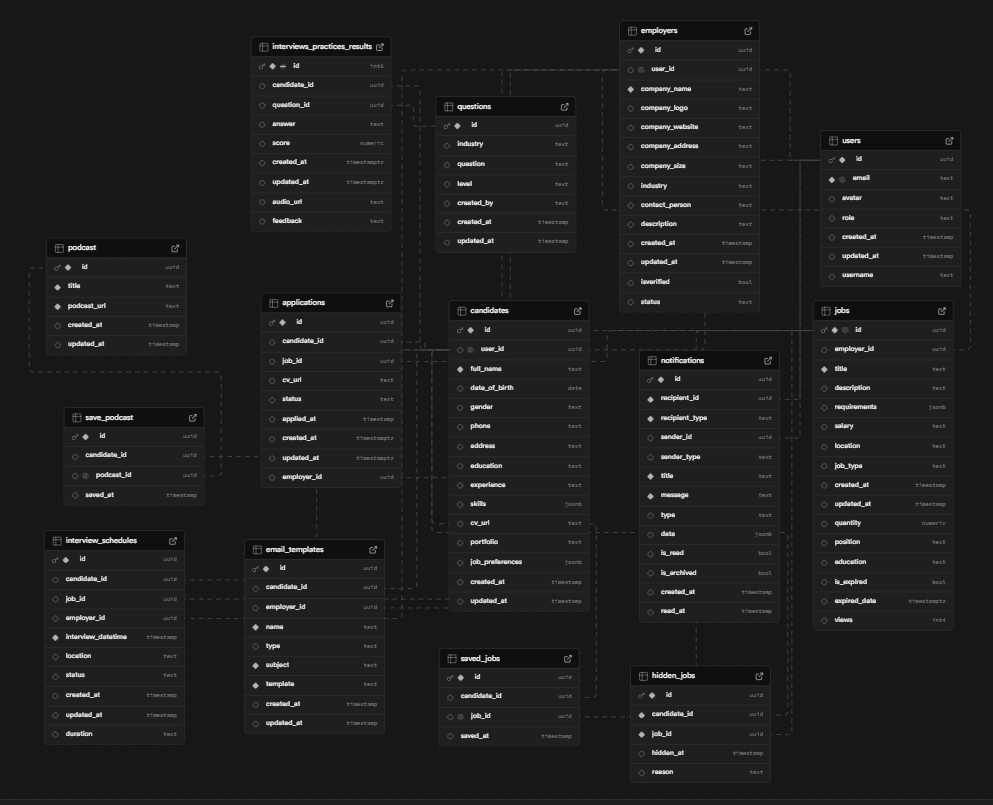
| **Thuộc tính** | **Chi tiết** |
| --- | --- |
| **Usecase Id** | UC\_009 |
| **Usecase Name** | Ứng tuyển và theo dõi trạng thái (Apply & Track Job Status) |
| **Description** | Ứng viên có thể ứng tuyển vào công việc mong muốn trực tiếp từ trang chi tiết công việc, và theo dõi trạng thái xử lý hồ sơ trong mục “Công việc đã ứng tuyển” tại trang cá nhân. |
| **Actor** | Ứng viên |
| **Priority** | Cao |
| **Trigger** | Ứng viên chọn nút “Ứng tuyển ngay” tại trang chi tiết công việc hoặc truy cập “Công việc đã ứng tuyển” trong hồ sơ cá nhân. |
| **Pre-condition** | Ứng viên đã đăng nhập và có CV được lưu trong hệ thống. |
| **Post-condition** | Ứng viên ứng tuyển thành công; hệ thống lưu thông tin ứng tuyển và hiển thị trạng thái trong danh sách “Công việc đã ứng tuyển”. |
| **Basic Flow** | 1. Ứng viên truy cập trang **Chi tiết công việc**.= |
|  | 2. Ứng viên nhấn nút **“Ứng tuyển ngay”**. |
|  | 3. Hệ thống kiểm tra: hồ sơ ứng viên có đầy đủ thông tin cần thiết (CV hợp lệ, chưa ứng tuyển trước đó). |
|  | 4. Nếu hợp lệ, hệ thống lưu thông tin ứng tuyển vào cơ sở dữ liệu, gắn với công việc và tài khoản ứng viên. |
|  | 5. Hệ thống hiển thị thông báo **“Ứng tuyển thành công”** và cập nhật vào danh sách “Công việc đã ứng tuyển”. |
|  | 6. Ứng viên có thể truy cập **“Công việc đã ứng tuyển”** trong **Trang cá nhân** để xem danh sách các công việc mình đã ứng tuyển cùng trạng thái hiện tại. |
| **Alternative Flow** | **A. Thiếu thông tin hồ sơ hoặc CV** |
|  | 1. (Từ bước 3) Nếu ứng viên chưa có CV hoặc hồ sơ chưa hoàn thiện, hệ thống hiển thị thông báo “Vui lòng tải lên CV hoặc hoàn thiện hồ sơ trước khi ứng tuyển.”. |
| **Alternative Flow** | **B. Đã ứng tuyển trước đó** |
|  | 1. Từ bước 3) Nếu ứng viên đã từng ứng tuyển vào công việc này, hệ không cho ứng viên ấn nút ứng tuyển nữa. |
| **Alternative Flow** | **C. Theo dõi trạng thái** |
|  | 1. Ứng viên mở mục **“Công việc đã ứng tuyển”** trong hồ sơ cá nhân. |
|  | 2. Hệ thống hiển thị danh sách công việc đã nộp kèm **trạng thái hiện tại** (Đã nộp, Đang xem, Mời phỏng vấn, Từ chối, Nhận việc). |
|  | 3. Ứng viên chỉ được xem thông tin, **không thể chỉnh sửa hoặc hủy** hồ sơ. |

**4.2.1.10 Use-case Luyện phỏng vấn**

| **Thuộc tính** | Chi tiết |
| --- | --- |
| **Usecase Id** | UC\_0010 |
| **Description** | Ứng viên có thể luyện tập phỏng vấn với hệ thống thông minh. Hệ thống sẽ lấy thông tin nghề nghiệp từ hồ sơ, cho phép chọn cấp độ phỏng vấn (Intern, Junior, Senior,...), chọn lấy câu hỏi từ ngân hàng hoặc yêu cầu AI tạo mới. Ứng viên trả lời bằng giọng nói, hệ thống ghi âm, chấm điểm và phản hồi nhận xét. |
| **Actor** | Ứng viên |
| **Priority** | Cao |
| **Trigger** | Ứng viên chọn màn hình “Practice interview”. |
| **Pre-condition** | Ứng viên đã đăng nhập và có thông tin nghề nghiệp lưu trong hồ sơ. |
| **Post-condition** | Kết quả luyện phỏng vấn (điểm số, nhận xét) được lưu vào lịch sử luyện tập của ứng viên. |
| **Basic Flow** | 1. Ứng viên chọn chức năng “Luyện phỏng vấn” trong ứng dụng. |
|  | 2. Hệ thống truy xuất thông tin nghề nghiệp từ hồ sơ trong database. |
|  | 3. Ứng viên chọn cấp độ phỏng vấn (Intern, Junior, Senior, ...). |
|  | 4. Ứng viên chọn nguồn câu hỏi: “Từ ngân hàng có sẵn” hoặc “Tạo mới bằng AI”. |
|  | 5. Hệ thống hiển thị danh sách câu hỏi. |
|  | 6. Ứng viên bắt đầu trả lời bằng giọng nói, hệ thống ghi âm và phân tích audio. |
|  | 7. AI chấm điểm dựa trên nội dung audio của ứng viên. |
|  | 8. Hệ thống hiển thị điểm, nhận xét, và gợi ý cải thiện. |
|  | 9. Hệ thống lưu kết quả luyện phỏng vấn vào hồ sơ ứng viên. |
| **Alternative Flow** | A. Không có thông tin nghề nghiệp |
|  | 1. Trong bước 2, Nếu không có thông tin nghề nghiệp → hệ thống yêu cầu ứng viên cập nhật CV trước khi luyện phỏng vấn. |

