

# **TRƯỜNG ĐẠI HỌC PHENIKAA**

## **KHOA CÔNG NGHỆ THÔNG TIN**



## BÁO CÁO ĐỒ ÁN CƠ SỞ

ĐỀ TÀI: PHÁT TRIỂN HỆ THỐNG TUYỂN SINH TRỰC TUYẾN

- |   |                     |          |                                 |
|---|---------------------|----------|---------------------------------|
| 1 | Ninh Thị Phương Mai | 22010074 | 22010074@st.phenikaa-uni.edu.vn |
| 2 | Đặng Hồng Nhung     | 22010022 | 22010022@st.phenikaa-uni.edu.vn |
| 3 | Đoàn Hồng Dương     | 22010005 | 22010005@st.phenikaa-uni.edu.vn |

## ***GVHD: Thầy Vũ Quang Dũng***

Hà Nội, Tháng 10, 2025

## MỤC LỤC

<b>MỤC LỤC.....</b>	<b>2</b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b>3</b>
<b>LỜI CẢM ƠN.....</b>	<b>4</b>
<b>LỜI CAM ĐOAN.....</b>	<b>5</b>
<b>CHƯƠNG I. GIỚI THIỆU.....</b>	<b>6</b>
A. Đặt vấn đề.....	6
B. Các giải pháp đã có.....	7
C. Giải pháp đề xuất.....	8
<b>CHƯƠNG II. THIẾT KẾ VÀ TRIỂN KHAI.....</b>	<b>10</b>
A. Các yêu cầu chức năng.....	10
1. Đối với các thí sinh (Users).....	10
2. Đối với bên nhà trường (Admins).....	13
B. Các yêu cầu phi chức năng.....	15
C. Các ràng buộc.....	19
3. Các ràng buộc về triển khai.....	20
4. Các ràng buộc kinh tế.....	21
5. Các ràng buộc về đạo đức.....	22
<b>CHƯƠNG III. MÔ HÌNH HỆ THỐNG VÀ THIẾT KẾ GIẢI PHÁP.....</b>	<b>25</b>
A. Các kịch bản của hệ thống (Use-cases).....	25
1. Đăng ký.....	25
2. Đăng nhập(Dành cho thí sinh).....	26
3. Điện thông tin cá nhân, học bạ và nguyện vọng.....	27
4. Đăng nhập (Dành cho admin).....	28
5. Theo dõi trạng thái hồ sơ.....	29
6. Đăng xuất.....	30
7. Xem danh sách thí sinh.....	31
8. Tìm kiếm thí sinh.....	31
9. Xem chi tiết hồ sơ thí sinh.....	32
10. Duyệt hồ sơ.....	33
11. Yêu cầu chỉnh sửa hồ sơ.....	33
B. Mô hình Use-case.....	34
C. Mô hình hoạt động.....	38
D. Các màn hình giao diện.....	44
<b>CHƯƠNG III. KẾ HOẠCH DỰ ÁN.....</b>	<b>51</b>
<b>CHƯƠNG IV. KẾT LUẬN.....</b>	<b>52</b>
A. Các vấn đề về đạo đức và làm việc chuyên nghiệp.....	52
B. Tác động xã hội.....	53
C. Kế hoạch cho kiến thức mới và chiến lược học tập.....	54
D. Kết quả đạt được và hạn chế.....	60
<b>CHƯƠNG V. TÀI LIỆU THAM KHẢO.....</b>	<b>63</b>

## **DANH MỤC HÌNH ẢNH**

*Hình 1. Sơ đồ tổng quát Use-case*

*Hình 2. Mô hình Use-case thí sinh*

*Hình 3. Mô hình Use-case admin*

*Hình 4. Biểu đồ hoạt động đăng nhập của thí sinh*

*Hình 5. Biểu đồ hoạt động đăng nhập của admin*

*Hình 6. Biểu đồ hoạt động đăng ký của thí sinh*

*Hình 7. Biểu đồ hoạt động nộp thông tin thí sinh*

*Hình 8. Biểu đồ hoạt động theo dõi trạng thái hồ sơ*

*Hình 9. Biểu đồ hoạt động duyệt hồ sơ*

*Hình 10. Biểu đồ hoạt động gửi yêu cầu chỉnh sửa*

*Hình 11. Giao diện đăng nhập*

*Hình 12. Giao diện đăng ký*

*Hình 13. Giao diện nhập thông tin cá nhân*

*Hình 14. Giao diện nhập thông tin học bạ*

*Hình 15. Giao diện nhập nguyện vọng*

*Hình 16. Giao diện theo dõi trạng thái hồ sơ*

*Hình 17. Giao diện trang chủ admin*

*Hình 18. Giao diện tìm kiếm thí sinh*

*Hình 19. Giao diện danh sách thí sinh*

*Hình 20. Giao diện xem chi tiết hồ sơ thí sinh*

*Hình 21. Giao diện yêu cầu chỉnh sửa thông tin thí sinh*

## LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn Thầy ThS. Vũ Quang Dũng. Trong quá trình thực hiện đề tài, nhờ sự hỗ trợ, tận tình chỉ dạy, hướng dẫn chi tiết và tạo điều kiện để em có thể hoàn thành đề tài xây dựng ứng dụng. Nhờ đó em có thêm kiến thức cơ sở làm nền tảng vững chắc để em có thể đi sâu hơn vào chuyên ngành em đã chọn và để giúp ích cho xã hội.

Em cũng xin chân thành cảm ơn thầy cô, cán bộ giảng viên trường Đại học Phenikaa đã tận tình truyền đạt kiến thức, dạy dỗ bảo ban em từ những ngày đầu bõ ngõ bước chân vào giảng đường Đại học, luôn tạo điều kiện tốt nhất để em có thể học hỏi, tích luỹ kiến thức, kỹ năng để hoàn thành đồ án cơ sở.

Do chưa có nhiều kinh nghiệm cũng như những hạn chế về kiến thức thực tế, trong bài báo cáo này chắc chắn sẽ không tránh khỏi những thiếu sót. Em rất mong có thể nhận được sự nhận xét, và sự góp ý từ phía Thầy để bài báo cáo của chúng em có thể hoàn thiện tốt hơn.

Lời cuối cùng, em rất biết ơn vì đã có cơ hội được học hỏi và làm việc dưới sự hướng dẫn của Thầy. Em xin kính chúc Thầy có nhiều sức khỏe, đạt được nhiều thành công và hạnh phúc trong cuộc sống.

**Chúng em xin chân thành cảm ơn!**

## **LỜI CAM ĐOAN**

Chúng em xin cam đoan báo cáo đồ án cơ sở là của chúng em, do chúng em thực hiện dưới sự hướng dẫn của ThS.Vũ Quang Dũng. Tất cả tham khảo, nghiên cứu và tài liệu liên quan đều được nêu rõ ràng và chi tiết trong danh mục tài liệu tham khảo. Các nội dung trình bày trong báo cáo này là hoàn toàn trung thực.

## CHƯƠNG I. GIỚI THIỆU

### A. Đặt vấn đề

Trong thời đại công nghệ số hiện nay, việc ứng dụng công nghệ thông tin vào lĩnh vực giáo dục ngày càng trở nên quan trọng và cần thiết. Tuy nhiên, quá trình tuyển sinh truyền thống vẫn còn tồn tại nhiều hạn chế như mất nhiều thời gian, tốn kém chi phí, dễ xảy ra sai sót trong khâu nhập liệu và quản lý hồ sơ. Ngoài ra, người học ở xa hoặc bận rộn cũng gặp nhiều khó khăn khi phải trực tiếp đến cơ sở đào tạo để nộp hồ sơ và hoàn tất thủ tục.

Đứng trước những thách thức đó, **Hệ thống tuyển sinh trực tuyến** ra đời như một giải pháp tối ưu, giúp đơn giản hóa quy trình, nâng cao hiệu quả quản lý và tạo điều kiện thuận lợi cho cả nhà trường lẫn thí sinh.

Hệ thống tuyển sinh trực tuyến mang lại nhiều lợi ích đáng kể, bao gồm:

- **Tính linh hoạt và tiện lợi:** Thí sinh có thể đăng ký, nộp hồ sơ và theo dõi kết quả ở bất kỳ đâu, chỉ cần có thiết bị kết nối Internet.  
**Tự động hóa quy trình:** Giúp nhà trường dễ dàng quản lý, lưu trữ và tra cứu dữ liệu, giảm thiểu sai sót trong khâu xử lý thông tin.
- **Tiết kiệm thời gian và chi phí:** Cả thí sinh và nhà trường đều tiết kiệm đáng kể chi phí in ấn, đi lại và nhân sự.
- **Minh bạch và nhanh chóng:** Quá trình xét tuyển và công bố kết quả được thực hiện tự động, đảm bảo tính công bằng và rõ ràng.
- **Dễ dàng mở rộng và tích hợp:** Có thể kết nối với các hệ thống khác như quản lý sinh viên, quản lý học phí, hoặc cổng thông tin đào tạo.

Những lợi ích này cho thấy việc xây dựng và triển khai **hệ thống tuyển sinh trực tuyến** là bước đi cần thiết để hiện đại hóa công tác quản lý giáo dục, nâng cao chất lượng phục vụ và đáp ứng xu thế chuyển đổi số trong giáo dục.

## B. Các giải pháp đã có

Hiện nay, có nhiều **hệ thống tuyển sinh trực tuyến** được triển khai tại các trường đại học và cơ sở giáo dục, mỗi hệ thống đều có đặc điểm và tính năng riêng. Dưới đây là một số ví dụ tiêu biểu:

### 1. Hệ thống tuyển sinh trực tuyến của Bộ Giáo dục và Đào tạo (moet.gov.vn)

**Ưu điểm:**

- Tích hợp dữ liệu toàn quốc, cho phép thí sinh đăng ký, điều chỉnh nguyện vọng trực tuyến.
- Đảm bảo tính bảo mật và minh bạch cao.
- Dễ dàng tra cứu kết quả và đối chiếu thông tin.
- **Hạn chế:**
  - Giao diện còn khá phức tạp, chưa thân thiện với người dùng mới.
  - Hệ thống thường quá tải vào các thời điểm cao điểm tuyển sinh.

### 2. Hệ thống tuyển sinh của Đại học Quốc gia Hà Nội(tuyensinh.vnu.edu.vn)

**Ưu điểm:**

- Giao diện trực quan, dễ sử dụng.
- Cho phép nộp hồ sơ, theo dõi trạng thái xét tuyển và nhận thông báo tự động.
- Tích hợp nhiều hình thức xét tuyển khác nhau.
- **Hạn chế:**
  - Chỉ áp dụng nội bộ trong hệ thống trường thành viên.
  - Cần tài khoản xác thực mới có thể sử dụng đầy đủ chức năng.

### 3. Hệ thống tuyển sinh của Đại học FPT (admission.fpt.edu.vn)

**Ưu điểm**

- Quy trình đăng ký nhanh, đơn giản, hỗ trợ trực tuyến 24/7.
- Cho phép thanh toán lệ phí và tải hồ sơ ngay trên hệ thống.
- Hỗ trợ tư vấn và tra cứu kết quả nhanh chóng.

#### **Hạn chế:**

- Giao diện thiên về marketing, đôi khi làm giảm tính tập trung của người dùng.
- Không mở rộng cho các trường khác, chỉ phục vụ hệ thống FPT.

#### **4. Cổng thông tin tuyển sinh của các trường cao đẳng và trung cấp (ví dụ: tuyensinhdaihoc.vn, tuyensinhso.vn)**

#### **Ưu điểm:**

- Tổng hợp thông tin tuyển sinh đa dạng từ nhiều trường.
- Hỗ trợ tìm kiếm, so sánh ngành học và nộp hồ sơ trực tuyến.

#### **Hạn chế:**

- Nhiều nội dung quảng cáo, gây nhiễu thông tin.
- Không phải cổng chính thức của trường, độ tin cậy không cao.

### **C. Giải pháp đề xuất**

Hệ thống tuyển sinh trực tuyến được xây dựng nhằm hỗ trợ quá trình đăng ký và quản lý thông tin thí sinh một cách thuận tiện và hiệu quả hơn. Ứng dụng giúp giảm bớt các thủ tục giấy tờ, tiết kiệm thời gian cho cả thí sinh và cán bộ tuyển sinh.

Hiện tại, hệ thống bao gồm các chức năng chính:

- **Đăng ký và lưu thông tin thí sinh:** Thí sinh có thể nhập thông tin cá nhân và thông tin học tập trực tiếp trên hệ thống.

- **Chọn nguyện vọng:** Hỗ trợ thí sinh lựa chọn ngành học hoặc trường đăng ký theo mong muốn.
- **Quản lý và duyệt hồ sơ:** Quản trị viên có thể xem danh sách hồ sơ, kiểm tra tính hợp lệ, **duyệt hoặc yêu cầu thí sinh chỉnh sửa** khi phát hiện sai sót.
- **Lưu trữ dữ liệu tập trung:** Toàn bộ thông tin được lưu trữ trong cơ sở dữ liệu, thuận tiện cho việc tìm kiếm và thống kê.

Hệ thống hướng đến mục tiêu **đơn giản hóa quy trình tuyển sinh, nâng cao hiệu quả quản lý**, và là **bước đầu trong việc chuyển đổi số công tác tuyển sinh** của nhà trường.

## **CHƯƠNG II. THIẾT KẾ VÀ TRIỂN KHAI**

### **A. Các yêu cầu chức năng**

#### **1. Đối với các thí sinh (Users).**

##### **- Đăng nhập**

###### **1. Mở đăng nhập:**

Người dùng truy cập theo đường link được sử dụng để đăng nhập vào hệ thống. Thông thường, họ sẽ truy cập một trang đăng nhập và điền thông tin đăng nhập của họ. Chức năng đăng nhập cho phép người dùng truy cập vào hệ thống tuyển sinh trực tuyến bằng tài khoản đã được đăng ký trước đó. Hệ thống sử dụng Firebase Authentication để xác thực tài khoản, đảm bảo an toàn và bảo mật thông tin người dùng.

###### **2. Nhập thông tin đăng nhập:**

Người dùng điền thông tin đăng nhập vào các trường thích hợp. Thông tin đăng nhập bao gồm tài khoản (số CCCD hoặc email) và mật khẩu.

###### **3. Gửi yêu cầu xác thực:**

Khi nhấn nút “Đăng nhập”, thông tin đăng nhập được gửi đến Firebase Authentication để xử lý.