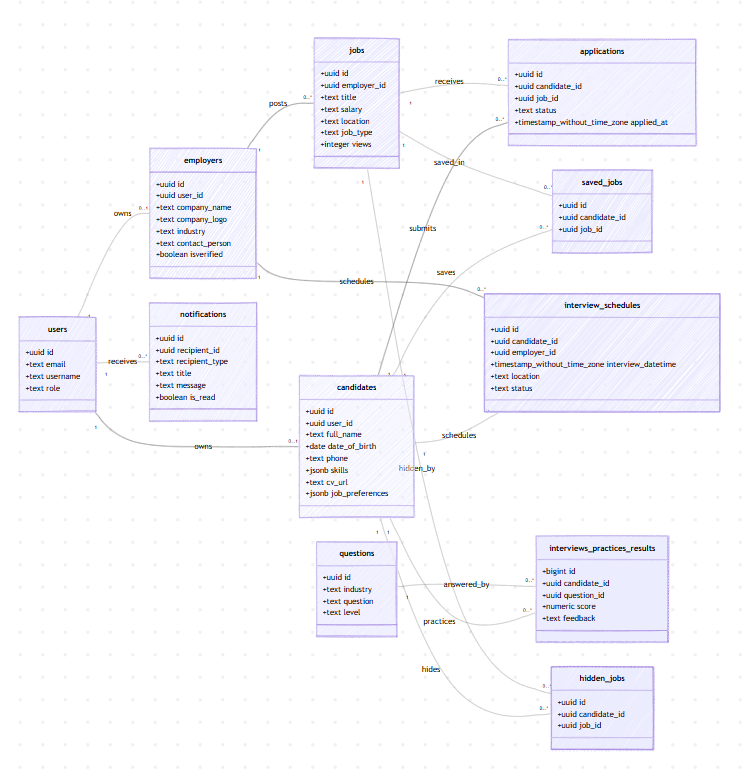
## 4.3 Mô hình lớp và đối tượng

### 4.3.1 Mô hình Erd



*Hình 4.3.1 Mô hình Erd*

### 4.3.2 Mô hình Class

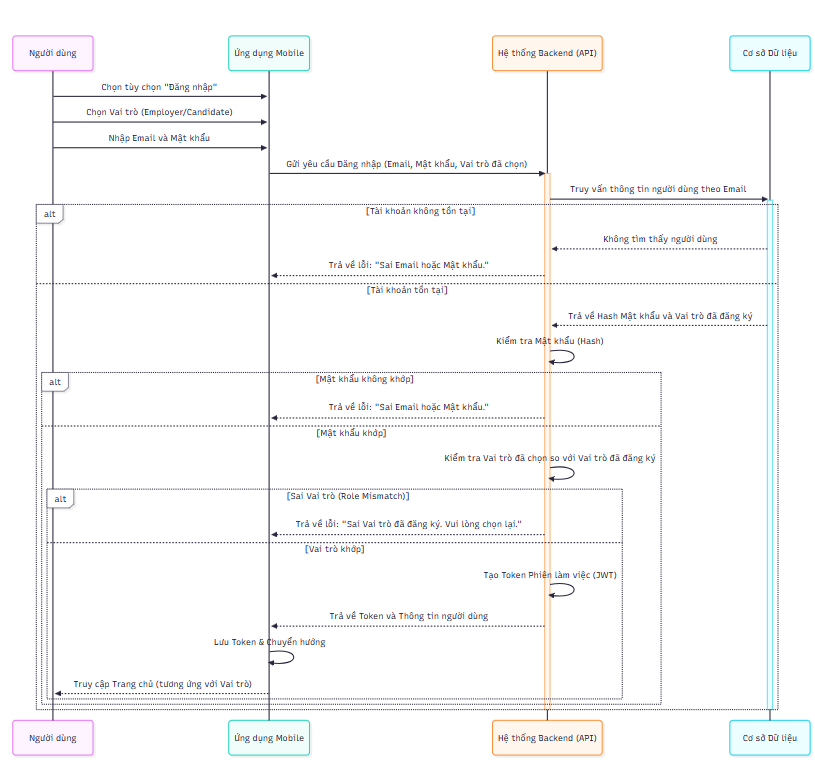


*Hình 4.3.2 Mô hình Class*

## 4.4Các biểu đồ tuần tự

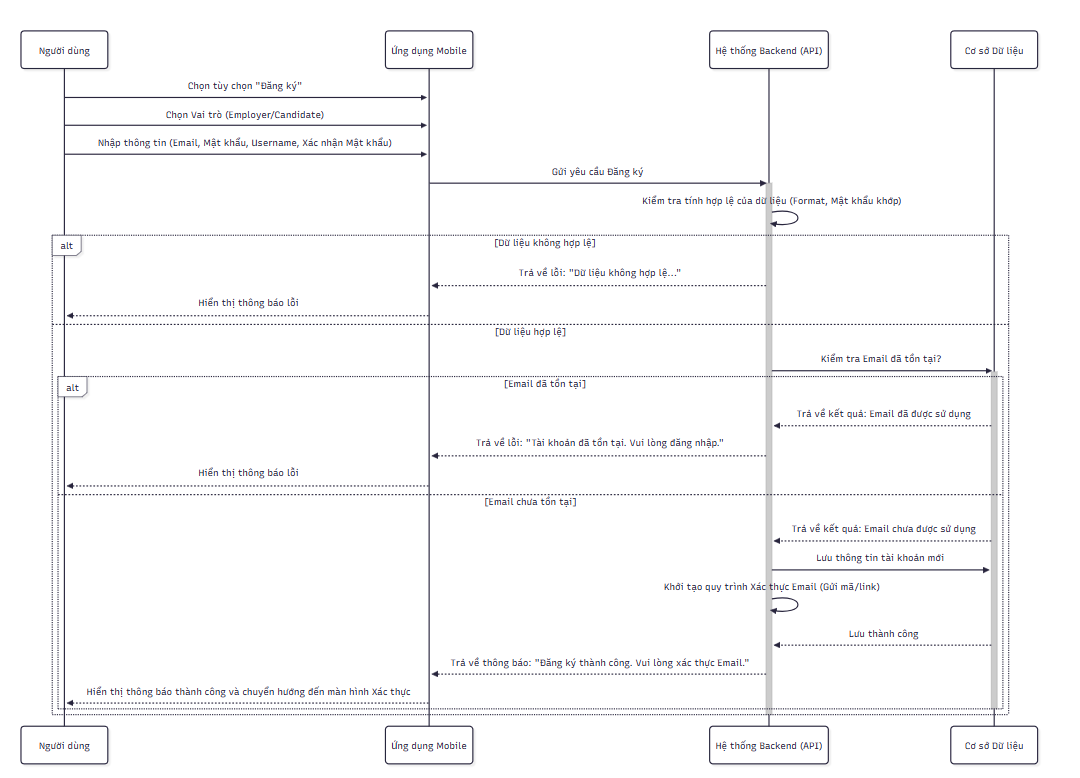
### 4.4.1Biểu đồ tuần tuần tự của người dùng

#### **a, Biểu đồ tuần tự đăng nhập tài khoản**



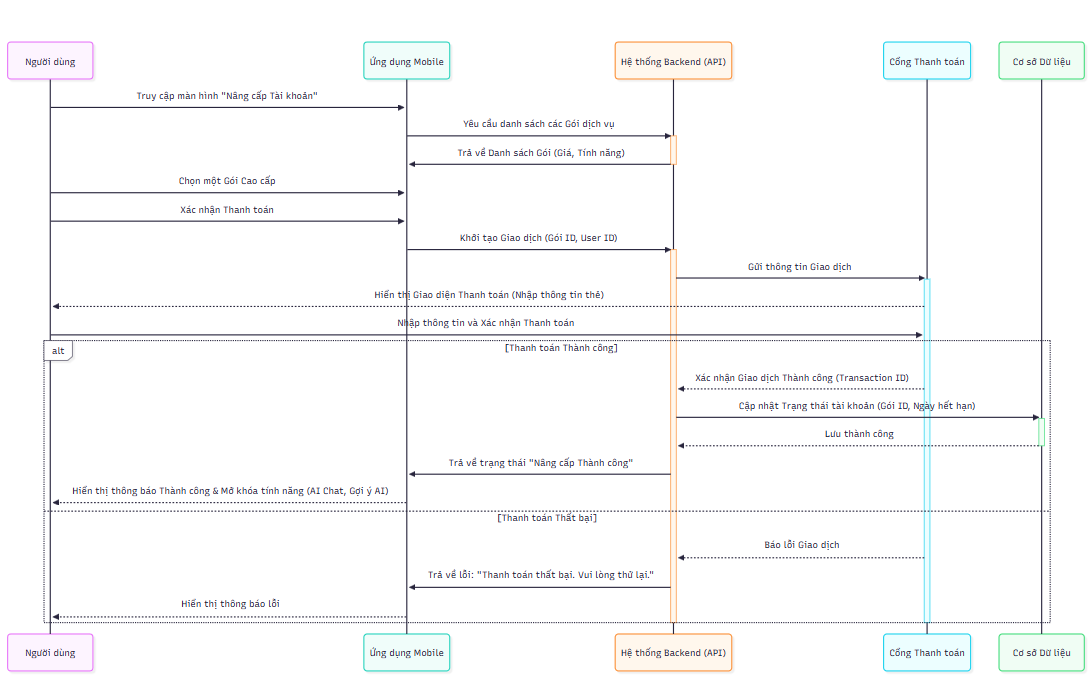
*Hình 4.4.1: Biểu đồ tuần tự đăng nhập tài khoản*

#### **b, Biểu đồ tuần tự đăng ký tài khoản**

****

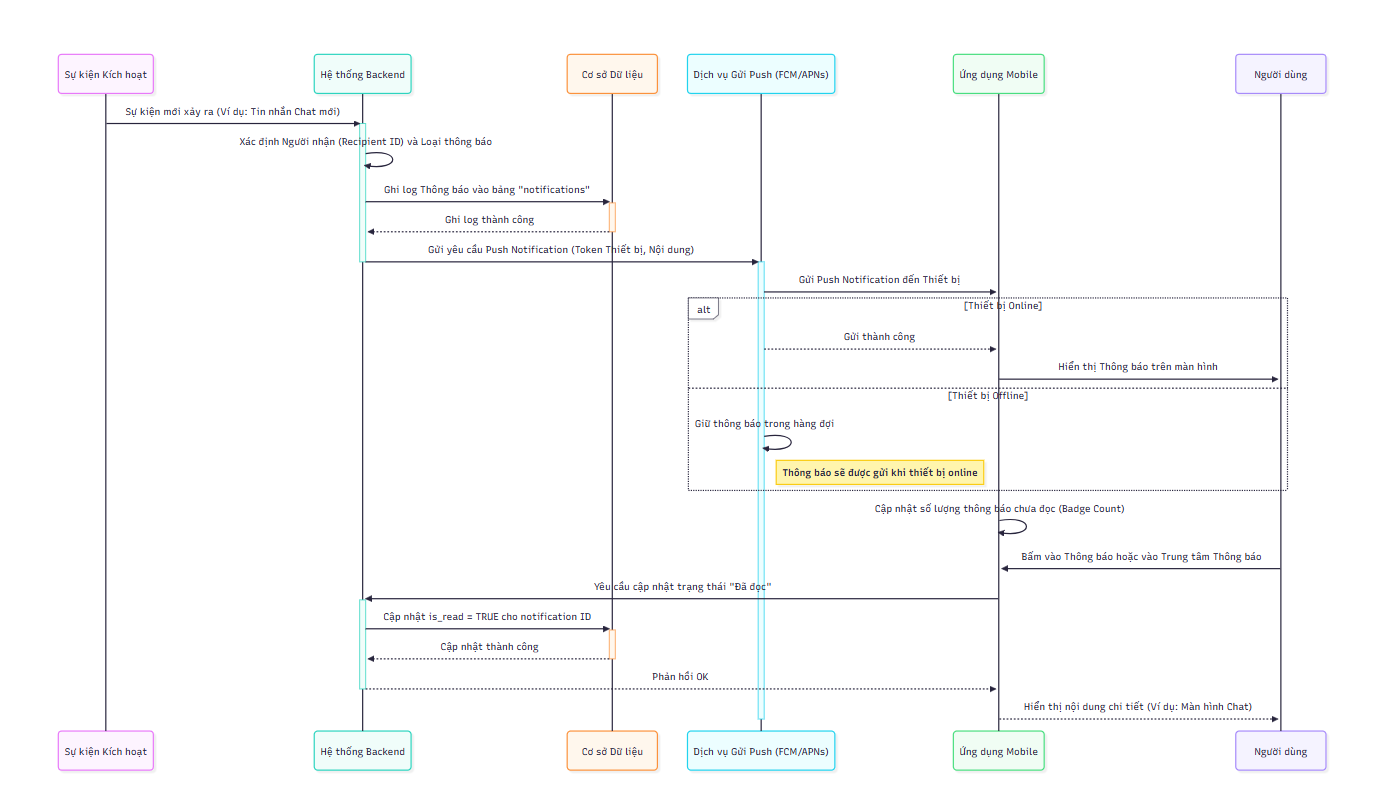
*Hình 4.4.2Biểu đồ tuần tự đăng ký tài khoản*

#### **c, Biểu đồ tuần tự nâng cấp tài khoản**



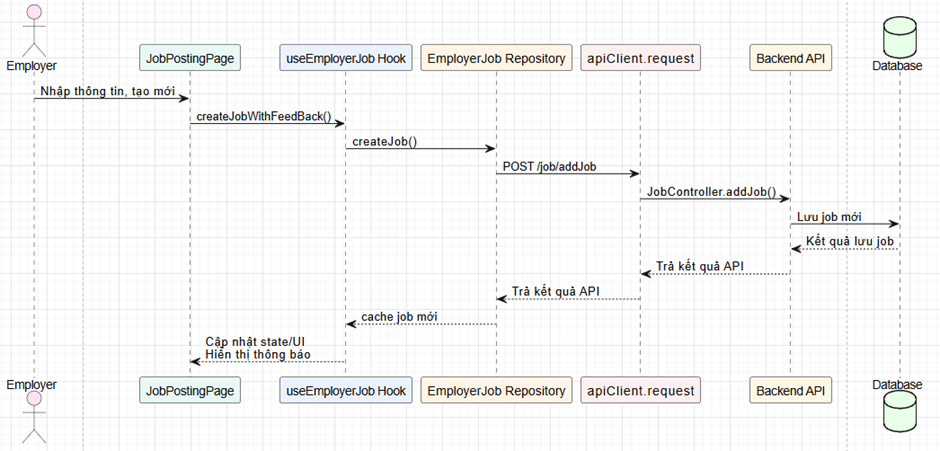
*Hình 4.4.3 Biểu đồ tuần tự Nâng cấp tài khoản*

#### **d, Biểu đồ tuần tự đẩy thông báo**

****

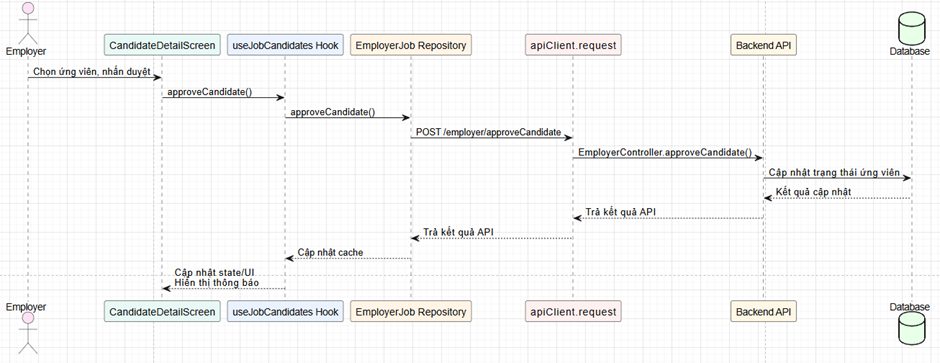
*Hình 4.4.4 Biểu đồ tuần tự đẩy thông báo*

**e, Biểu đồ tuần tự đăng tin tuyển dụng mới**

****

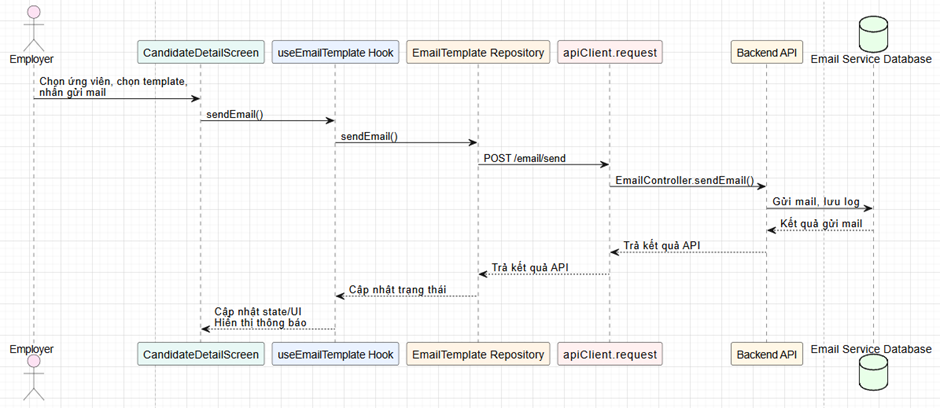
Hình 4.4.5 Biểu đồ tuần tự đăng tin tuyển dụng mới

**f, Biểu đồ tuần tự duyệt ứng viên**

****

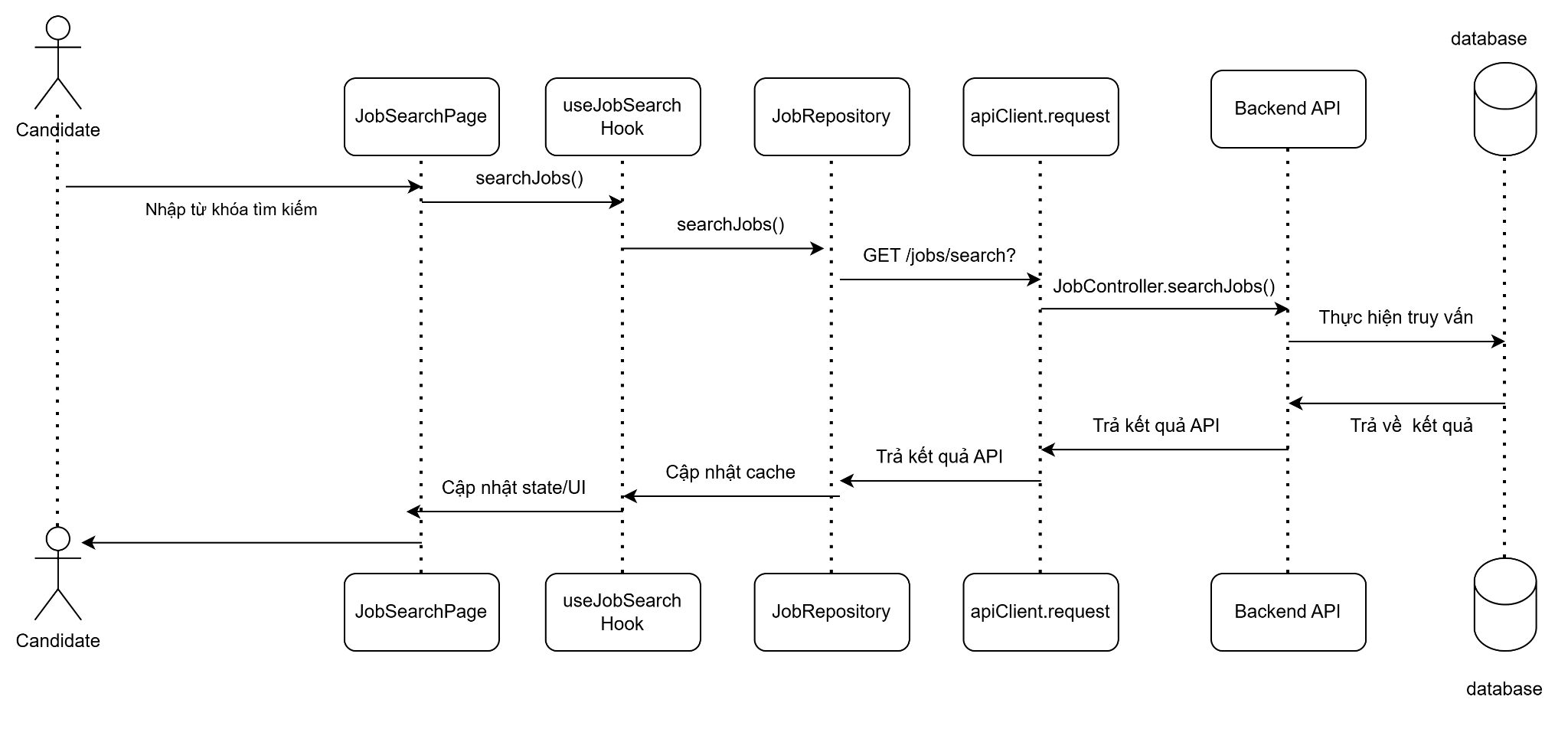
Hình 4.4.6 Biểu đồ tuần tự duyệt ứng viên

**g, Biểu đồ tuần tự gửi email cho ứng viên**

****

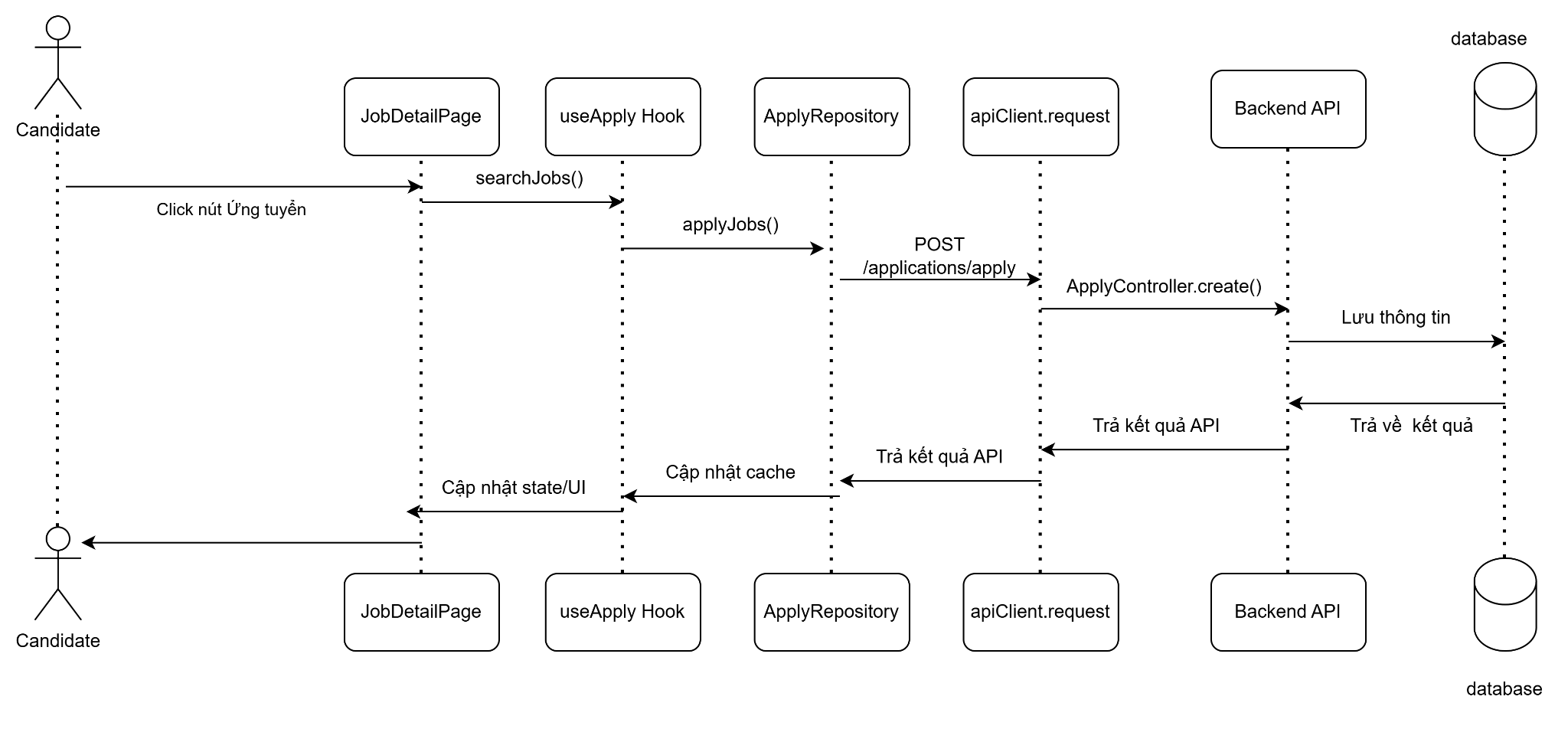
Hình 4.4.7 Biểu đồ tuần tự gửi email cho ứng viên