###### **4. Xác minh thông tin đăng nhập:**

Firebase kiểm tra tính hợp lệ của tài khoản và mật khẩu trong cơ sở dữ liệu người dùng đã đăng ký:

- Nếu thông tin hợp lệ → Hệ thống cấp mã thông báo (token) xác thực cho phiên làm việc.

- Nếu không hợp lệ → Firebase trả về thông báo lỗi (ví dụ: sai mật khẩu, tài khoản không tồn tại).

## 5. Thông báo kết quả đăng nhập:

Dựa trên kết quả kiểm tra, hệ thống thông báo kết quả cho người dùng. Nếu đăng nhập thành công, họ sẽ được chuyển đến giao diện làm việc hoặc trang chính của ứng dụng. Trong trường hợp đăng nhập không thành công, sẽ nhận thông báo lỗi cụ thể để người dùng kiểm tra và nhập lại.

### **- Đăng ký tài khoản:**

Hệ thống cho phép người dùng tạo tài khoản mới để truy cập và sử dụng các chức năng của hệ thống tuyển sinh trực tuyến. Quá trình đăng ký được thực hiện thông qua Firebase Authentication, đảm bảo tính bảo mật và xác thực người dùng chính xác.

Quy trình cụ thể như sau:

#### 1. Truy cập giao diện đăng ký:

Người dùng chọn mục “Đăng ký” trên trang Đăng nhập. Lúc này hệ thống sẽ hiển thị trang Đăng ký trên giao diện.

#### 2. Nhập thông tin cá nhân:

Người dùng nhập đầy đủ các trường thông tin bắt buộc bao gồm Họ và tên, email, số điện thoại và mật khẩu đăng ký. Khi nhập, những trường nào nhập không hợp lệ sẽ báo lỗi dưới trường thông tin nhập. Chỉ khi người dùng nhập đúng các trường thông tin thì mới có thể tiếp tục bước tiếp theo để đăng ký.

#### 3. Gửi yêu cầu đăng ký:

Sau khi điền thông tin, người dùng nhấn nút “Đăng ký”. Hệ thống gửi những dữ liệu này đến Firebase Authentication để tạo tài khoản mới và lưu thông tin người dùng vào cơ sở dữ liệu (Firestore Database).

#### 4. Xác thực email:

Firebase sẽ tự động gửi email xác minh đến địa chỉ email mà người dùng đã đăng ký. Người dùng cần truy cập hộp thư và nhấn vào liên kết xác nhận để hoàn tất quá trình đăng ký.

#### 5. Kích hoạt tài khoản:

Sau khi xác minh thành công, tài khoản được kích hoạt. Người dùng có thể đăng nhập và sử dụng đầy đủ các chức năng của hệ thống.

#### **- *Điền thông tin cá nhân, điểm học bạ và nguyện vọng:***

Chức năng cho phép thí sinh nhập đầy đủ các thông tin cá nhân như họ tên, ngày sinh, giới tính, CCCD, email, số điện thoại, địa chỉ, trường THPT, năm tốt nghiệp, cùng với điểm học bạ và tối đa ba nguyện vọng xét tuyển.

Toàn bộ dữ liệu được lưu trữ trên Firebase, đảm bảo tính bảo mật, đồng bộ và khả năng truy xuất cao.

Chức năng này giúp thí sinh dễ dàng quản lý hồ sơ của mình trực tuyến, đồng thời hỗ trợ admin trong việc theo dõi, kiểm tra và duyệt thông tin một cách nhanh chóng và chính xác.

#### **- *Theo dõi trạng thái hồ sơ(Đã duyệt/ Cần chỉnh sửa):***

Chức năng cho phép thí sinh theo dõi trạng thái xử lý hồ sơ của mình sau khi gửi thông tin. Hệ thống sẽ hiển thị rõ ràng trạng thái hiện tại như “Đã duyệt” hoặc “Cần chỉnh sửa” dựa trên kết quả xem xét của admin.

Dữ liệu trạng thái được cập nhật trực tiếp trên Firebase, giúp đảm bảo thông tin phản hồi nhanh chóng và chính xác. Nhờ đó, thí sinh có thể dễ dàng biết

được tình trạng hồ sơ để kịp thời điều chỉnh khi cần, đồng thời hỗ trợ admin quản lý và phản hồi hiệu quả hơn.

## 2. Đối với bên nhà trường (Admins)

Hệ thống cung cấp cho Admin (phía bên nhà trường) một giao diện riêng để quản lý toàn bộ dữ liệu của thí sinh. Các chức năng được thiết kế nhằm hỗ trợ quá trình theo dõi, kiểm tra và duyệt hồ sơ một cách hiệu quả, đảm bảo tính chính xác và minh bạch trong quá trình tuyển sinh.

### - Xem danh sách thí sinh:

Admin có thể truy cập danh sách toàn bộ thí sinh đã đăng ký và gửi hồ sơ trực tuyến. Danh sách hiển thị các thông tin cơ bản như họ tên, CCCD, email, số điện thoại, nguyện vọng.

Dữ liệu được lưu trữ và đồng bộ tự động trên Firebase, giúp đảm bảo rằng mọi thay đổi đều được cập nhật theo thời gian thực, tránh tình trạng sai lệch thông tin giữa các bên.

### Lợi ích:

- Giúp admin quản lý tập trung và thống nhất dữ liệu.
- Tiết kiệm thời gian tổng hợp thông tin.
- Dễ dàng theo dõi tình trạng đăng ký của toàn bộ thí sinh.

### - Tìm kiếm thí sinh:

Chức năng tìm kiếm hỗ trợ admin nhanh chóng tra cứu thông tin thí sinh dựa trên nhiều tiêu chí như họ tên, CCCD, số điện thoại hoặc nguyện vọng. Công cụ này đặc biệt hữu ích khi số lượng thí sinh lớn, giúp admin tiết kiệm thời gian và giảm thiểu thao tác thủ công.

### Lợi ích:

- Nâng cao hiệu suất làm việc của admin.
- Dễ dàng truy cập và đổi chiểu thông tin khi cần.

- Giảm sai sót trong quá trình quản lý dữ liệu.

#### **- Xem chi tiết hồ sơ:**

Admin có thể mở từng hồ sơ cụ thể để xem toàn bộ thông tin chi tiết, bao gồm thông tin cá nhân, học bạ, và tối đa 3 nguyện vọng mà thí sinh đã đăng ký. Từ đó, admin có thể đánh giá mức độ chính xác và đầy đủ của dữ liệu, đồng thời đối chiếu với tiêu chí tuyển sinh của trường.

#### **Lợi ích:**

- Giúp admin có cái nhìn toàn diện về từng thí sinh.
- Hạn chế sai sót khi xét duyệt hồ sơ.
- Tạo cơ sở dữ liệu minh bạch phục vụ quá trình tuyển sinh.

#### **- Duyệt hồ sơ:**

Sau khi xác minh thông tin hợp lệ, admin có thể duyệt hồ sơ để xác nhận tính hợp lệ của đăng ký. Khi hồ sơ được duyệt, hệ thống sẽ cập nhật trạng thái sang “Đã duyệt” trên Firebase và hiển thị kết quả trực tiếp cho thí sinh.

#### **Lợi ích:**

- Tự động hóa quy trình xử lý hồ sơ.
- Giảm thời gian phản hồi giữa admin và thí sinh.
- Giúp quá trình xét tuyển minh bạch, rõ ràng.

#### **- Yêu cầu chỉnh sửa:**

Nếu phát hiện thông tin thiếu hoặc sai lệch, admin có thể gửi yêu cầu chỉnh sửa đến thí sinh. Hệ thống sẽ tự động chuyển trạng thái hồ sơ sang “Cần chỉnh sửa”, đồng thời gửi thông báo cụ thể về lỗi cần khắc phục.

#### **Lợi ích:**

- Giúp đảm bảo tính chính xác của dữ liệu đầu vào.

- Cho phép thí sinh dễ dàng cập nhật lại hồ sơ.
- Giảm rủi ro sai thông tin trong quá trình xét tuyển.

## B. Các yêu cầu phi chức năng

### 2.2.1. *Hiệu suất:*

Hiệu suất là một yếu tố then chốt trong hệ thống tuyển sinh trực tuyến. Hệ thống cần đảm bảo khả năng xử lý nhanh, phản hồi kịp thời và hoạt động ổn định trong suốt quá trình người dùng thao tác. Cụ thể, các thao tác như đăng nhập, điền thông tin cá nhân – học bạ – nguyện vọng, tải dữ liệu và hiển thị danh sách thí sinh cần được thực hiện nhanh chóng, không gây ra độ trễ đáng kể. Hệ thống sử dụng Firebase để lưu trữ và truy xuất dữ liệu, giúp giảm thời gian phản hồi, đảm bảo tính ổn định và hiệu quả ngay cả khi có nhiều người dùng truy cập cùng lúc.

### 2.2.2. *Tính sẵn sàng:*

Hệ thống phải đảm bảo khả năng truy cập mọi lúc, mọi nơi, giúp thí sinh có thể đăng ký và theo dõi hồ sơ tuyển sinh bất kỳ thời điểm nào. Tính sẵn sàng cao được đảm bảo thông qua việc lưu trữ và quản lý dữ liệu trên nền tảng điện toán đám mây Firebase, giúp hệ thống duy trì hoạt động liên tục, kể cả trong các khung giờ cao điểm. Điều này đặc biệt quan trọng trong giai đoạn cao điểm của kỳ tuyển sinh, khi lượng người truy cập đồng thời tăng mạnh.

### 2.2.3. *Tính bảo mật:*

Bảo mật là yếu tố quan trọng trong hệ thống, vì hệ thống lưu trữ nhiều thông tin cá nhân quan trọng như CCCD, số điện thoại, email, học bạ và nguyện vọng của thí sinh. Hệ thống sử dụng Firebase Authentication để xác thực người dùng qua email, đảm bảo chỉ người dùng hợp lệ mới có thể truy cập vào tài khoản của mình. Bên cạnh đó, dữ liệu được mã hóa và phân quyền truy cập rõ ràng, tránh tình trạng truy cập trái phép hoặc rò rỉ thông tin. Cơ chế xác thực và kiểm soát quyền hạn giữa thí sinh và admin được thiết kế riêng biệt, đảm bảo tính riêng tư và an toàn dữ liệu.

### 2.2.4. *Tính đáng tin cậy:*

Hệ thống cần duy trì hoạt động ổn định và chính xác trong mọi thời điểm để không ảnh hưởng đến quá trình đăng ký và xét tuyển. Các chức năng như lưu thông tin hồ sơ, cập nhật trạng thái, duyệt hồ sơ và phản hồi giữa admin – thí sinh phải được đảm bảo hoạt động chính xác, tránh lỗi mất dữ liệu hoặc gián đoạn quá trình xử lý.

Nhờ tích hợp Firebase, dữ liệu được đồng bộ thời gian thực (real-time), giúp hệ thống hạn chế rủi ro khi xảy ra sự cố. Ngoài ra, việc sao lưu dữ liệu thường xuyên giúp dễ dàng phục hồi và khắc phục lỗi, đảm bảo tính liên tục của dịch vụ.

### ***2.2.5. Tiêu chí thiết kế và tương tác người dùng***

Trong hệ thống tuyển sinh trực tuyến, thiết kế giao diện và khả năng tương tác đóng vai trò quan trọng trong việc nâng cao trải nghiệm người dùng, giúp thí sinh và admin có thể sử dụng hệ thống một cách dễ dàng, thuận tiện và hiệu quả.

Các yếu tố về bố cục, màu sắc, và cấu trúc điều hướng đều được xem xét kỹ lưỡng nhằm đảm bảo tính trực quan, nhất quán và thân thiện cho mọi đối tượng sử dụng.

#### **\* Thanh Menu (Dành cho thí sinh):**

Thanh menu là một thành phần quan trọng giúp người dùng, đặc biệt là thí sinh, dễ dàng điều hướng trong hệ thống và truy cập các chức năng cần thiết.

Trong giao diện dành cho thí sinh, thanh menu bao gồm các mục chính sau:

- **Hướng dẫn đăng ký:** Cung cấp các bước chi tiết để thí sinh có thể thực hiện đăng ký hồ sơ xét tuyển một cách dễ dàng, tránh sai sót.
- **Tin tức:** Hiển thị các thông báo, tin tuyển sinh, và lịch xét tuyển mới nhất giúp thí sinh nắm bắt kịp thời các thông tin quan trọng.
- **Đăng ký xét tuyển:** Cho phép thí sinh truy cập vào biểu mẫu đăng ký, nhập thông tin cá nhân, lựa chọn nguyện vọng và gửi hồ sơ trực tuyến.
- **Đăng xuất:** Cho phép thí sinh thoát khỏi tài khoản sau khi hoàn tất quá trình đăng ký hoặc tra cứu thông tin, đảm bảo an toàn và bảo mật dữ liệu cá nhân.

Thanh menu này được bố trí ở vị trí dễ nhìn (thường là phía trên cùng của trang), giúp người dùng nhanh chóng truy cập các chức năng chính chỉ với vài thao tác. Giao diện thanh menu cần rõ ràng, phân chia hợp lý, và có biểu tượng trực quan để hỗ trợ thao tác nhanh hơn.

\* **Giao diện thiết kế:**

- **Sử dụng màu sắc đơn giản:** Màu chủ đạo nên nhẹ nhàng, đồng bộ với phong cách của trang web, giúp người dùng dễ dàng tập trung vào nội dung chính.
- **Tối ưu cho trải nghiệm người dùng:** Các nút bấm, biểu mẫu và thanh tìm kiếm được bố trí hợp lý, thuận tiện cho người dùng thao tác mà không cần cuộn trang quá nhiều.
- **Tương thích đa nền tảng:** Giao diện cần hiển thị tốt trên cả máy tính và điện thoại di động, giúp thí sinh có thể truy cập hệ thống mọi lúc, mọi nơi.
- **Thông nhất phong cách thiết kế:** Màu sắc, font chữ và biểu tượng cần đồng nhất giữa các trang để tạo cảm giác chuyên nghiệp và thân thiện.

\* **Lợi ích:**

- Giúp thí sinh dễ dàng sử dụng và tiết kiệm thời gian khi thao tác trên hệ thống.
- Cải thiện trải nghiệm người dùng, giảm tỷ lệ sai sót trong quá trình đăng ký.  
Nâng cao hình ảnh chuyên nghiệp của hệ thống xét tuyển trực tuyến.
- Tăng tính hiệu quả và trực quan, giúp cả người dùng mới cũng có thể nhanh chóng làm quen và thao tác thuận lợi.