**h, Biểu đồ tuần tự tìm kiếm việc làm**

****

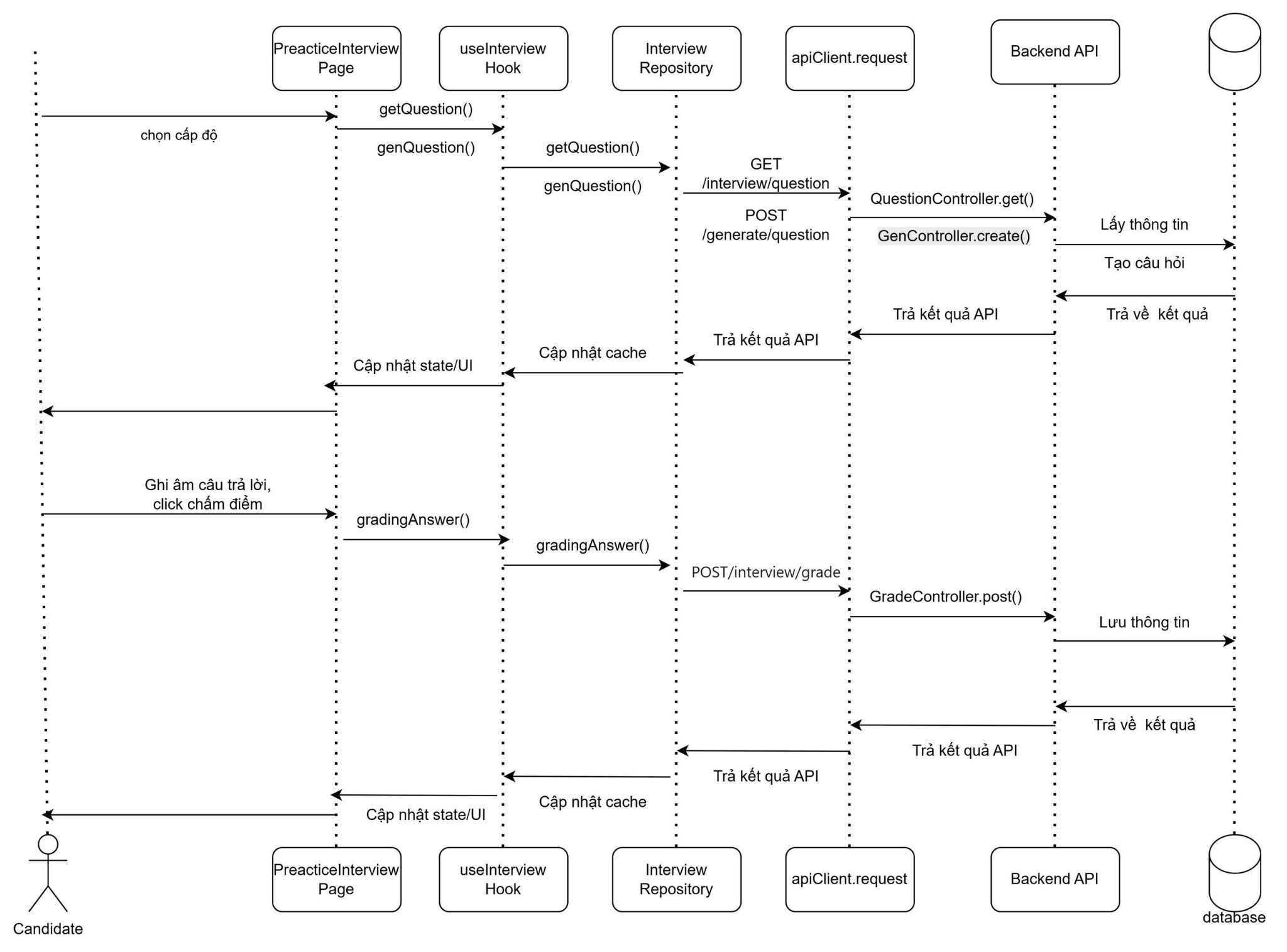
Hình 4.4.8 Biểu đồ tuần tự tìm kiếm việc làm

**i, Biểu đồ tuần tự ứng tuyển**

****

Hình 4.4.9 Biểu đồ tuần tự ứng tuyển

**k, Biểu đồ tuần tự luyện phỏng vấn**

****

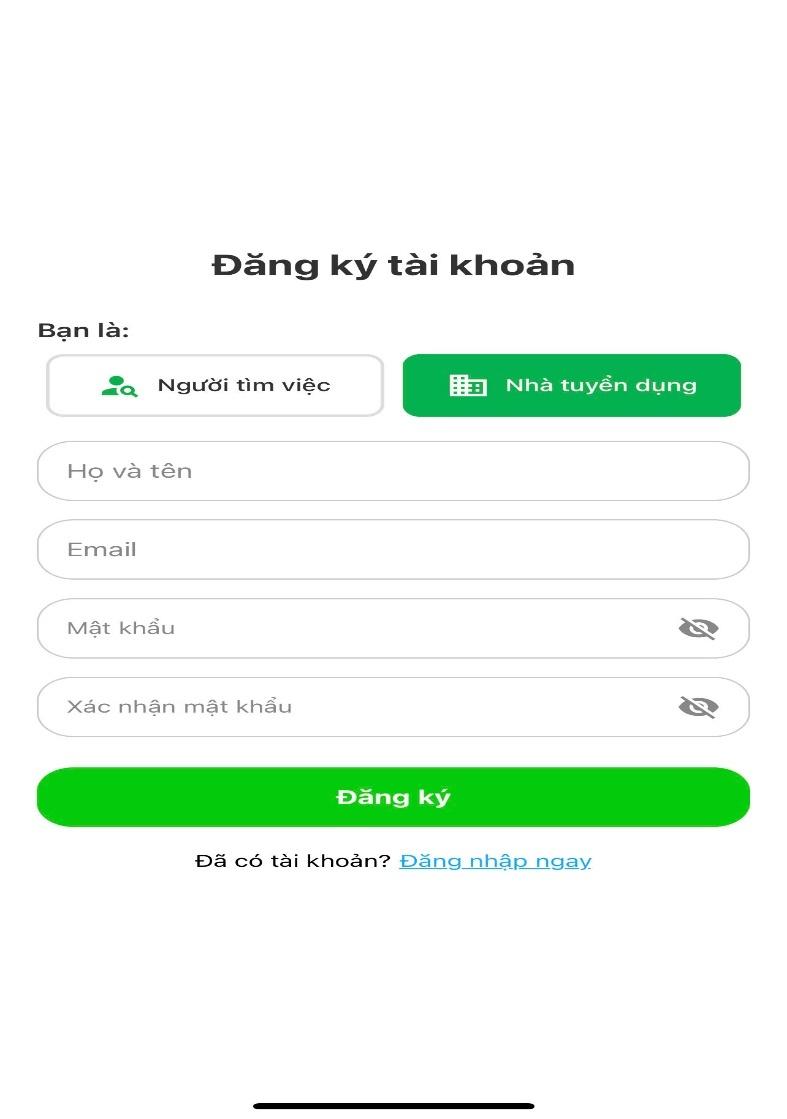
Hình 4.4.10 Biểu đồ tuần tự luyện phỏng vấn

# 5.Các màn hình giao diện người dùng

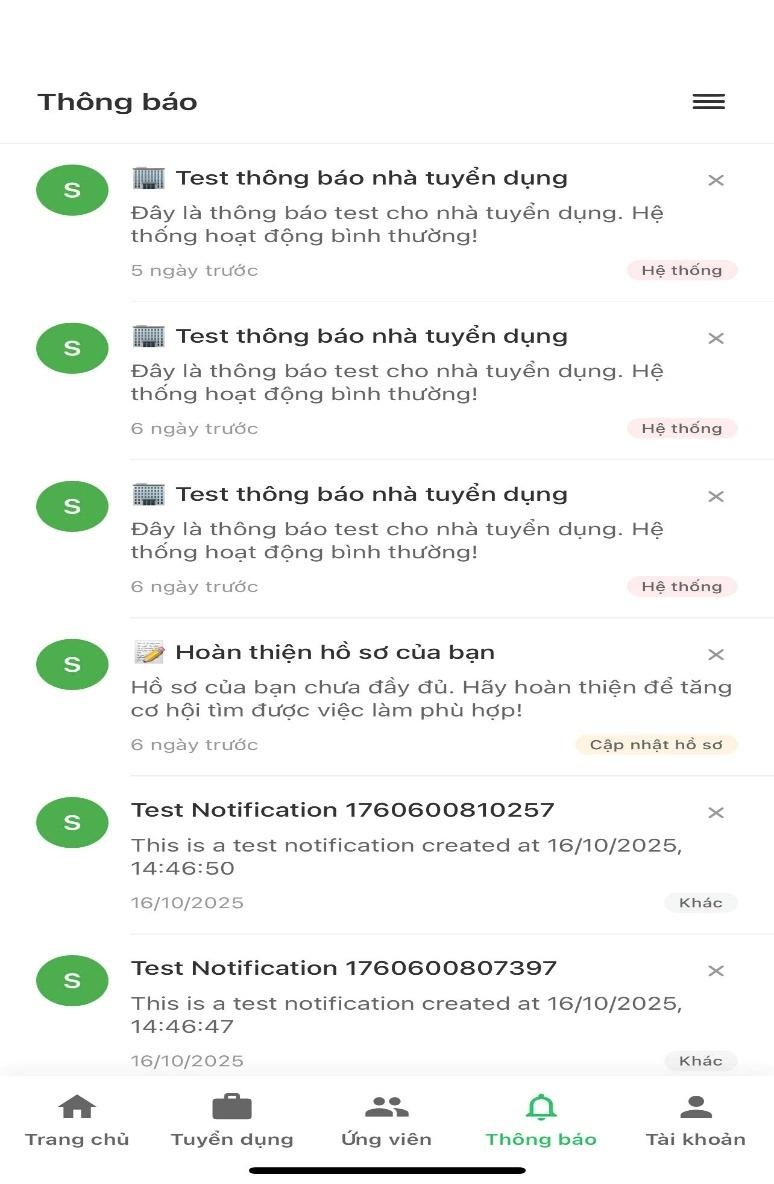
Phần này mô tả các màn hình giao diện người dùng (UI Screens) cốt lõi của ứng dụng di động **Job Bridge**. Giao diện được thiết kế để đảm bảo tính **trực quan, hiện đại** và **thân thiện** với cả hai đối tượng người dùng (Ứng viên và Nhà tuyển dụng), hỗ trợ đầy đủ các chức năng đã được xác định:



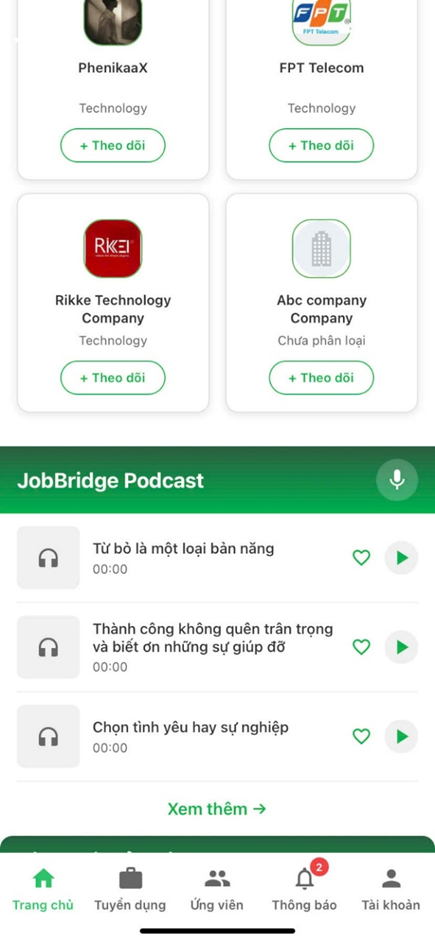
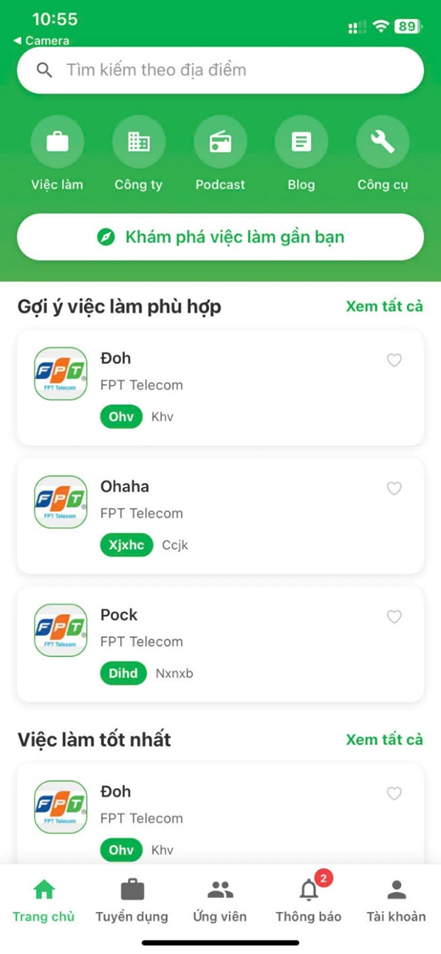
*Hình 5.1 Màn hình đăng nhập của người dùng (User Login Screen)*

**

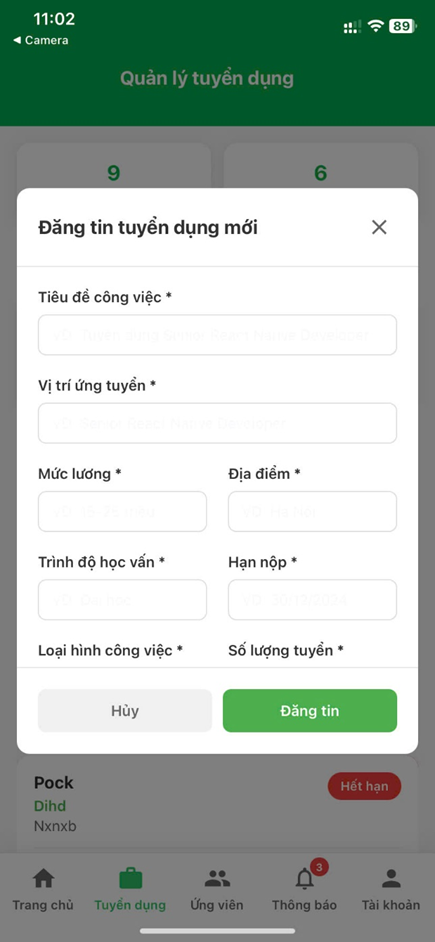
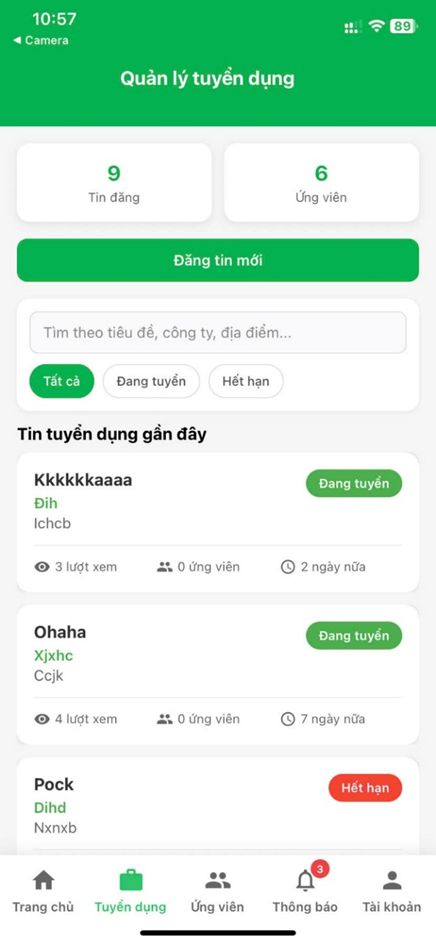
*Hình 5.2 Màn hình đăng ký của người dùng (User Register screen)*

**

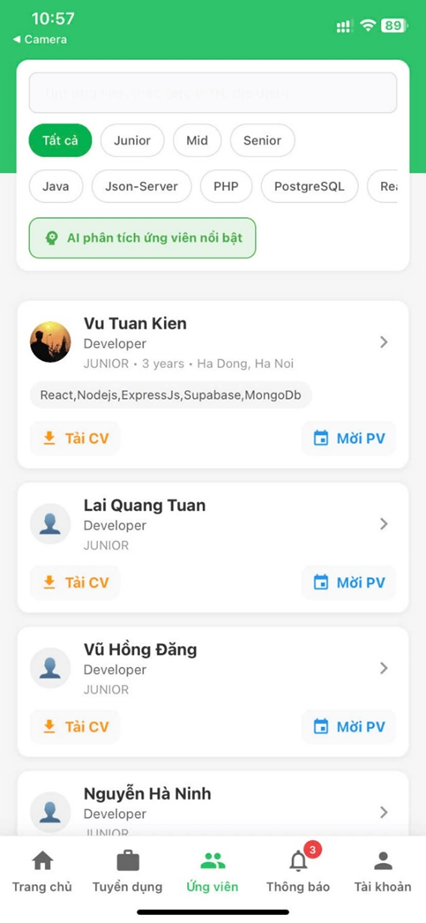
*Hình 5.3 Màn hình nhận thông báo của người dùng (User Notification screen)*



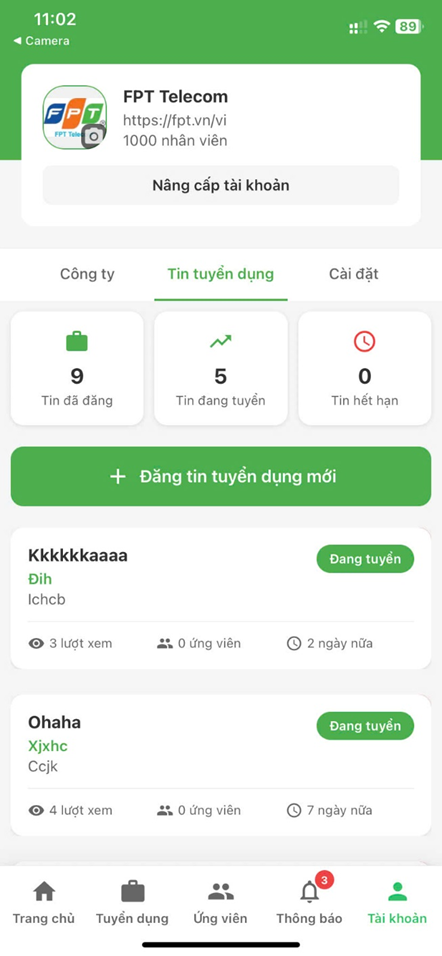
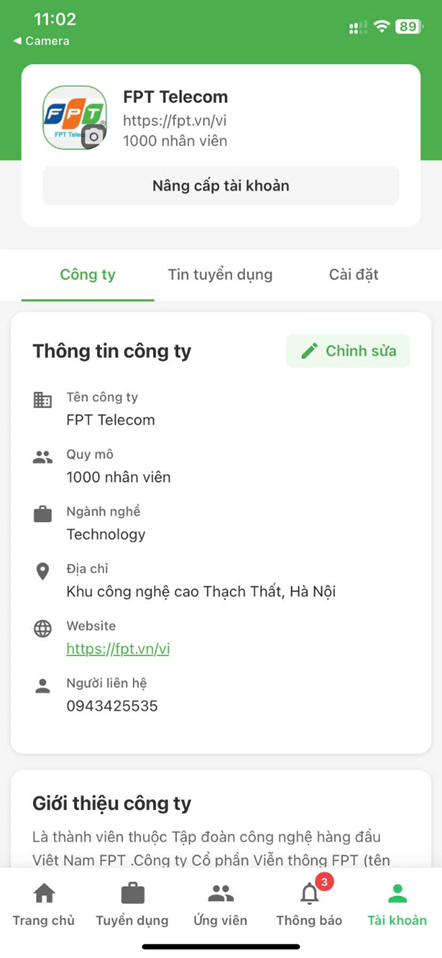
Hình 5.4 Màn hình Trang chủ



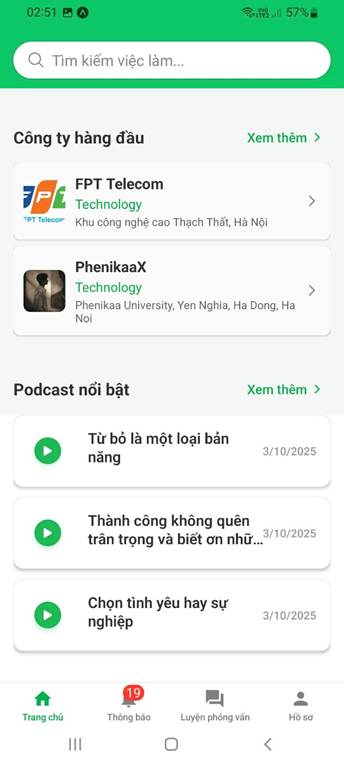
Hình 5.5 Màn hình trang Tuyển dụng



Hình 5.6 Màn hình trang Ứng viên

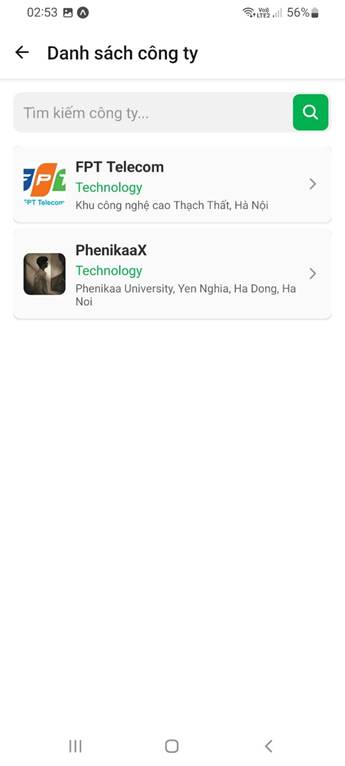


Hình 5.7 Màn hình trang Tài khoản

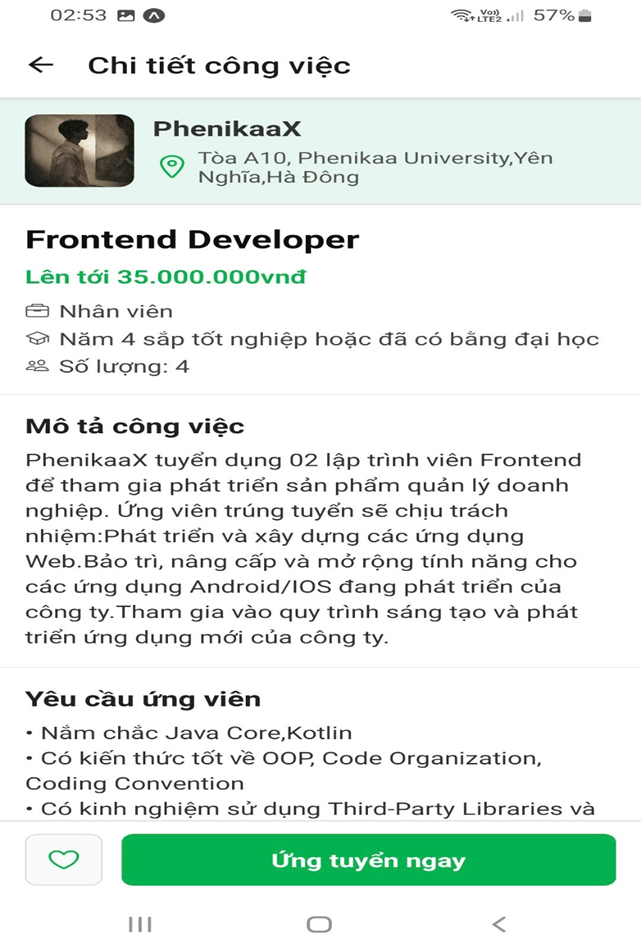
 

Hình 5.8 Màn hình trang chủ Candidate

A screenshot of a phone

AI-generated content may be incorrect.