## 2.2.6. *Khả năng mở rộng*

Khả năng mở rộng là yếu tố quan trọng trong quá trình thiết kế và phát triển ứng dụng, giúp hệ thống có thể đáp ứng tốt sự gia tăng về số lượng người dùng, dữ liệu và chức năng trong tương lai. Ứng dụng cần được xây dựng theo kiến trúc linh hoạt, dễ dàng nâng cấp, mở rộng quy mô hoặc tích hợp thêm các mô-đun, dịch vụ mà không gây ảnh hưởng đến hiệu suất và tính ổn định chung.

### Cụ thể:

- **Mở rộng về người dùng:** Hệ thống phải có khả năng phục vụ đồng thời nhiều thí sinh và quản trị viên truy cập mà vẫn đảm bảo tốc độ xử lý nhanh, không bị gián đoạn.
- **Mở rộng về tính năng:** Ứng dụng được thiết kế theo hướng mô-đun (module-based), giúp việc thêm mới hoặc nâng cấp các chức năng như thống kê, phân tích dữ liệu, hoặc gửi thông báo tự động trở nên dễ dàng.
- **Mở rộng về dữ liệu:** Khi số lượng hồ sơ và thông tin thí sinh tăng lên, hệ thống vẫn cần đảm bảo hiệu suất truy xuất, lưu trữ và tìm kiếm nhanh chóng.
- **Khả năng tích hợp:** Ứng dụng có thể tích hợp với các hệ thống khác như cổng thông tin tuyển sinh, hệ thống quản lý đào tạo, hoặc nền tảng thanh toán điện tử nếu cần thiết, giúp tối ưu hóa quy trình làm việc và chia sẻ dữ liệu linh hoạt.

### \* Lợi ích:

- Giúp ứng dụng **phát triển lâu dài** và đáp ứng được **nhu cầu mở rộng trong tương lai**.
- Giảm chi phí và thời gian **bảo trì, nâng cấp hệ thống**.
- **Tăng khả năng thích ứng** với sự thay đổi của công nghệ và nhu cầu người dùng.
- Đảm bảo **trải nghiệm người dùng ổn định** ngay cả khi quy mô sử dụng tăng cao.

## 2.2.7. Quản lý lỗi và tương thích

- **Quản lý lỗi:** Cần thiết kế quy trình quản lý lỗi để đảm bảo khi có lỗi xảy ra, hệ thống cung cấp thông báo lỗi rõ ràng và hướng dẫn người dùng để giải quyết vấn đề một cách nhanh chóng. Cần có cơ chế thông báo lỗi chi tiết và dễ hiểu cho người dùng.
- **Tương thích:** Ứng dụng cần hoạt động trơn tru trên các thiết bị và máy tính cá nhân khác nhau, bao gồm các hệ điều hành và kích thước màn hình khác nhau. Điều này giúp đảm bảo rằng người dùng có thể truy cập và sử dụng ứng dụng mà không gặp phải vấn đề tương thích.

## C. Các ràng buộc

- **Quản lý lỗi:**

Hệ thống cần được thiết kế với cơ chế quản lý lỗi rõ ràng và hiệu quả nhằm đảm bảo quá trình sử dụng của người dùng không bị gián đoạn. Khi có lỗi xảy ra, ứng dụng phải cung cấp thông báo lỗi chi tiết, dễ hiểu và hướng dẫn cụ thể để người dùng có thể nhanh chóng khắc phục hoặc thử lại thao tác.

Việc quản lý lỗi tốt không chỉ giúp người dùng hiểu rõ nguyên nhân sự cố mà còn hỗ trợ đội ngũ phát triển trong quá trình phát hiện, ghi log và xử lý lỗi hệ thống. Các lỗi quan trọng cần được lưu lại trong cơ sở dữ liệu hoặc hệ thống log để phục vụ việc kiểm tra, phân tích và nâng cấp hệ thống sau này.

- **Tính tương thích:**

Ứng dụng cần đảm bảo khả năng tương thích cao với nhiều nền tảng, thiết bị và môi trường khác nhau để người dùng có thể truy cập và sử dụng một cách thuận tiện. Cụ thể, ứng dụng phải hoạt động ổn định trên các trình duyệt web phổ biến (như Chrome, Edge, Firefox, Safari) và nhiều kích thước màn hình khác nhau, bao gồm máy tính, laptop, máy tính bảng và điện thoại di động.

Bên cạnh đó, việc đảm bảo tương thích với nhiều hệ điều hành khác nhau (Windows, macOS, Android, iOS, Linux, v.v.) giúp ứng dụng tiếp cận được nhiều đối tượng người dùng hơn, đồng thời duy trì giao diện và hiệu suất ổn định trên mọi nền tảng.

### **3. Các ràng buộc về triển khai**

#### **\* Nền tảng và tương thích:**

Ứng dụng được phát triển dưới dạng website, sử dụng các công nghệ nền tảng HTML, CSS và JavaScript, cho phép chạy trực tiếp trên trình duyệt web mà không cần cài đặt.

Hệ thống có khả năng hoạt động tốt trên nhiều thiết bị khác nhau như máy tính cá nhân, laptop, máy tính bảng và điện thoại di động, đảm bảo tính đa nền tảng và tương thích cao với các trình duyệt phổ biến như Chrome, Edge, và Firefox.

#### **\* Ngôn ngữ và công cụ phát triển:**

Ứng dụng được phát triển bằng các ngôn ngữ lập trình web cơ bản gồm:

- **HTML**: xây dựng cấu trúc giao diện.
- **CSS**: định dạng và thiết kế giao diện người dùng (UI).
- **JavaScript**: xử lý logic, tương tác người dùng và kết nối dữ liệu.
- Công cụ hỗ trợ phát triển gồm:
  - **Visual Studio Code** làm môi trường lập trình chính.
  - **Git và GitHub** được sử dụng để quản lý mã nguồn, cộng tác và kiểm soát phiên bản (commit, push, pull request).

#### **\* Môi trường máy chủ và cơ sở dữ liệu:**

Hệ thống sử dụng Firebase làm nền tảng lưu trữ và xử lý dữ liệu:

- Firebase Authentication: xác thực người dùng (đăng ký, đăng nhập, xác thực email).
- Firebase Firestore: lưu trữ thông tin cá nhân, học bạ và nguyện vọng của thí sinh.
- Dữ liệu được lưu trữ trên hạ tầng đám mây của Google, đảm bảo tốc độ truy xuất nhanh và tính sẵn sàng cao.

#### \* **Bảo mật thông tin cá nhân:**

Ứng dụng áp dụng nhiều biện pháp bảo mật để bảo vệ thông tin của người dùng:

- Xác thực người dùng qua Firebase Auth.
- Mã hóa dữ liệu khi lưu trữ và truyền tải.

Điều này giúp đảm bảo tính riêng tư, toàn vẹn và an toàn dữ liệu.

#### \* **Ngôn ngữ giao diện:**

Ứng dụng được phát triển bằng **tiếng Việt**, giúp người dùng dễ dàng sử dụng. Trong tương lai, hệ thống có thể mở rộng hỗ trợ **đa ngôn ngữ** bằng cách tích hợp **Google Translate API** hoặc các gói **localization** để phục vụ người dùng quốc tế.

#### \* **Khả năng mở rộng:**

Ứng dụng được thiết kế với cấu trúc **module hóa**, dễ dàng mở rộng thêm các tính năng mới như:

- Quản lý thông báo kết quả.
- Tự động gửi email xác nhận.
- Hệ thống báo cáo và thống kê cho admin.

Việc mở rộng không ảnh hưởng đến hiệu suất hoặc trải nghiệm người dùng hiện tại.

#### \* **Dự phòng và khôi phục sau sự cố:**

Firebase tự động cung cấp cơ chế sao lưu và đồng bộ dữ liệu theo thời gian thực, giúp hạn chế mất mát dữ liệu. Trong trường hợp xảy ra sự cố, dữ liệu có thể được khôi phục thông qua bản sao lưu (backup) trên nền tảng đám mây. Cơ chế này giúp đảm bảo tính liên tục hoạt động và độ tin cậy cao cho ứng dụng.

## 4. Các ràng buộc kinh tế

#### \* Chi phí và công cụ:

Hệ thống được phát triển với mục tiêu tối ưu chi phí và tận dụng công nghệ miễn phí:

- Ứng dụng hoàn toàn miễn phí cho người dùng, không yêu cầu thanh toán khi đăng ký hoặc sử dụng.
- Quá trình phát triển sử dụng các công nghệ mã nguồn mở và miễn phí, bao gồm HTML, CSS, JavaScript và Firebase.
- Firebase cung cấp gói miễn phí (Free Tier) cho xác thực người dùng, lưu trữ dữ liệu và triển khai web.  
Mã nguồn dự án được lưu trữ và quản lý trên GitHub, tận dụng dịch vụ miễn phí để làm việc nhóm và kiểm soát phiên bản.
- Công cụ lập trình chính là Visual Studio Code, một IDE miễn phí và phổ biến, giúp giảm chi phí phần mềm phát triển.

#### \* Dịch vụ và tính năng:

- Ứng dụng cung cấp toàn bộ tính năng cốt lõi miễn phí, bao gồm: đăng ký, đăng nhập, điền thông tin cá nhân, họcば, nguyện vọng và theo dõi trạng thái hồ sơ.
- Trong tương lai, có thể mở rộng thêm các dịch vụ trả phí hoặc gói nâng cao như: xuất báo cáo tự động, thông kê tuyển sinh, hoặc gửi email xác nhận hàng loạt.
- Hệ thống được thiết kế linh hoạt để dễ dàng tích hợp thêm tính năng thương mại mà không cần thay đổi lớn trong cấu trúc hiện tại.

## 5. Các ràng buộc về đạo đức

#### Bảo mật dữ liệu người dùng:

- Hệ thống cam kết bảo vệ an toàn dữ liệu cá nhân của thí sinh, bao gồm thông tin cá nhân, học bạ và nguyện vọng xét tuyển.
- Tất cả dữ liệu được lưu trữ và quản lý thông qua Firebase, với các cơ chế bảo mật như xác thực người dùng, phân quyền truy cập và mã hóa dữ liệu.
- Hệ thống không chia sẻ thông tin người dùng với bất kỳ bên thứ ba nào nếu không có sự cho phép rõ ràng từ người dùng.
- Người dùng có toàn quyền sử dụng các chức năng xác thực, thay đổi mật khẩu và quản lý tài khoản để bảo vệ thông tin cá nhân của mình.

#### **Quyền riêng tư:**

- Ứng dụng tôn trọng và bảo vệ quyền riêng tư của người dùng.
- Các chính sách và quy trình bảo mật được áp dụng nhằm đảm bảo rằng thông tin cá nhân của thí sinh không bị truy cập, tiết lộ hoặc sử dụng sai mục đích.
- Hệ thống chỉ thu thập những thông tin cần thiết cho quá trình đăng ký và xét tuyển, đồng thời cung cấp tùy chọn để người dùng kiểm soát và chỉnh sửa dữ liệu của mình khi cần.

#### **Trách nhiệm về nội dung:**

- Nhà phát triển và quản trị viên hệ thống **chịu trách nhiệm về toàn bộ nội dung, dữ liệu và thông tin hiển thị trên ứng dụng.**
- Mọi thông tin liên quan đến tuyển sinh, hướng dẫn hoặc thông báo đều được kiểm duyệt và cập nhật thường xuyên để đảm bảo tính **chính xác, minh bạch và đáng tin cậy.**
- Ứng dụng không chứa hoặc lan truyền nội dung sai lệch, gây hiểu nhầm hay vi phạm quy định pháp luật.

#### **2.3.4. Các ràng buộc về độ tin cậy**

Trong quá trình phát triển ứng dụng học từ vựng tiếng anh, độ tin cậy của hệ thống đóng một vai trò tối quan trọng. Việc đảm bảo rằng ứng dụng hoạt động một cách ổn định và an toàn không chỉ đảm bảo sự hài lòng của khách hàng mà còn bảo vệ thông tin quan trọng và tạo cơ sở cho hoạt động hiệu quả của kho.

Các ràng buộc về độ tin cậy đã được xem xét kỹ lưỡng và triển khai trong quá trình phát triển, với những yếu tố quan trọng như bảo mật dữ liệu, tích hợp mô hình tính năng tương đương, và khả năng kiểm tra và sửa lỗi liên tục.

Hệ thống đã tích hợp các công nghệ bảo mật để đảm bảo tính an toàn trong việc truyền tải dữ liệu qua mạng. Các chủ thích hình ảnh và danh mục hình ảnh cũng đã được tạo để giúp người dùng dễ dàng tìm kiếm và điều hướng trong ứng dụng. Đặc biệt, chúng ta đã xây dựng cơ sở dữ liệu ổn định và hiệu quả, cùng với việc sử dụng các công cụ giám sát để đảm bảo rằng hệ thống hoạt động một cách liên tục và đáng tin cậy.

Điều này đảm bảo rằng ứng dụng không chỉ cung cấp một trải nghiệm dễ dàng sử dụng mà còn đáp ứng các tiêu chuẩn cao về độ tin cậy, an toàn và bảo mật, giúp tạo nền tảng cho sự thành công của siêu thị và sự hài lòng của khách hàng.

### CHƯƠNG III. MÔ HÌNH HỆ THỐNG VÀ THIẾT KẾ GIẢI PHÁP

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

Các Actors của hệ thống:

Actor	Mô tả
Users(Thí sinh)	Là người sử dụng hệ thống để thực hiện các chức năng như đăng ký tuyển sinh, cập nhật thông tin cá nhân, xem kết quả xét tuyển và theo dõi trạng thái hồ sơ của mình.
Admin(Nhà trường)	Là người chịu trách nhiệm quản lý toàn bộ dữ liệu trong hệ thống. Admin có thể phê duyệt, chỉnh sửa, xóa hoặc cập nhật thông tin hồ sơ của thí sinh, đồng thời theo dõi và thống kê dữ liệu tuyển sinh.

##### 1. Đăng ký

**Actor:** Người dùng (Thí sinh)

##### Mô tả:

Use case này cho phép thí sinh tạo tài khoản mới để sử dụng hệ thống tuyển sinh trực tuyến. Quá trình đăng ký được thực hiện thông qua **Firebase Authentication**, đảm bảo tính bảo mật và xác thực người dùng.

##### Điều kiện bắt đầu (Entry Condition):

- Người dùng mở trang web hoặc ứng dụng tuyển sinh.
- Người dùng chưa có tài khoản trong hệ thống.