Hình 5.9 Màn hình tìm kiếm và danh sách công ty



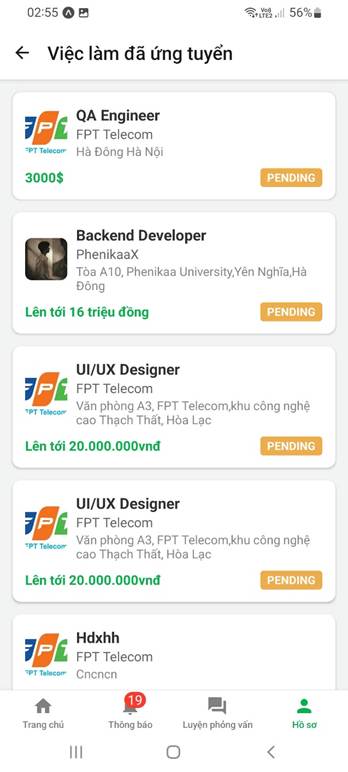
Hình 5.10 Màn hình chi tiết công việc



Hình 5.11 Màn hình trang nhà tuyển dụng(công ty)



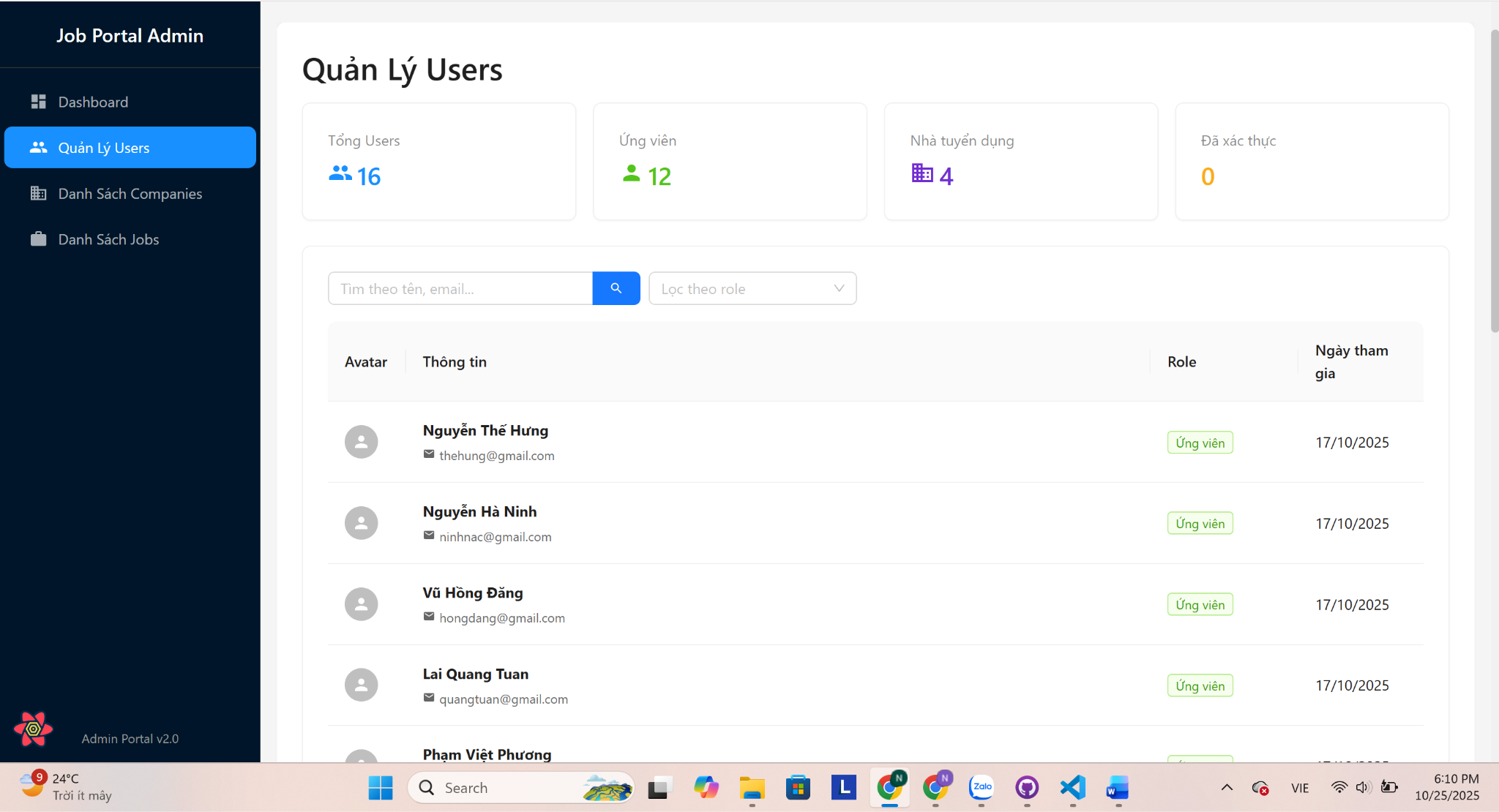
Hình 5.12 Màn hình trang luyện phỏng vấn

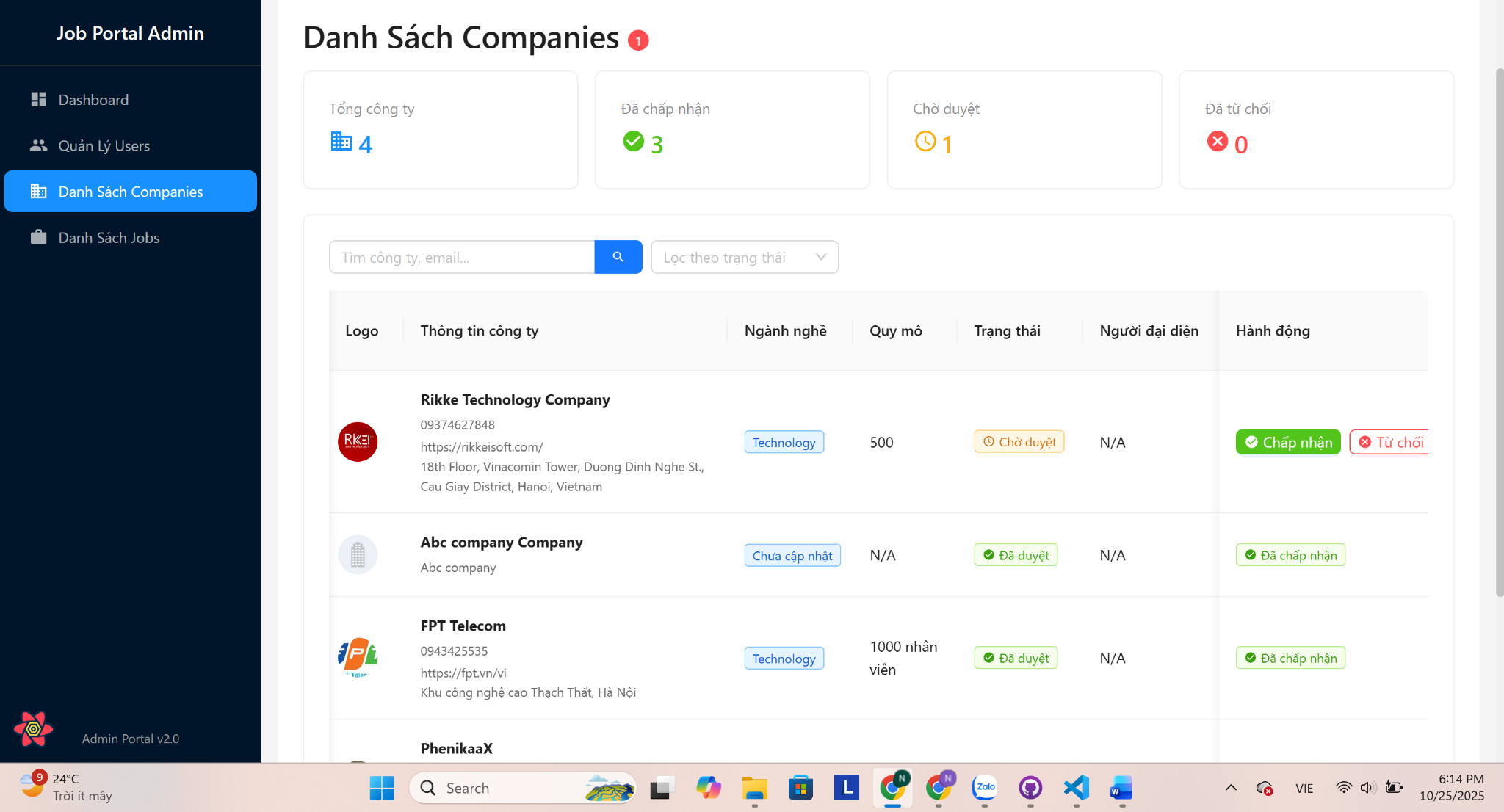
Hình 5.13 Một số màn hình khác của ứng viên