##### Điều kiện kết thúc (Exit Condition):

- Tài khoản mới được tạo thành công trên Firebase Authentication.
- Thông tin tài khoản (email, tên, ID người dùng,...) được lưu vào cơ sở dữ liệu Firebase Realtime Database hoặc Firestore.
- Người dùng nhận được thông báo đăng ký thành công và có thể đăng nhập ngay.

## **Luồng sự kiện chính (Flow of Events):**

1. Người dùng chọn chức năng “**Đăng ký**” trên giao diện chính.
2. Hệ thống hiển thị **form đăng ký**.
3. Người dùng nhập thông tin cần thiết: họ tên, email, mật khẩu và xác nhận mật khẩu.
4. Người dùng nhấn nút “**Đăng ký**”.
5. Hệ thống gửi yêu cầu tạo tài khoản đến **Firebase Authentication**.
6. Firebase kiểm tra tính hợp lệ của email và mật khẩu (ví dụ: email đúng định dạng, mật khẩu ≥ 6 ký tự).
7. Nếu hợp lệ, Firebase tạo tài khoản mới và trả về **UID** (User ID).
8. Hệ thống lưu thông tin bổ sung (họ tên, số điện thoại, vai trò,...) vào **Firebase Database** với UID vừa tạo.
9. Hệ thống hiển thị thông báo “**Đăng ký thành công**” và chuyển hướng đến trang đăng nhập hoặc trang chính.

## **Luồng thay thế (Alternative Flow):**

- Nếu email đã tồn tại trên Firebase → Hệ thống hiển thị thông báo “Email này đã được đăng ký”.
- Nếu mật khẩu không đủ mạnh hoặc nhập sai định dạng → Hiển thị thông báo lỗi cụ thể và yêu cầu người dùng nhập lại.

## **2. Đăng nhập(Dành cho thí sinh)**

**Actor:** Người dùng (Thí sinh)

### **Mô tả:**

Use case này cho phép thí sinh đăng nhập vào hệ thống tuyển sinh để truy cập các chức năng như điền thông tin cá nhân, học bạ, nguyện vọng và theo dõi trạng thái hồ sơ. Việc xác thực người dùng được thực hiện thông qua Firebase Authentication, đảm bảo tính bảo mật và an toàn dữ liệu.

### **Điều kiện bắt đầu (Entry Condition):**

- Người dùng mở trang web hoặc ứng dụng tuyển sinh.
- Người dùng đã có tài khoản hợp lệ trên hệ thống.

### **Điều kiện kết thúc (Exit Condition):**

Người dùng đăng nhập thành công và được chuyển hướng đến **trang menu chính** của hệ thống.

## **Luồng sự kiện chính (Flow of Events):**

1. Người dùng truy cập vào ứng dụng.
2. Hệ thống hiển thị form đăng nhập.
3. Người dùng nhập email và mật khẩu, hoặc chọn đăng nhập nhanh bằng Google.
4. Người dùng nhấn nút “Đăng nhập”.
5. Hệ thống gửi thông tin xác thực đến Firebase Authentication.
6. Firebase kiểm tra tính hợp lệ của tài khoản (email, mật khẩu hoặc tài khoản Google).
7. Nếu thông tin hợp lệ, Firebase trả về thông tin xác thực người dùng (UID).
8. Hệ thống lấy thông tin người dùng từ Firebase Database và khởi tạo phiên làm việc (session).
9. Người dùng được điều hướng đến thanh menu chính của ứng dụng.

## **Luồng thay thế (Alternative Flow):**

- Nếu email hoặc mật khẩu không chính xác → Hệ thống hiển thị thông báo lỗi “Tài khoản hoặc mật khẩu không đúng”.
- Nếu tài khoản chưa được xác minh qua email → Hệ thống yêu cầu người dùng xác minh trước khi đăng nhập.
- Nếu đăng nhập Google thất bại → Hiển thị thông báo lỗi “Đăng nhập bằng Google không thành công, vui lòng thử lại”.

### **3. *Điền thông tin cá nhân, học bạ và nguyện vọng.***

Actor: Người dùng.

Entry Condition: Người dùng mở ứng dụng.

Exit Condition: Người dùng đăng nhập vào ứng dụng hoặc hoàn tất quy trình đổi mật khẩu.

Flow of Events:

1. Người dùng truy cập vào ứng dụng.
2. Người dùng chọn một trong các tùy chọn: đổi mật khẩu, hoặc quên mật khẩu.
  - o Nếu chọn Đổi mật khẩu:
    1. Người dùng đăng nhập vào ứng dụng.

2. Chọn phần thông tin.
3. Người dùng nhập email hoặc tên tài khoản để yêu cầu đổi mật khẩu.
4. Hệ thống gửi liên kết hoặc mã xác thực đến email của người dùng.
5. Người dùng nhập mã xác thực hoặc truy cập liên kết để thiết lập mật khẩu mới.
6. Hệ thống cập nhật mật khẩu mới và thông báo cho người dùng.
7. Người dùng được điều hướng tới thanh Menu chính.
  - o Nếu chọn Quên mật khẩu:
    1. Người dùng nhập email hoặc tên tài khoản để yêu cầu khôi phục mật khẩu.
    2. Hệ thống gửi liên kết hoặc mã xác thực đến email của người dùng.
    3. Người dùng nhập mã xác thực hoặc truy cập liên kết để tạo mật khẩu mới.
    4. Hệ thống cập nhật mật khẩu mới và thông báo cho người dùng.
    5. Người dùng được điều hướng tới thanh Menu chính.

#### **4. Đăng nhập (Dành cho admin).**

**Actor:** Admin

**Mô tả:**

Use case này cho phép Admin đăng nhập vào hệ thống quản lý thí sinh. Tài khoản Admin được hệ thống tạo sẵn bằng JavaScript, không thông qua Firebase Authentication. Sau khi đăng nhập thành công, Admin có thể truy cập các chức năng như xem danh sách thí sinh, tìm kiếm, duyệt hồ sơ và gửi yêu cầu chỉnh sửa.

**Điều kiện bắt đầu (Entry Condition):**

- Admin mở trang web hoặc ứng dụng quản trị.
- Admin có thông tin tài khoản hợp lệ (được khai báo trực tiếp trong mã JavaScript, ví dụ trong biến adminAccount).

**Điều kiện kết thúc (Exit Condition):**

- Admin đăng nhập thành công và được chuyển hướng đến giao diện quản trị (“Trang chủ Admin”).

### **Luồng sự kiện chính (Flow of Events):**

1. Admin truy cập vào ứng dụng quản trị.
2. Hệ thống hiển thị form đăng nhập (gồm trường “Tên đăng nhập” và “Mật khẩu”).
3. Admin nhập thông tin đăng nhập.
4. Admin nhấn nút “Đăng nhập”.
5. Hệ thống nhận dữ liệu và tiến hành đối chiếu với tài khoản admin được khai báo sẵn trong JavaScript
6. Hệ thống kiểm tra xem tên đăng nhập và mật khẩu có trùng khớp với adminAccount hay không.
7. Nếu hợp lệ, hệ thống khởi tạo phiên làm việc (session/localStorage).
8. Hệ thống chuyển hướng Admin đến giao diện “Trang chủ Admin”.

### **Luồng thay thế (Alternative Flow):**

1. Tên đăng nhập hoặc mật khẩu không đúng: Nếu tên đăng nhập hoặc mật khẩu không trùng khớp với tài khoản admin:
  - Hệ thống hiển thị thông báo: **“Tên đăng nhập hoặc mật khẩu không đúng”**.
  - Admin được yêu cầu nhập lại.
2. Bỏ trống trường thông tin: Nếu Admin để trống một trong hai trường và nhấn “Đăng nhập”:
  - Hệ thống hiển thị thông báo: **“Vui lòng nhập đầy đủ thông tin”**.

### **5. Theo dõi trạng thái hồ sơ**

**Actor:** Người dùng (Thí sinh)

**Mô tả:** Cho phép thí sinh xem trạng thái hiện tại của hồ sơ mà mình đã nộp.

**Điều kiện bắt đầu (Entry Condition):**

- Thí sinh đã đăng nhập vào hệ thống.

#### **Điều kiện kết thúc (Exit Condition):**

- Thí sinh thấy được trạng thái hồ sơ của mình (ví dụ: Mới nộp, Đã duyệt, Yêu cầu bồi sung, Bị từ chối).

#### **Luồng sự kiện chính (Flow of Events):**

1. Thí sinh đăng nhập vào trang chính.
2. Hệ thống tự động tải dữ liệu HoSoTuyenSinh của thí sinh đó từ Firebase Database.
3. Hệ thống đọc giá trị của trường trangThai trong hồ sơ.
4. Hiển thị trạng thái này ở vị trí nổi bật trên giao diện (ví dụ: "Trạng thái hồ sơ: **Đã duyệt**").

#### **Luồng thay thế (Alternative Flow):**

- Nếu thí sinh chưa nộp hồ sơ (chưa có dữ liệu HoSoTuyenSinh) → Hiển thị thông báo "Bạn chưa nộp hồ sơ" và nút kêu gọi hành động "Nộp hồ sơ ngay".

## **6. Đăng xuất**

**Actor:** Người dùng (Thí sinh, Admin)

**Mô tả:** Cho phép người dùng kết thúc phiên làm việc một cách an toàn.

#### **Điều kiện bắt đầu (Entry Condition):**

- Người dùng đang đăng nhập trong hệ thống.

#### **Điều kiện kết thúc (Exit Condition):**

- Phiên làm việc bị hủy.
- Người dùng được chuyển hướng về trang Đăng nhập.

#### **Luồng sự kiện chính (Flow of Events):**

1. Người dùng nhấn vào nút "Đăng xuất" (thường nằm trong menu tài khoản).
2. Hệ thống gọi hàm signOut() của Firebase Authentication.
3. Firebase chấm dứt phiên làm việc và xóa token xác thực cục bộ (local storage).

4. Hệ thống chuyển hướng người dùng về trang Đăng nhập.

## 7. Xem danh sách thí sinh

**Actor:** Admin

**Mô tả:** Hiển thị cho Admin một danh sách tổng quan tất cả các hồ sơ thí sinh đã nộp vào hệ thống.

**Điều kiện bắt đầu (Entry Condition):**

- Admin đã đăng nhập thành công vào trang quản trị.

**Điều kiện kết thúc (Exit Condition):**

- Hệ thống hiển thị một bảng (table) danh sách các hồ sơ.

**Luồng sự kiện chính (Flow of Events):**

1. Admin nhấp vào mục "Quản lý Hồ sơ" hoặc "Danh sách Thí sinh" trên thanh menu.
2. Hệ thống gửi truy vấn đến Firebase Database để lấy về toàn bộ danh sách HoSoTuyenSinh.
3. Hệ thống hiển thị dữ liệu ra một bảng, bao gồm các cột chính (ví dụ: Họ tên, Email, Trạng thái, Ngày nộp).
4. (Tùy chọn) Hệ thống cho phép lọc nhanh (ví dụ: chỉ xem hồ sơ "Mới nộp") hoặc phân trang nếu có quá nhiều hồ sơ.

**Luồng thay thế (Alternative Flow):**

2. Nếu chưa có hồ sơ nào trong hệ thống → Hiển thị thông báo "Chưa có dữ liệu".

## 8. Tìm kiếm thí sinh

**Actor:** Admin

**Mô tả:** Cho phép Admin tìm kiếm nhanh một hồ sơ cụ thể.

**Điều kiện bắt đầu (Entry Condition):**

- Admin đang ở trang "Xem danh sách thí sinh".

**Điều kiện kết thúc (Exit Condition):**

- Danh sách thí sinh được lọc lại theo từ khóa tìm kiếm.

### **Luồng sự kiện chính (Flow of Events):**

1. Admin nhập từ khóa (ví dụ: họ tên, email, hoặc CCCD) vào ô tìm kiếm.
2. Admin nhấn Enter hoặc nút "Tìm".
3. Hệ thống thực hiện truy vấn lọc (query) trên Firebase Database dựa trên từ khóa.
4. Hệ thống cập nhật lại bảng danh sách chỉ hiển thị các kết quả phù hợp.

### **Luồng thay thế (Alternative Flow):**

3. Nếu không tìm thấy kết quả nào → Hiển thị thông báo "Không tìm thấy hồ sơ phù hợp".

## **9. Xem chi tiết hồ sơ thí sinh**

**Actor:** Admin

**Mô tả:** Admin xem toàn bộ thông tin chi tiết của một thí sinh cụ thể.

### **Điều kiện bắt đầu (Entry Condition):**

- Admin đang ở trang "Xem danh sách thí sinh".

### **Điều kiện kết thúc (Exit Condition):**

- Hiển thị trang chi tiết hồ sơ của thí sinh đó.

### **Luồng sự kiện chính (Flow of Events):**

4. Admin nhấn vào nút "Xem chi tiết" (hoặc nhấn vào tên) của một thí sinh trong danh sách.
5. Hệ thống lấy hoSoId (hoặc userId) của thí sinh đó.
6. Hệ thống điều hướng sang trang "Chi tiết hồ sơ" và truyền hoSoId.
7. Trang chi tiết truy vấn toàn bộ thông tin của HoSoTuyenSinh đó (bao gồm thông tin cá nhân, học bạ, và danh sách các NguyenVong liên quan).
8. Hệ thống hiển thị tất cả thông tin này một cách rõ ràng.

## **10. Duyệt hồ sơ**

**Actor:** Admin

**Mô tả:** Admin ra quyết định (duyệt, từ chối, yêu cầu bổ sung) đối với một hồ sơ.

### **Điều kiện bắt đầu (Entry Condition):**

- Admin đang ở trang "Xem chi tiết hồ sơ thí sinh".
- Hồ sơ đang ở trạng thái "Mới nộp".

### **Điều kiện kết thúc (Exit Condition):**

- Trường trangThai của hồ sơ trong Firebase Database được cập nhật.

### **Luồng sự kiện chính (Flow of Events):**

1. Sau khi xem xét kỹ thông tin, Admin tìm đến khu vực "Hành động".
2. Admin nhấp vào một trong các nút: "Duyệt" (Approve) hoặc "Yêu cầu bổ sung" (Request Change).
3. Hệ thống hiển thị hộp thoại xác nhận (ví dụ: "Bạn có chắc muốn duyệt hồ sơ này?").
4. Admin xác nhận.
5. Hệ thống gửi lệnh cập nhật lên Firebase Database, thay đổi giá trị của trangThai (ví dụ: thành da\_duyet hoặc yeu\_cau\_bo\_sung).
6. Hệ thống hiển thị thông báo "Cập nhật trạng thái thành công!".
7. Trạng thái mới của hồ sơ sẽ được cập nhật trên trang chi tiết và trang danh sách.