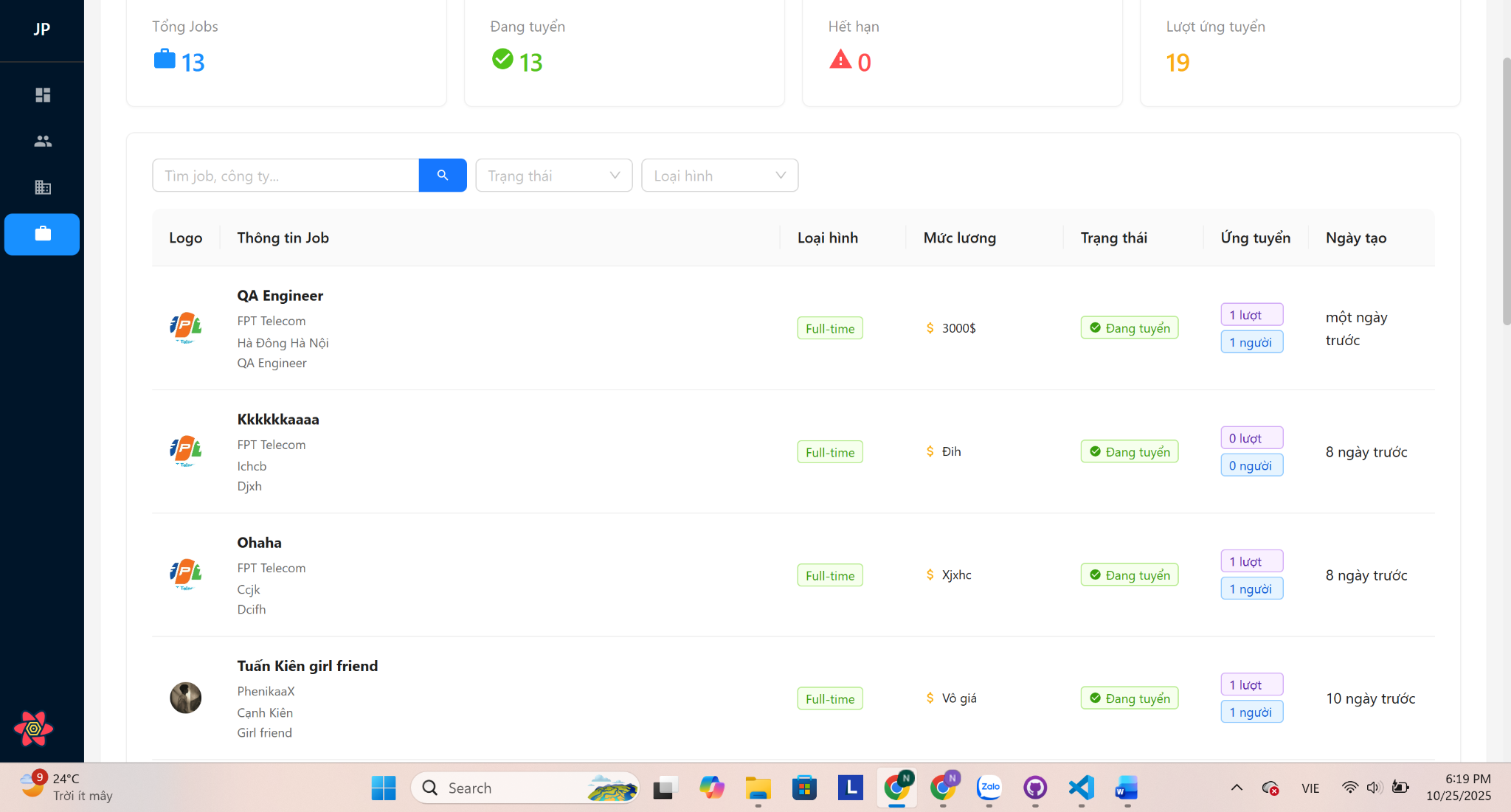
Hình 5.14 Giao diện dashboard của admin web



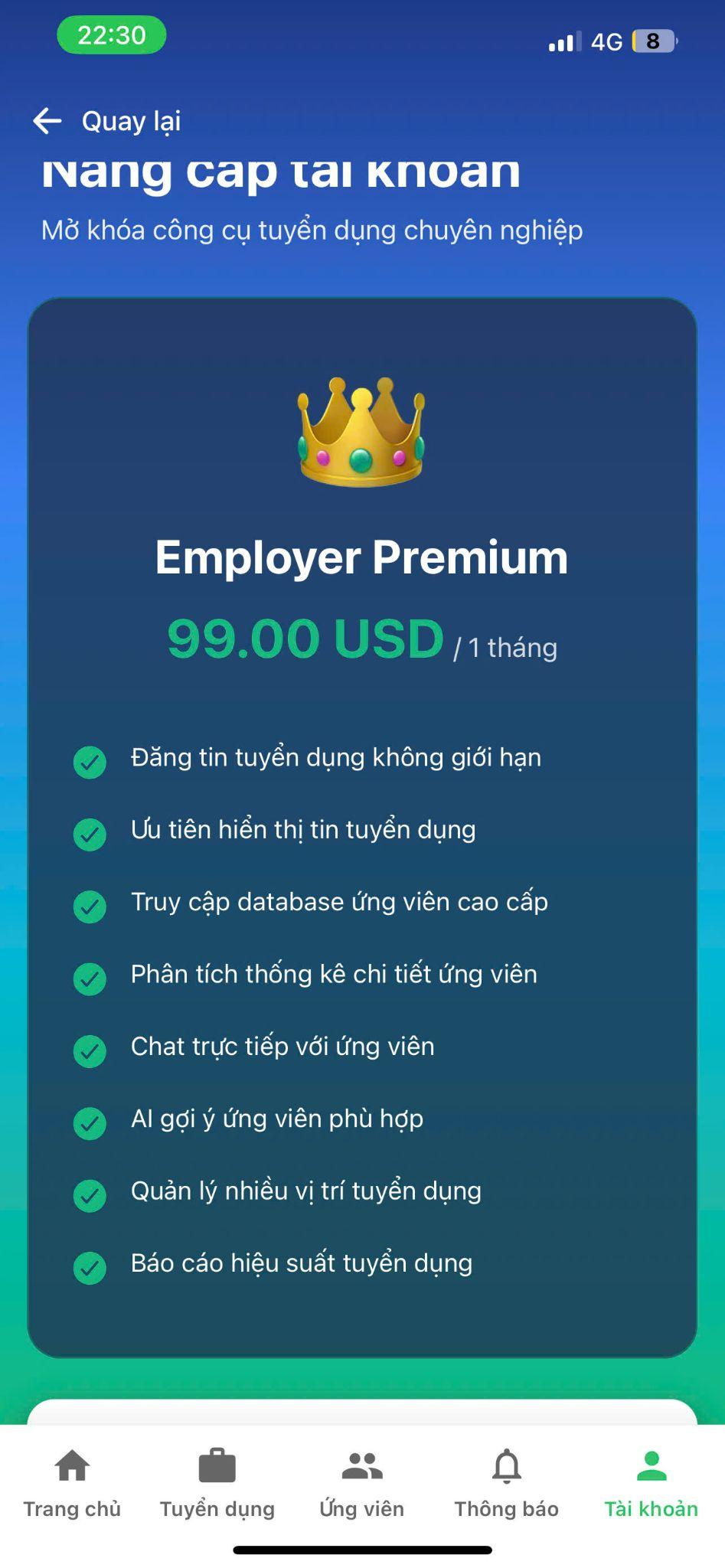
Hình 5.15 Giao diện quản lý người dùng của admin web



Hình 5.16 Giao diện danh sách công ty của admin web



Hình 5.17 Giao diện danh sách job của admin web



# 

Hình 5.18 Giao diện thanh toán và nâng cấp tài khoản

# 

# 

# 

# 

# 

# 

# 

# 

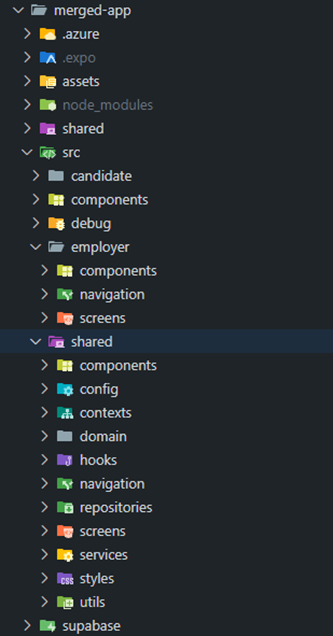
# 

# 

# 6. Công nghệ, thư viện, framework cho frontend

## 6.1 Employer frontend

* **Ngôn ngữ:** JavaScript (ES6+)
* **Framework**: React Native (Expo)
* **Quản lý UI:**
  + React Native core components
  + React Native Paper (UI components)
  + React Navigation (điều hướng giữa các màn hình)
* **Quản lý trạng thái:**
  + React Context API
  + useState, useEffect, custom hooks
* **Kết nối API:**
  + fetch/axios
  + Repository pattern
* **Lưu trữ local:**
  + AsyncStorage (lưu token, cache, thông tin tạm thời)
* **Thư viện phụ trợ:**
  + lodash (xử lý dữ liệu)
  + moment (xử lý thời gian)
  + react-native-dotenv (quản lý biến môi trường)
  + react-native-vector-icons (icon)
* **Thư mục chính:  
   employer**
  + **components/:** Các thành phần UI nhỏ, chia theo chức năng (account, candidates, jobs, notifications, modals, ...).
  + **screens/:** Các màn hình chính (Home, Job List, Job Detail, Candidate List, Candidate Detail, Notification, ...).
  + **navigation/:** Điều hướng (EmployerTabNavigator.js, ...).
  + **services/:** Kết nối API, xử lý dữ liệu (EmployerJobRepository.js, EmployerRepository.js, ...).
  + **hooks/:** Custom hooks cho employer (useEmployerJobs.js, useJobCandidates.js, ...).
  + **styles/:** Style riêng cho employer (nếu có).
* **Thư mục dùng chung:  
   shared** (các component, context, utils, ... dùng chung cho toàn app



# 7.Backend và cơ sở dữ liệu

## 7.1 Bộ công nghệ (TechStack)

* **Ngôn ngữ Lập trình**: Node.js
* **Framework**: Express.js (Sử dụng cho việc định tuyến API và Middleware).
* **Cơ sở Dữ liệu & BaaS**: Supabase (PostgreSQL Database, Authentication, Realtime, và Storage).
* **Kiến trúc**: Model-View-Controller (MVC) (Thường được gọi là Model-Controller trong ngữ cảnh API).
* **Ngôn ngữ Truy vấn Database**: SQL (PostgreSQL) và Supabase Client Libraries.
* **Mô tả Cấu trúc Thư mục**:

Tên Thư mục/Tệp,Chức năng

├─ **src**,Thư mục Nguồn (Source): Chứa toàn bộ mã nguồn của ứng dụng Backend.

├─ **controllers**: Nơi chứa Logic Nghiệp vụ (Business Logic). Xử lý các yêu cầu từ routes, gọi models/services để thao tác dữ liệu, và trả về phản hồi HTTP."

│  ├─ **AdminControllers**:"Controllers riêng cho các chức năng quản trị viên (Admin), như quản lý người dùng, kiểm duyệt tin đăng, v.v."

│  └─ **ClientControllers**:"Controllers chung cho người dùng cuối (Ứng viên và Nhà tuyển dụng) xử lý các yêu cầu như đăng nhập, hồ sơ, tìm kiếm, v.v."

├─ **index.js**:"Tệp Khởi chạy Chính: Điểm bắt đầu của ứng dụng (Entry Point). Thiết lập Express server, kết nối các middleware và routes."

├─ **middlewares**: Chứa các hàm xử lý trung gian, thực hiện các tác vụ trước hoặc sau khi Controllers được gọi, như xác thực JWT, phân quyền, kiểm tra lỗi, Rate Limiting, v.v."

├─ **redis**: Chứa logic và cấu hình để kết nối và sử dụng Redis (ví dụ: Caching, Quản lý phiên, Rate limiting, Message queue)."

├─ **routes**: Định nghĩa các điểm cuối (Endpoints) API và chỉ định Controller nào sẽ xử lý yêu cầu cho mỗi đường dẫn.

│  ├─ **AdminRoutes**:Định nghĩa các đường dẫn API dành riêng cho quản trị viên.

│  ├─ **ClientRoutes:**Định nghĩa các đường dẫn API chung cho người dùng cuối (Ứng viên/Nhà tuyển dụng).

│  └─ **EmployerRoutes**"Định nghĩa các đường dẫn API riêng biệt cho các chức năng quản lý tuyển dụng của Nhà tuyển dụng (ví dụ: Đăng tin, Quản lý hồ sơ)."