### **Luồng thay thế (Alternative Flow):**

Nếu Admin chọn "Yêu cầu bổ sung" → Hệ thống có thể (tùy chọn) hiển thị một ô để Admin nhập lý do/ghi chú (ví dụ: "Thiếu ảnh học bạ"). Ghi chú này sẽ được lưu lại và hiển thị cho thí sinh.

## **11. Yêu cầu chỉnh sửa hồ sơ**

**Actor:** Admin

### **Mô tả:**

Use case này cho phép Admin truy cập vào hệ thống quản lý thông tin thí sinh để truy cập các chức năng như duyệt hồ sơ, xem và xóa.

### **Điều kiện bắt đầu (Entry Condition):**

- Admin mở trang web hoặc ứng dụng quản lý thông tin sinh viên.

### **Điều kiện kết thúc (Exit Condition):**

- Admin đăng xuất khỏi hệ thống quản lý.

## **Luồng sự kiện chính (Flow of Events):**

1. Admin truy cập vào ứng dụng.
2. Hệ thống hiển thị danh sách thông tin của thí sinh.
3. Admin tìm kiếm hoặc xem chi tiết thông tin sinh viên.
4. Admin ấn nút “Xem”.
5. Hệ thống gửi yêu cầu tải chi tiết hồ sơ của thí sinh đến Firebase Authentication.
6. Firebase nhận yêu cầu và trả về bảng thông tin của thí sinh.
7. Admin chọn “Tù chối hồ sơ”.
8. Hệ thống yêu cầu Admin nhập lí do từ chối và trường thông tin cần được chỉnh sửa.
9. Admin nhập các thông tin được yêu cầu.
10. Hệ thống thông báo đã gửi lí do từ chối và khóa các trường thông tin không cần chỉnh sửa.