│  └─ **index.js**:Tệp tập hợp tất cả các route con và gắn chúng vào ứng dụng chính.

├─ **services**: Chứa các hàm có thể tái sử dụng hoặc các logic phức tạp, tách biệt khỏi Controllers (ví dụ: giao tiếp với API AI, gửi email, tích hợp cổng thanh toán)."

├─ **supabase**: Chứa cấu hình và các hàm tiện ích liên quan đến Supabase.

│  └─ **config.js**:Chứa thông tin cấu hình kết nối tới Supabase (API URL, Secret Key, v.v.)."

├─ **templates**: Thường dùng để lưu trữ các mẫu HTML/text cho Email (ví dụ: mẫu xác thực email, mẫu thông báo phỏng vấn) hoặc các mẫu phản hồi cố định."

└─ **utils**: Chứa các hàm hỗ trợ chung, không liên quan đến logic nghiệp vụ cốt lõi (ví dụ: hàm mã hóa/giải mã, định dạng ngày tháng, các hằng số chung)."

## 7.2. Chi tiết Phát triển & Chức năng (Functionality Implementation)

### 7.2.1. Module Xác thực và Ủy quyền (Authentication & Authorization)

* **Xác thực**: Sử dụng dịch vụ Supabase Auth để xử lý Đăng ký/Đăng nhập (Email/Mật khẩu)
* **Quản lý Token**: Trình bày cách Backend xử lý Token (JWT) do Supabase cung cấp để duy trì phiên làm việc.
* **Phân quyền (Authorization)**: Mô tả cách triển khai phân quyền dựa trên Vai trò (role: candidate/employer) qua Row Level Security (RLS) của PostgreSQL hoặc qua Middleware của Express.js.

### 7.2.2. Xử lý Dữ liệu và Mô hình (Models & Data Handling)

* **Thiết kế Schema Supabase**: Trình bày các bảng chính (Users, Candidates, Employers, Jobs, Applications) và các ràng buộc (Foreign Keys, RLS Policies).
* **Thao tác Dữ liệu**: Sử dụng Supabase Client trong các lớp Models để thực hiện các thao tác CRUD (Create, Read, Update, Delete) một cách hiệu quả.

### 7.2.3. Tích hợp Dịch vụ Bên ngoài (External Services Integration)

* **Dịch vụ AI**: Mô tả cách Controller gọi Services để giao tiếp với các API AI bên ngoài (ví dụ: OpenAI hoặc dịch vụ khác) cho các tính năng như "Luyện tập trò chuyện AI" hoặc "Gợi ý ứng viên AI".
* **Thông báo Đẩy (Push Notifications):** Cách Backend kích hoạt dịch vụ gửi Push (ví dụ: FCM) khi có sự kiện (tạo thông báo, cập nhật trạng thái).
* **Lưu trữ Tệp (File Storage**): Sử dụng Supabase Storage để lưu trữ các tệp lớn (ví dụ: CV của Ứng viên), bao gồm cách quản lý quyền truy cập.

## 7.3. Hiệu năng và Bảo mật Hệ thống

### 7.3.1. Bảo mật API

* **Row Level Security (RLS)**: Vai trò của RLS trong việc bảo vệ dữ liệu, đảm bảo người dùng chỉ được truy cập dữ liệu của chính họ hoặc dữ liệu được phép.
* **Middleware Bảo mật**: Các Middleware được sử dụng trong Express.js (ví dụ: Helmet để bảo mật tiêu đề HTTP, CORS).
* **Rate Limiting**: Mô tả cách triển khai Rate Limiting (thông qua thư viện hoặc dịch vụ) để ngăn chặn tấn công DDoS cơ bản.

### 7.3.2. Khả năng Mở rộng (Scalability)

* **Lợi ích của Supabase**: Giải thích cách Supabase (dịch vụ được quản lý) giúp giảm tải công việc quản lý cơ sở dữ liệu và cho phép mở rộng dễ dàng.
* Thiết kế Bất đồng bộ (Async/Await): Đảm bảo tất cả các thao tác I/O (Database, API ngoài) đều được xử lý bất đồng bộ trong Node.js để tối ưu hóa hiệu suất và khả năng xử lý đồng thời

# 8.Một số thành phần khác của đồ án

## 8.1 Kế hoạch dự án

Kế hoạch dự án được xây dựng để đảm bảo hoàn thành ứng dụng tìm kiếm việc làm trong khoảng thời gian từ tháng 8/2025 đến tháng 10/2025 (khoảng 12 tuần). Dự án được chia thành các giai đoạn cụ thể như sau:

* **Tuần 1-2 (cuối 8/2024):** Phân tích yêu cầu và lập kế hoạch. Thu thập thông tin, xác định các chức năng chính, yêu cầu phi chức năng và công nghệ sử dụng (SERN stack).
* **Tuần 3-4 (đầu 9/2025):** Thiết kế hệ thống. Xây dựng mô hình Use-case, biểu đồ lớp, và giao diện người dùng sơ bộ.
* **Tuần 5-8 (giữa 9/2025 đến đầu 10/2025):** Lập trình và triển khai. Phát triển Frontend (React Native) và Backend (Node.js, Express.js, Supabase), tích hợp API và kiểm thử từng module.
* **Tuần 9-10 (giữa 10/2025):** Kiểm thử và tối ưu hóa. Sửa lỗi, kiểm tra hiệu năng, đảm bảo website hoạt động ổn định trên các thiết bị và trình duyệt khác nhau.
* **Tuần 11-12 (giữa 10/2025 đến cuối 3/2025):** Hoàn thiện tài liệu và trình bày. Viết báo cáo, chuẩn bị demo sản phẩm và nộp đồ án. Kế hoạch này được quản lý chặt chẽ thông qua công cụ Google Sheets để theo dõi tiến độ và đảm bảo hoàn thành đúng hạn.

## 8.2 Đảm bảo thực hiện đúng làm việc nhóm

Dự án được thực hiện bởi một nhóm (gồm 4 thành viên), với các biện háp sau để đảm bảo làm việc nhóm hiệu quả:

* **Phân công nhiệm vụ rõ ràng**: Mỗi thành viên chịu trách nhiệm một phần cụ thể , rõ ràng.
* **Giao tiếp thường xuyên**: Tổ chức họp nhóm hàng tuần qua Google Meet hoặc gặp trực tiếp để cập nhật tiến độ và giải quyết vấn đề.
* **Sử dụng công cụ hỗ trợ**: Dùng Git/GitHub để quản lý mã nguồn, đảm bảo tích hợp code đồng bộ và tránh xung đột.
* **Đánh giá nội bộ**: Các thành viên kiểm tra chéo công việc của nhau (peer review) để đảm bảo chất lượng và phát hiện lỗi sớm.
* Những biện pháp này giúp tối ưu hóa sự phối hợp, giảm thiểu rủi ro và đảm bảo tất cả thành viên đóng góp công bằng vào dự án.

## 8.3 Các vấn đề về đạo đức và làm việc chuyên nghiệp

* Trong quá trình phát triển ứng dụng, nhóm chú trọng đến các khía cạnh đạo đức và chuyên nghiệp:
  + **Bản quyền và sở hữu trí tuệ:** Không sao chép giao diện oặc mã nguồn từ các website khác mà không được phép; hình ảnh sản phẩm sử dụng là tự tạo hoặc từ nguồn miễn phí bản quyền (như Unsplash).
  + **Bảo mật thông tin:** Đảm bảo dữ liệu người dùng (email, thông tin cá nhân) được mã hóa và lưu trữ an toàn trong supabase, không chia sẻ cho bên thứ ba.
  + **Trách nhiệm xã hội:** Tránh quảng bá sản phẩm mỹ phẩm giả mạo hoặc không rõ nguồn gốc trên ứng dụng, đảm bảo tính minh bạch với khách hàng.
  + **Thái độ chuyên nghiệp:** Tuân thủ thời hạn, tôn trọng ý kiến nhóm, và duy trì thái độ tích cực trong quá trình làm việc.
* Dự án là cơ hội để nhóm học hỏi và nâng cao kỹ năng:
  + Kiến thức mới: Làm quen với SERN stack (React Native, Node.js, Supabase, Express.js), hiểu cách tích hợp Frontend và Backend qua API, và sử dụng các công cụ như Git, Postman.
  + Chiến lược học tập:
  + Tự học qua tài liệu trực tuyến (MDN Web Docs, React documentation, freeCodeCamp).
  + Tham gia các khóa học ngắn hạn trên Udemy hoặc Coursera về MERN stack.
  + Thực hành qua các dự án nhỏ trước khi áp dụng vào đồ án chính.
  + Mục tiêu dài hạn: Sau đồ án, nhóm sẽ mở rộng kiến thức về tối ưu hóa hiệu năng website, tích hợp thanh toán trực tuyến, hoặc phát triển ứng dụng di động liên quan.

# 9.Kết luận

Dự án **"Xây dựng ứng dụng di động Job Bridge sử dụng Node.js, Express.js và Supabase"** đã đạt được mục tiêu cốt lõi là tạo ra một nền tảng kết nối nhân sự hiệu quả, hỗ trợ **Ứng viên** tìm việc và **Nhà tuyển dụng** quản lý quy trình tuyển dụng một cách trực quan.

Trong khoảng thời gian triển khai từ cuối tháng 8/2024 đến cuối tháng 10/2025, nhóm đã hoàn thành các chức năng chính bao gồm:

* **Xác thực đa vai trò** (Ứng viên/Nhà tuyển dụng).
* **Quản lý hồ sơ** và **tin tuyển dụng**.
* **Chức năng Nâng cấp Tài khoản** và **Phỏng vấn AI** cốt lõi.

Đồng thời, dự án đã đáp ứng tốt các yêu cầu về **giao diện di động** (tông màu Xanh/Trắng), **hiệu năng** tải dữ liệu, và tuân thủ kiến trúc **MVC** chặt chẽ, sử dụng Supabase để đảm bảo tính sẵn sàng.

Mặc dù còn một số hạn chế cần khắc phục (ví dụ: tối ưu hóa thuật toán AI, tích hợp sâu hơn với cổng thanh toán chính thức), đây là **bước đầu thành công quan trọng**, đặt nền móng vững chắc cho việc mở rộng tính năng và mở rộng quy mô người dùng trong tương lai.

Quan trọng hơn, dự án không chỉ mang lại giá trị thực tiễn cho thị trường lao động mà còn giúp nhóm phát triển **kỹ năng lập trình chuyên sâu** với Node.js/Express.js, khả năng làm việc nhóm theo mô hình Agile, và nâng cao **tư duy thiết kế hệ thống** (System Design) và giải quyết vấn đề. Kết quả này là tiền đề quý giá để tiếp tục khám phá và ứng dụng các công nghệ mới vào các dự án thực tế sau này.

# Tài liệu tham khảo

* Node.js : https://nodejs.org/docs/latest/api/
* Express.js : https://expressjs.com/en/4x/api.html
* Supabase : https://supabase.com/docs
* PostgreSQL (Trong Supabase) : https://www.postgresql.org/docs/current/
* JSON Web Tokens (JWT) : https://jwt.io/introduction
* Kiến trúc MVC : https://expressjs.com/en/guide/routing.html
* React Native (Frontend/Context) : https://react.dev/learn/your-first-component
* Expo <https://docs.expo.dev/>
* Axios https://axios-http.com/docs/intro