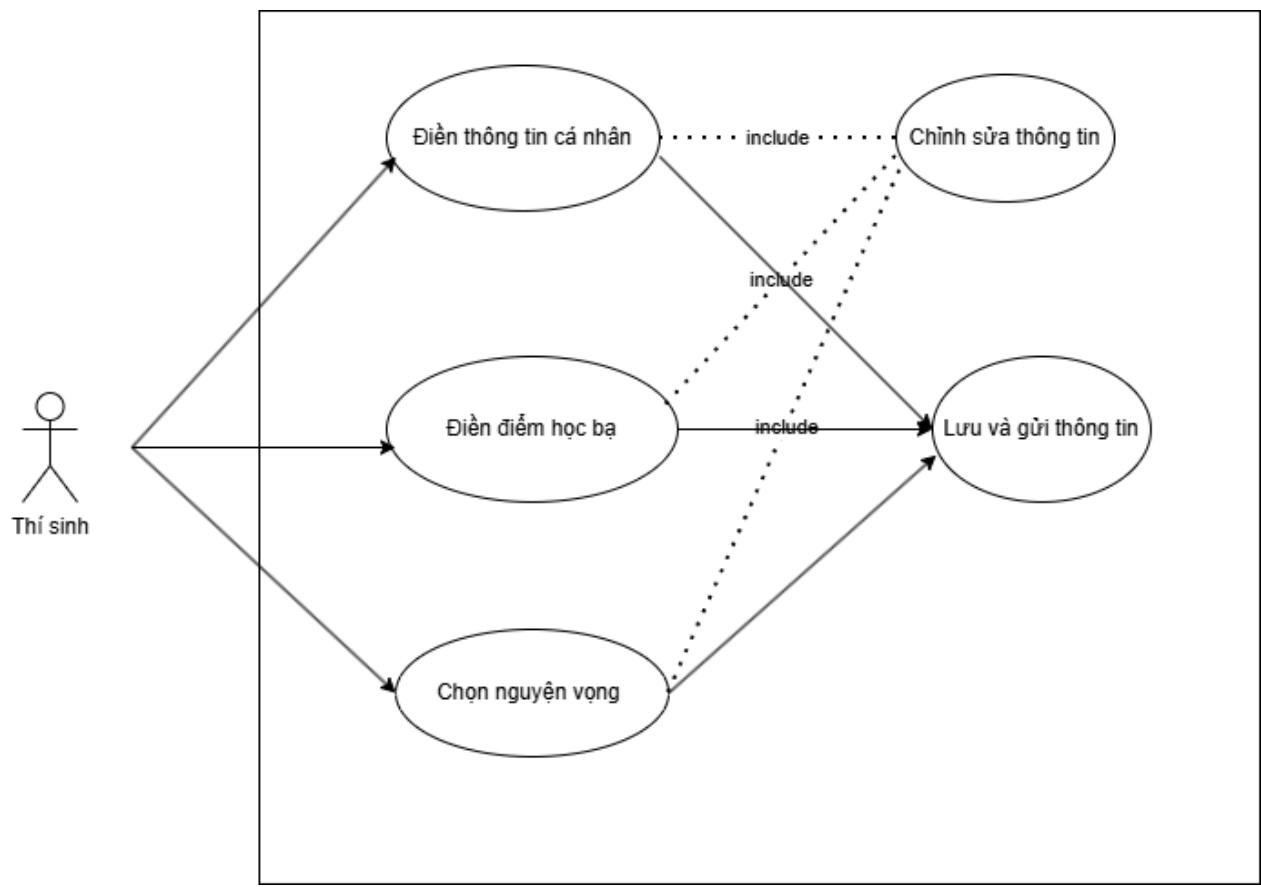
## **B. Mô hình Use-case**

### **1. Mô hình Use-case tổng quát**



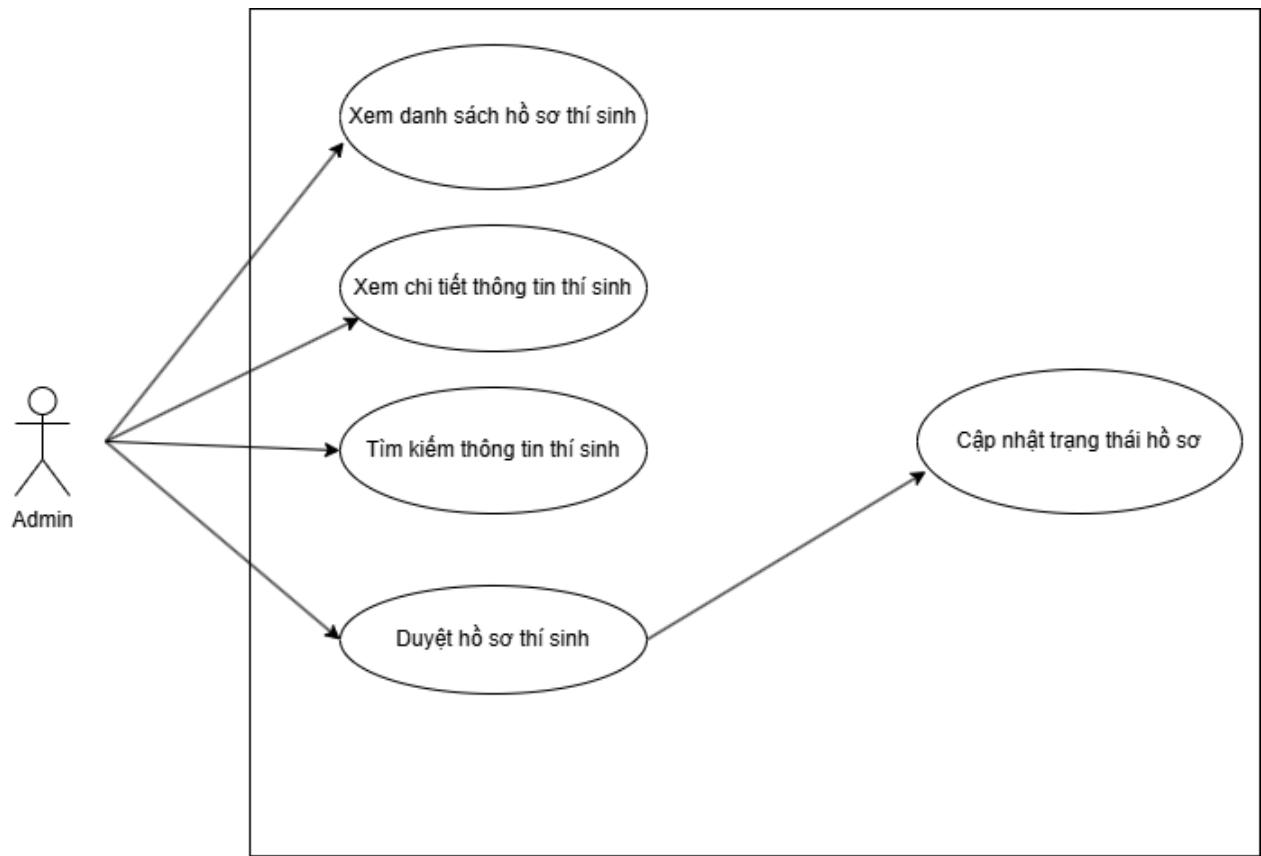
*Hình 1. Sơ đồ tổng quát Use-case*

## 2. Mô hình Use-case thí sinh



Hình 2. Mô hình Use-case thí sinh

### 3. Mô hình Use-case admin

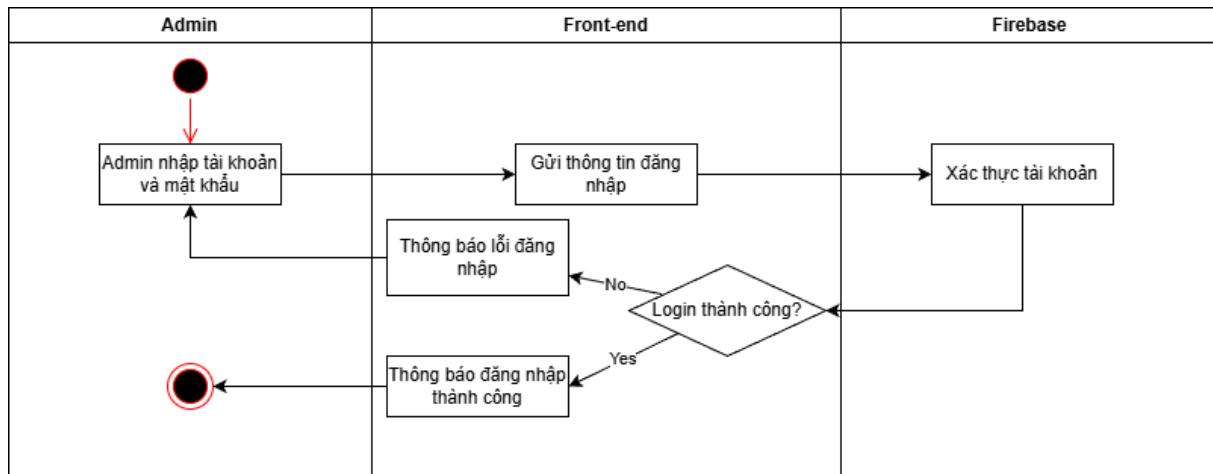


*Hình 3. Mô hình Use-case admin*

## C. Mô hình hoạt động

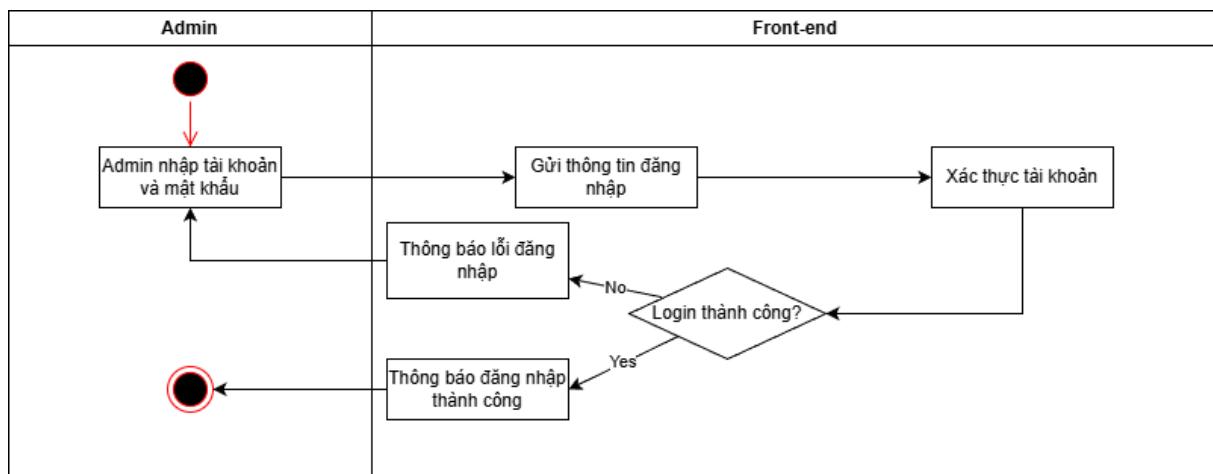
### 1. Đăng nhập

#### 1.1. Đăng nhập (Thí sinh)



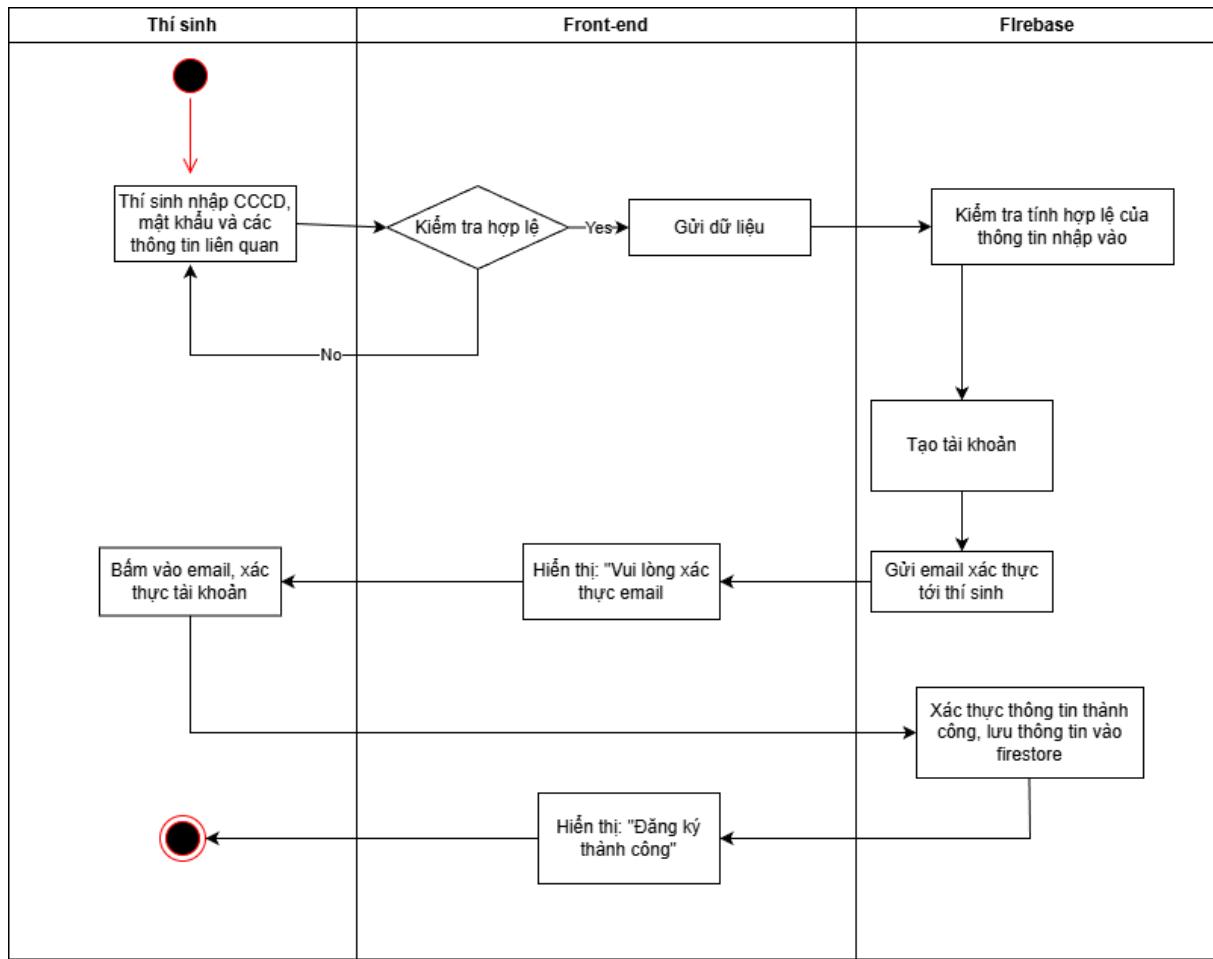
Hình 4. Biểu đồ hoạt động đăng nhập của thí sinh

#### 1.2. Đăng nhập (Admin)



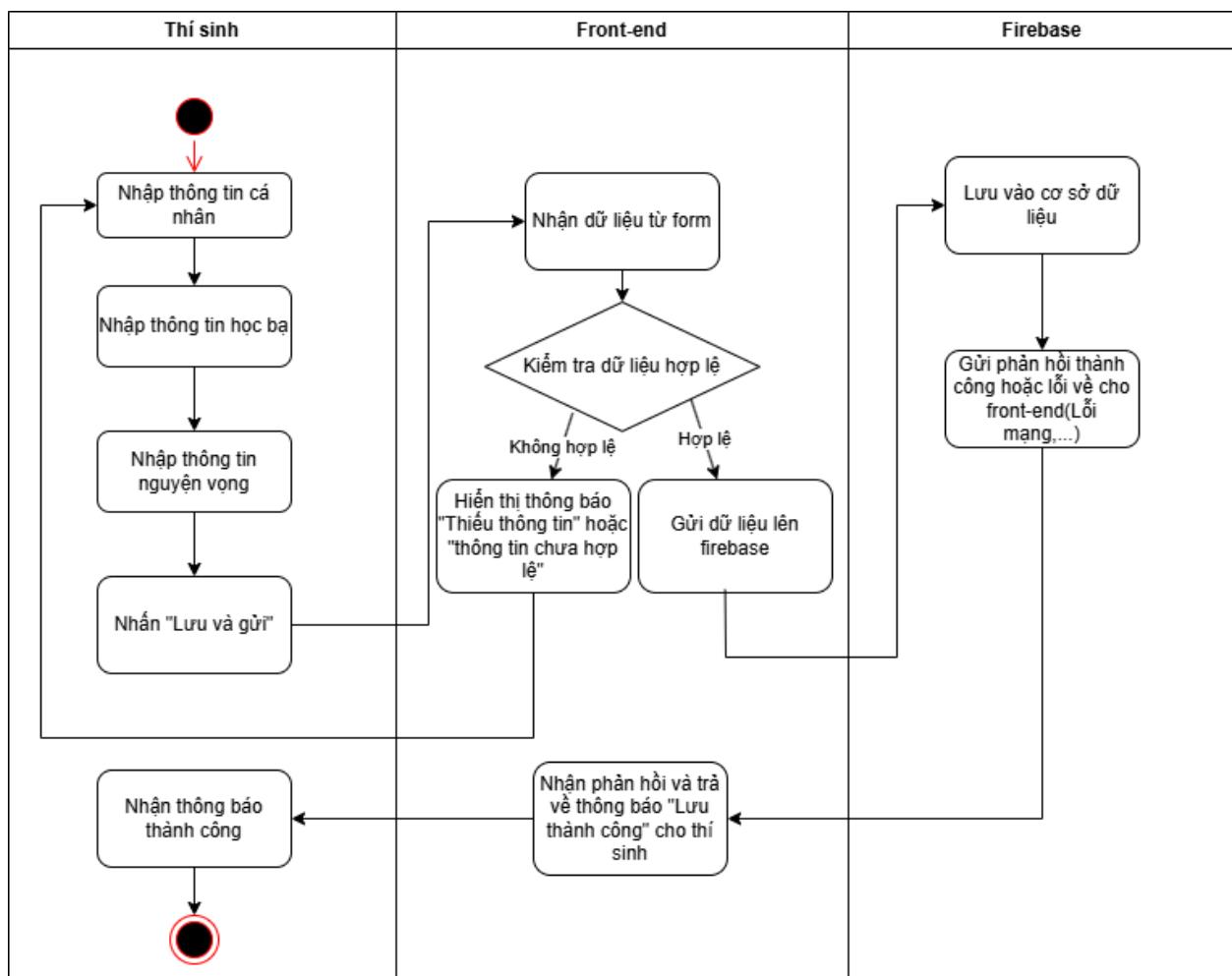
Hình 5. Biểu đồ hoạt động đăng nhập của admin

## 2. Đăng ký



Hình 6. Biểu đồ hoạt động đăng ký của thí sinh

### 3. Nộp thông tin tuyển sinh



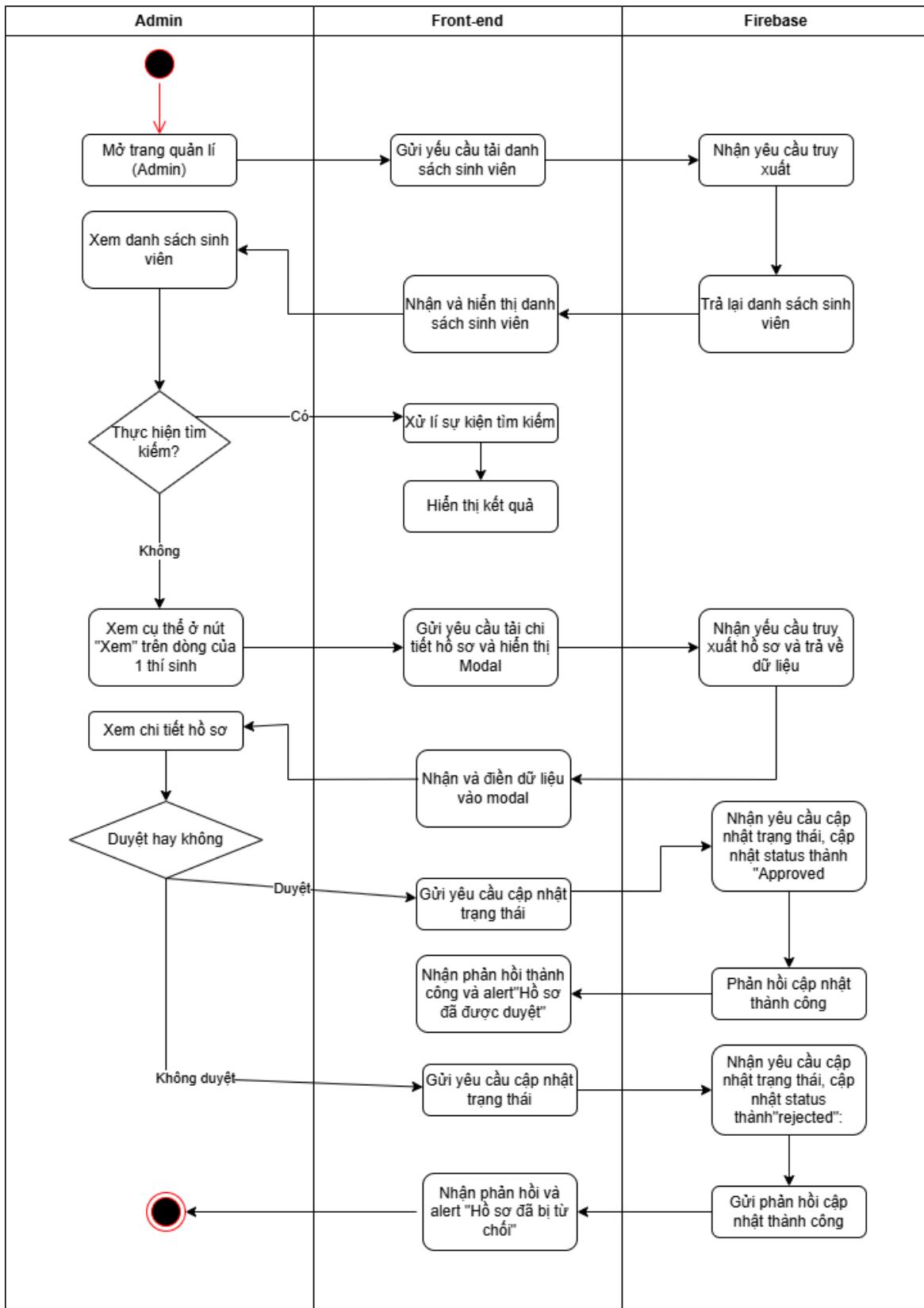
Hình 7. Biểu đồ hoạt động nộp thông tin thí sinh

#### 4. Theo dõi trạng thái hồ sơ



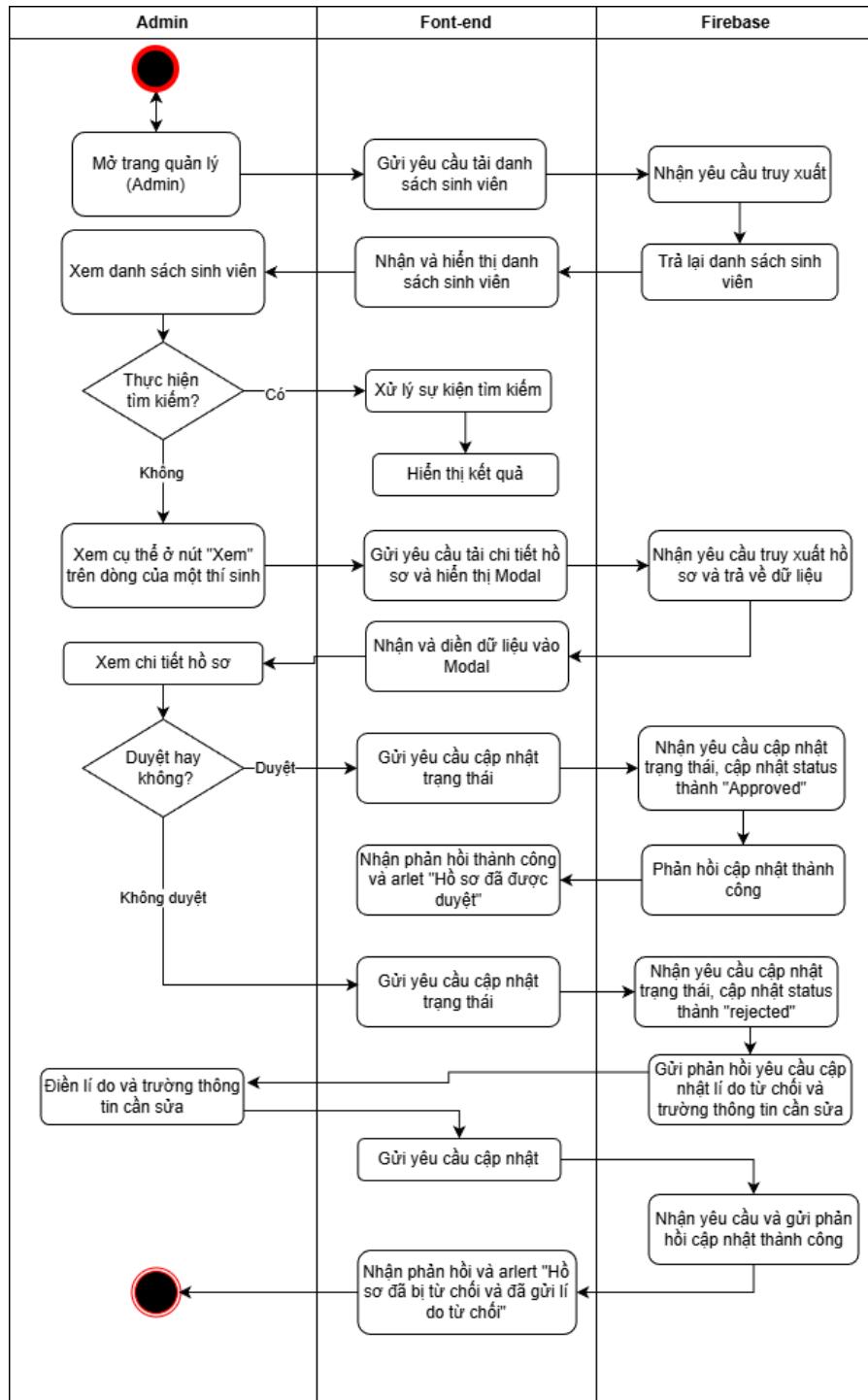
Hình 8. Biểu đồ hoạt động theo dõi trạng thái hồ sơ

## 5. Duyệt hồ sơ (Admin)



Hình 9. Biểu đồ hoạt động duyệt hồ sơ

## 6. Gửi yêu cầu chỉnh sửa (Admin)



Hình 10. Biểu đồ hoạt động gửi yêu cầu

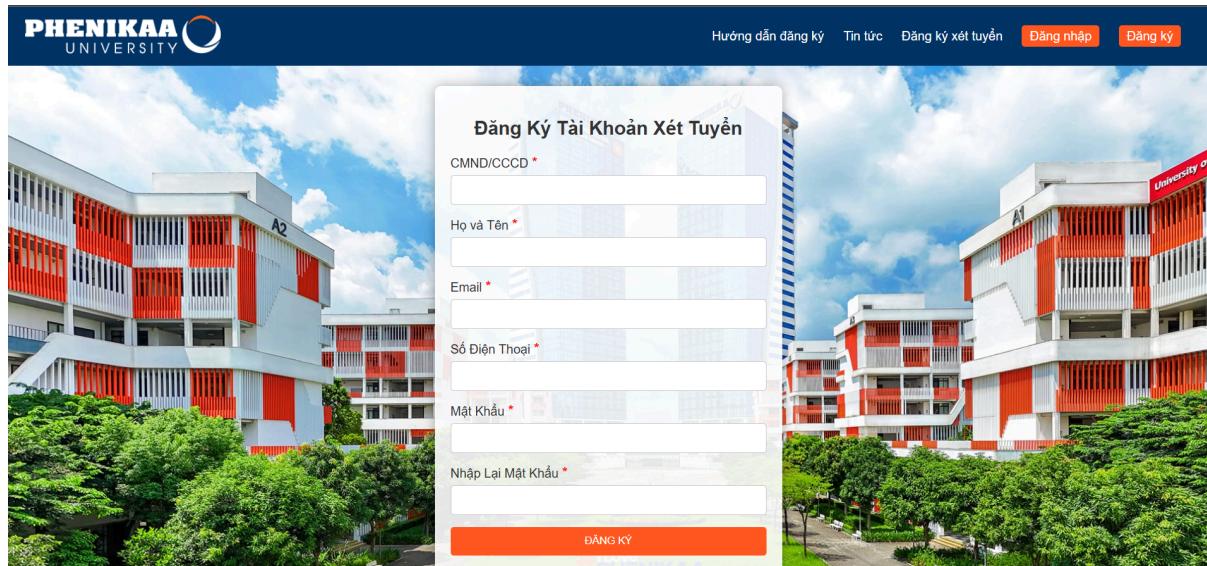
## D. Các màn hình giao diện

### 1. Giao diện đăng nhập

đăng ký tài khoản' (If you don't have an account, click [register account](#)). There are also links for 'QUÊN MẬT KHẨU?' (Forgot Password?) and 'PHENIKAA UNIVERSITY'." data-bbox="116 159 875 417"/>

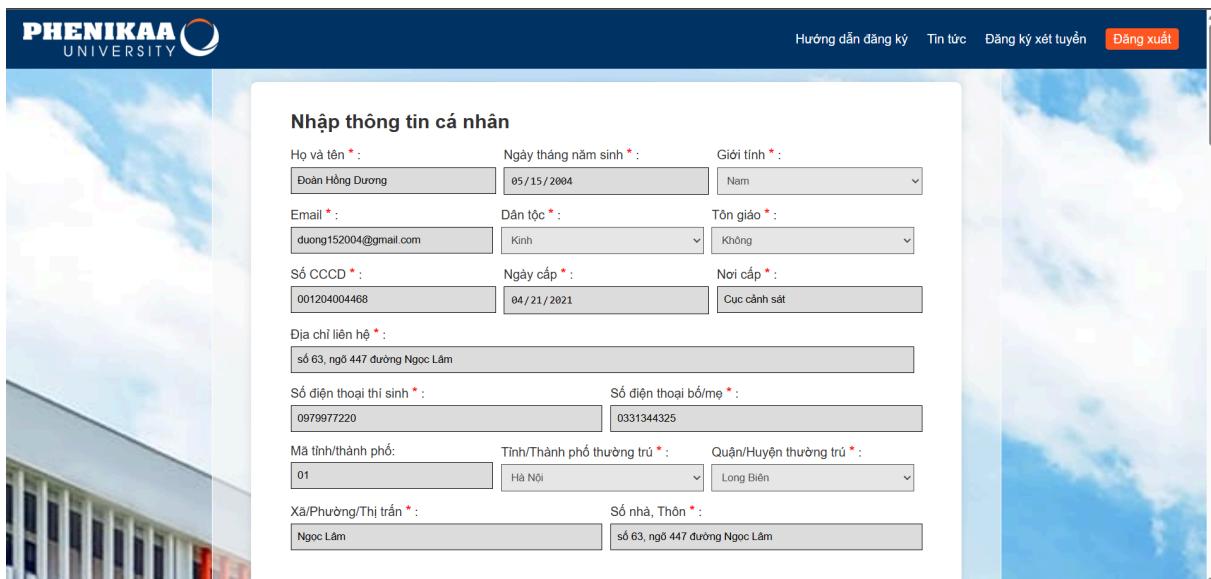
Hình 11. Giao diện đăng nhập

### 2. Giao diện đăng ký



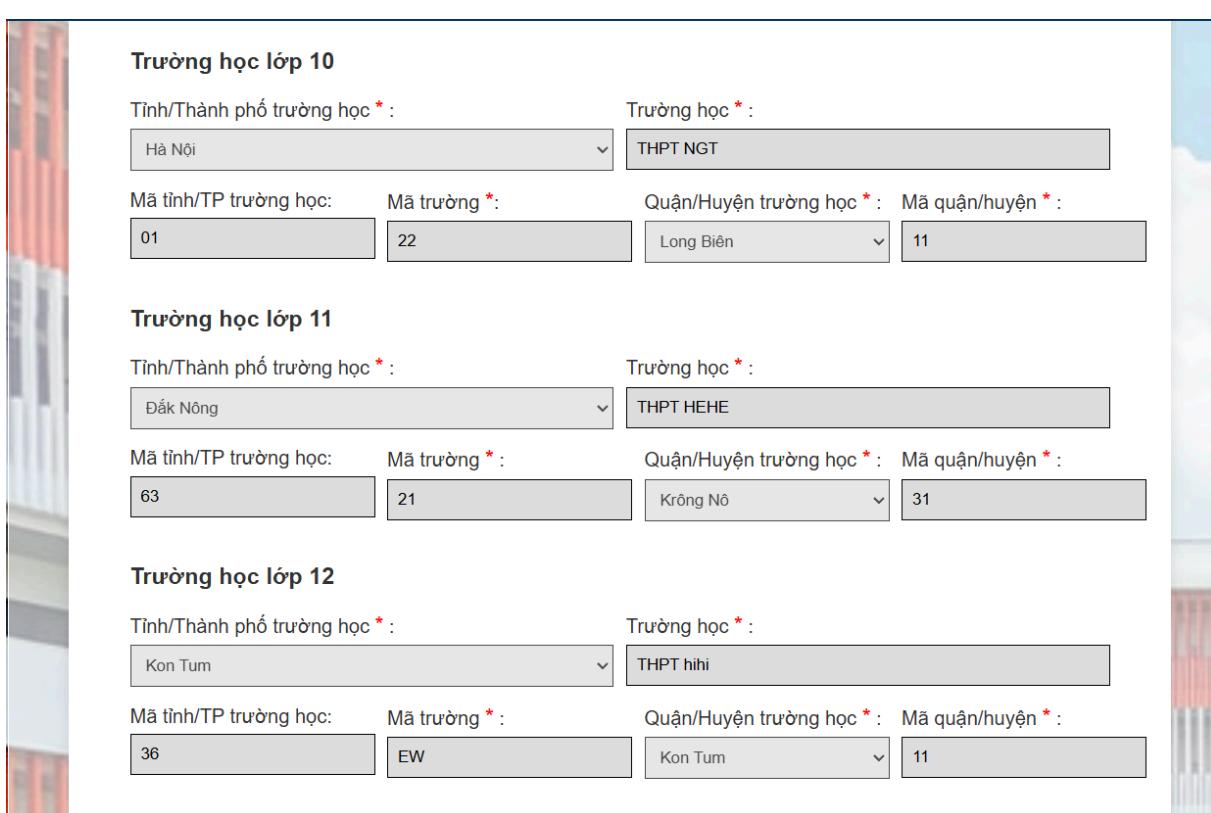
Hình 12. Giao diện đăng ký

### 3. Giao diện nhập thông tin cá nhân (dành cho thí sinh)



The screenshot shows the Phenikaa University application form for personal information entry. The header includes the university logo and navigation links: Hướng dẫn đăng ký, Tin tức, Đăng ký xét tuyển, and Đăng xuất. The main form section is titled "Nhập thông tin cá nhân" (Enter personal information) and contains the following fields:

- Họ và tên \*: Đoàn Hồng Dương
- Ngày tháng năm sinh \*: 05/15/2004
- Giới tính \*: Nam
- Email \*: duong152004@gmail.com
- Dân tộc \*: Kinh
- Tôn giáo \*: Không
- Số CCCD \*: 001204004468
- Ngày cấp \*: 04/21/2021
- Nơi cấp \*: Cục Cảnh sát
- Địa chỉ liên hệ \*: số 63, ngõ 447 đường Ngọc Lâm
- Số điện thoại thí sinh \*: 0979977220
- Số điện thoại bố/mẹ \*: 0331344325
- Mã tỉnh/thành phố: 01
- Tỉnh/Thành phố thường trú \*: Hà Nội
- Quận/Huyện thường trú \*: Long Biên
- Xã/Phường/Thị trấn \*: Ngọc Lâm
- Số nhà, Thôn \*: số 63, ngõ 447 đường Ngọc Lâm

The screenshot shows the application form for selecting the school level. It includes sections for "Trường học lớp 10", "Trường học lớp 11", and "Trường học lớp 12". Each section contains fields for selecting the province/city, school code, district/ward, and district/ward code.

- Trường học lớp 10:**
  - Tỉnh/Thành phố trường học \*: Hà Nội
  - Trường học \*: THPT NGT
  - Mã tỉnh/TP trường học: 01
  - Mã trường \*: 22
  - Quận/Huyện trường học \*: Long Biên
  - Mã quận/huyện \*: 11
- Trường học lớp 11:**
  - Tỉnh/Thành phố trường học \*: Đăk Nông
  - Trường học \*: THPT HEHE
  - Mã tỉnh/TP trường học: 63
  - Mã trường \*: 21
  - Quận/Huyện trường học \*: Krông Nô
  - Mã quận/huyện \*: 31
- Trường học lớp 12:**
  - Tỉnh/Thành phố trường học \*: Kon Tum
  - Trường học \*: THPT hihi
  - Mã tỉnh/TP trường học: 36
  - Mã trường \*: EW
  - Quận/Huyện trường học \*: Kon Tum
  - Mã quận/huyện \*: 11

**Hình 13. Giao diện nhập thông tin cá nhân**

#### 4. Giao diện nhập thông tin học bạ (dành cho thí sinh)

The screenshot shows a web page titled "NHẬP THÔNG TIN HỌC BẠ" (Enter Academic Record Information) under the "Học bạ" (Academic Record) section. At the top, there is a navigation bar with links: "Hướng dẫn đăng ký", "Tin tức", "Đăng ký xét tuyển", and a red button labeled "Đăng xuất" (Logout). The main content area displays a table titled "Bảng điểm học bạ" (Grade Sheet Table) with columns for subjects and grades across three periods: Kỳ 1, Kỳ 2, and Kỳ 12. The subjects listed are Toán, Lý, Hóa, Sinh, Văn, Sử, Địa, Tiếng Anh, Giáo Dục Công Dân, Tiếng Nhật, Tiếng Trung, and Tiếng Hàn. Most subjects show a grade of 9 or 8 across all periods.

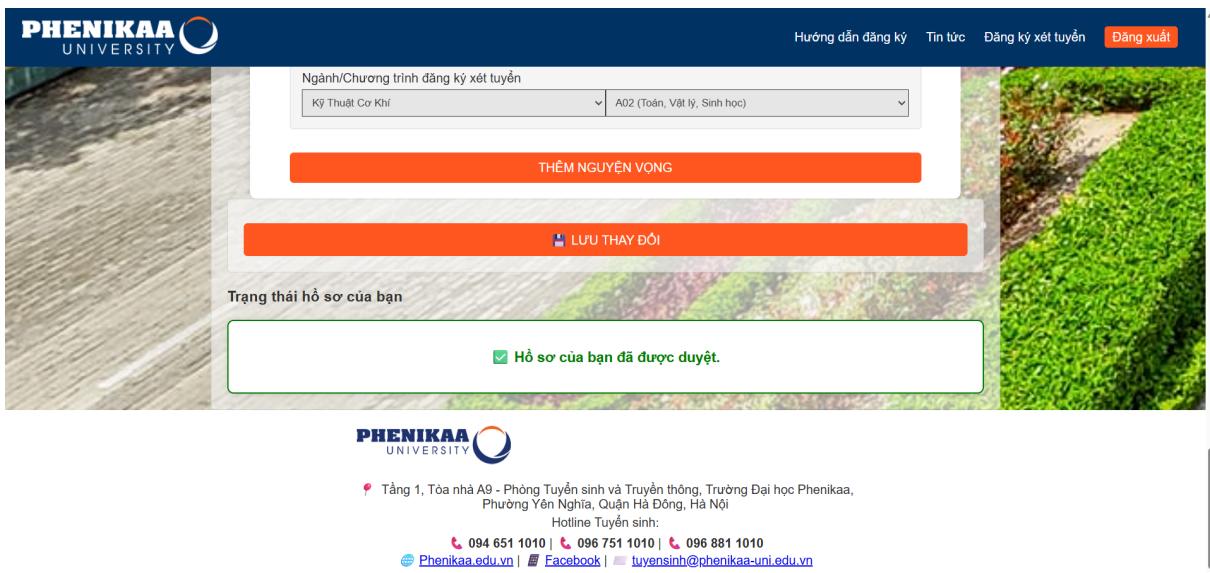
**Hình 14. Giao diện nhập thông tin học bạ (dành cho thí sinh)**

#### 5. Giao diện nhập nguyện vọng (dành cho thí sinh)

The screenshot shows a web page titled "ĐĂNG KÝ NGUYỆN VỌNG" (Enter Preferences) under the "Đăng ký xét tuyển" (Admission Registration) section. The page lists three preference slots. Each slot includes a dropdown menu for selecting a program and a dropdown menu for specifying required subjects. The first slot is for "Nguyễn vong 1" with "Vật Liệu Tiên Tiến Và Công Nghệ Nano" and "A00 (Toán, Vật lý, Hóa học)". The second slot is for "Nguyễn vong 2" with "An Toàn Thông Tin (Một Số Học Phản Chuyên Ngành Học)" and "A01 (Toán, Vật lý, Tiếng Anh)". The third slot is for "Nguyễn vong 3" with "Kỹ Thuật Cơ Khí" and "A02 (Toán, Vật lý, Sinh học)". Below the slots is a large orange button labeled "THÊM NGUYỆN VỌNG" (Add Preference).

**Hình 15. Giao diện nhập nguyện vọng (dành cho thí sinh)**

## 6. Giao diện theo dõi trạng thái hồ sơ



Hình 16. Giao diện theo dõi trạng thái hồ sơ

## 7. Giao diện trang chủ bên admin

The screenshot shows the Phenikaa University administration system. The top bar includes the university logo, the title 'Hệ thống quản lý tuyển sinh', and a date dropdown '2024-2025'. Below this is a form for managing applications, with fields for 'Họ và tên', 'CMND/CCCD', 'Số điện thoại', and 'Ngành học đăng kí'. Buttons for 'Đặt lại' and 'Áp dụng' are at the bottom right. The main area displays a table of application details:

STT	Họ và tên	CMND/CCCD	Số điện thoại	Ngành học đăng kí	Thứ tự nguyện vọng	Xem
1	Đoàn Hồng Dương	001204004468	0979977220	Vật Liệu Tiên Tiến Và Công Nghệ Nano An Toàn Thông Tin (Một Số Học Phần Chuyên Ngành Học Bằng Tiếng Anh) Kỹ Thuật Cơ Khí	1 2 3	Xem
2	Khánh Linh	036304007927	0389903969	Kỹ Thuật Cơ Khí Kỹ Thuật Cơ Điện Tử	1 2	Xem

Hình 17. Giao diện trang chủ bên admin

## 8. Giao diện tìm kiếm thí sinh

Quản lý hồ sơ

Họ và tên: \_\_\_\_\_

Số điện thoại: \_\_\_\_\_

CMND/CCCD: \_\_\_\_\_

Ngành học đăng ký:

Chọn ngành

Đặt lại

Áp dụng

**Hình 18. Giao diện tìm kiếm thí sinh**

## 9. Giao diện danh sách thí sinh

STT	Họ và tên	CMND/CCCD	Số điện thoại	Ngành học đăng ký	Thứ tự nguyện vọng	Xem
1	Đoàn Hồng Dương	001204004468	0979977220	Vật Liệu Tiên Tiến Và Công Nghệ Nano An Toàn Thông Tin (Một Số Học Phần Chuyên Ngành Học Bằng Tiếng Anh) Kỹ Thuật Cơ Khí	1 2 3	Xem
2	Khánh Linh	036304007927	0389903969	Kỹ Thuật Cơ Khí Kỹ Thuật Cơ Điện Tử Vật Liệu Tiên Tiến Và Công Nghệ Nano	1 2 3	Xem
3	Ninh Thị Phương Mai	036304007927	0389903969	Kỹ Thuật Y Sinh (Hệ Tự Sinh) Kỹ Thuật Điện Tử - Viễn Thông (Hệ Thống Thông Minh Và IoT) Kỹ Thuật Cơ Khí	1 2 3	Xem
4	Duong	001204004458	0979877220	Kỹ Thuật Cơ Điện Tử Tài Năng Khoa Học Máy Tính Chip Bán Dẫn Và Công Nghệ Đóng Gói	1 2 3	Xem
				Khoa Học Máy Tính	1	

**Hình 19. Giao diện danh sách thí sinh**

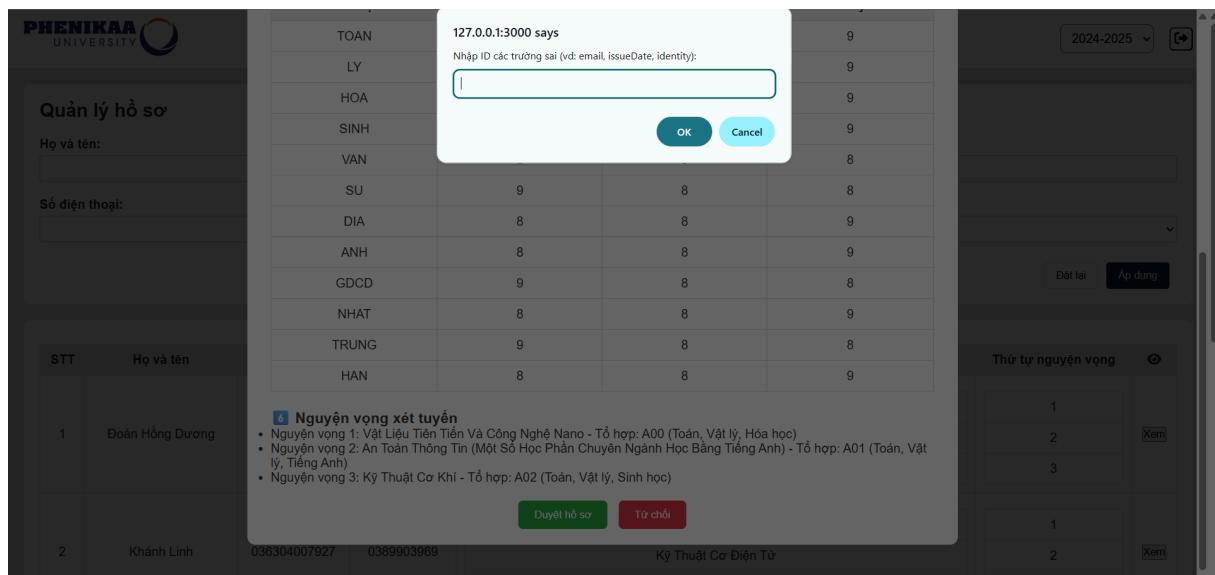
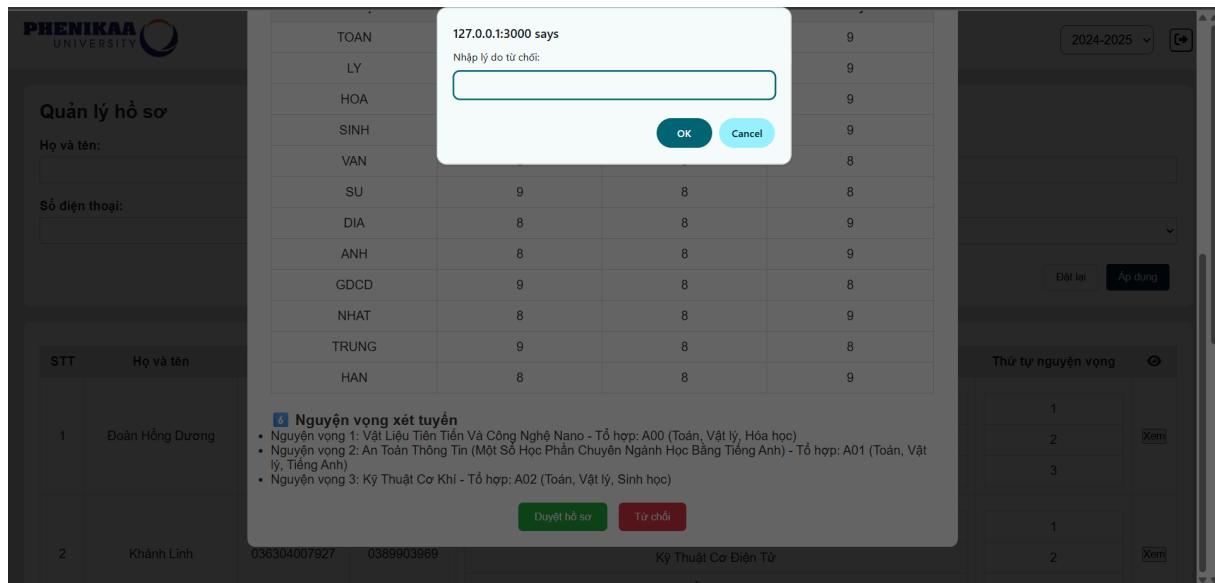
## 10. Giao diện xem chi tiết hồ sơ thí sinh

STT	Họ và tên	Thứ tự nguyện vọng		
1	Đoàn Hồng Dương	<b>1</b>	<b>2</b>	<b>3</b>
2	Khánh Linh	<b>1</b>	<b>2</b>	<b>3</b>
3	Ninh Thị Phương Mai	<b>1</b>	<b>2</b>	<b>3</b>
4	Duong	<b>1</b>	<b>2</b>	<b>3</b>

<b>1</b> Thông tin cá nhân Họ và tên: Đoàn Hồng Dương Email: duong152004@gmail.com	Giới tính: male Dân tộc: kinh	Ngày sinh: 2004-05-15 Tôn giáo: no	
<b>2</b> Thông tin CCCD Số CCCD: 001204004468	Ngày cấp: 2021-04-21	Nơi cấp: Cục Cảnh sát	
<b>3</b> Địa chỉ liên hệ Tỉnh/TP: Hà Nội Số nhà/H thôn: số 63, ngõ 447 đường Ngọc Lâm	Quận/Huyện: Long Biên	Xã/Phường: Ngọc Lâm SĐT thí sinh: 0979977220 SĐT phụ huynh: 0331344325	
<b>4</b> Thông tin trường học Lớp 10: THPT NGT (Long Biên, Hà Nội)	Lớp 11: THPT HEHE (Krông Nô, Đăk Nông)	Lớp 12: THPT hihi (Kon Tum, Kon Tum)	
<b>5</b> Kết quả học bạ			
MON HỌC	11 - KỲ 1	11 - KỲ 2	12 - KỲ 1
TOAN	9	9	9
LY	9	9	9
HOA	9	9	9
SINH	8	8	9
VAN	8	9	8
SU	9	8	8
DIA	8	8	9

**Hình 20. Giao diện xem chi tiết hồ sơ thí sinh**

## 11. Giao diện yêu cầu chỉnh sửa thông tin thí sinh



**Hình 21. Giao diện yêu cầu chỉnh sửa thông tin thí sinh**

### CHƯƠNG III. KẾ HOẠCH DỰ ÁN

Bảng phân công chi tiết các công việc của từng thành viên:

Kế hoạch chi tiết cho dự án:

Vai trò	Ninh Thị Phương Mai	Đoàn Hồng Dương	Đặng Hồng Nhung	
		Thành viên	Trưởng nhóm	Thành viên
STT	Nội dung công việc	Phần % công việc thực hiện		
1	Lập kế hoạch	33,33%	33,33%	33,33%
2	Nghiên cứu các tài nguyên sử dụng	80%	20%	0%
3	Hoàn thiện việc thêm, lưu dữ liệu cơ bản	100%	0%	0%
4	Hoàn thiện chức năng duyệt và cập nhật trạng thái	0%	100%	0%
5	Tích hợp thêm modal chi tiết thí sinh	0%	100%	50%
6	Hoàn thiện chức năng theo GVHD đề xuất	0%	0%	100%
7	Viết báo cáo	40%	40%	20%
8	Kiểm thử sản phẩm	33,33%	33,33%	33,33%
9	Kiểm tra báo cáo	33,33%	33,33%	33,33%
10	Làm slides	0%	0%	100%
11	Toàn bộ dự án	35%	35%	30%

## CHƯƠNG IV. KẾT LUẬN

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

Vấn đề đạo đức trong phát triển và quản lý hệ thống tuyển sinh trực tuyến không chỉ là khía cạnh kỹ thuật, mà còn là một trách nhiệm xã hội và nghề nghiệp vô cùng quan trọng. Trong bối cảnh số hóa hiện nay, dữ liệu cá nhân của thí sinh như họ tên, số CMND, điểm thi, trường đăng ký,... là những thông tin nhạy cảm cần được bảo vệ tuyệt đối. Việc bảo đảm an toàn dữ liệu không chỉ là yêu cầu pháp lý, mà còn là chuẩn mực đạo đức bắt buộc của mỗi người phát triển hệ thống.

Thông tin cá nhân phải được xem là tài sản quý giá của người dùng, không được phép lạm dụng hoặc chia sẻ trái phép. Việc rò rỉ hoặc sử dụng sai mục đích dữ liệu có thể gây ảnh hưởng nghiêm trọng đến quyền lợi của thí sinh và làm tổn hại uy tín của hệ thống. Vì vậy, nhóm phát triển cần tuân thủ nghiêm ngặt các nguyên tắc bảo mật, áp dụng công nghệ xác thực an toàn như Firebase Authentication, mã hóa dữ liệu và bảo đảm rằng mọi thao tác truy cập dữ liệu đều được kiểm soát chặt chẽ.

Ngoài ra, đạo đức nghề nghiệp còn thể hiện ở cách tương tác với người dùng. Hệ thống không nên gửi thông báo không cần thiết, không làm phiền hoặc sử dụng thông tin của người dùng cho mục đích thương mại. Mọi tính năng thông báo hay liên hệ cần được thiết kế dựa trên sự tôn trọng người dùng và mục tiêu hỗ trợ học tập, đảm bảo rằng người dùng luôn cảm thấy tin tưởng và an tâm khi sử dụng ứng dụng.

Làm việc chuyên nghiệp trong quá trình phát triển hệ thống là nền tảng để duy trì chất lượng và uy tín lâu dài. Điều này bao gồm việc làm việc nhóm hiệu quả, quản lý tiến độ khoa học, kiểm thử kỹ lưỡng, cập nhật công nghệ thường xuyên và cung cấp dịch vụ hỗ trợ kịp thời cho người dùng. Một nhóm phát triển chuyên nghiệp không chỉ tạo ra sản phẩm tốt, mà còn duy trì tính minh bạch, trách nhiệm và tôn trọng đạo đức nghề nghiệp trong mọi khâu thực hiện.

Trong thời đại công nghệ 4.0, đạo đức và tính chuyên nghiệp không chỉ là yếu tố bổ sung, mà chính là nền tảng vững chắc cho sự thành công bền vững của hệ thống tuyển sinh. Bằng cách đặt người dùng làm trung tâm và tuân thủ các giá trị đạo đức, chúng ta có thể xây dựng một nền tảng công nghệ đáng tin cậy, an toàn và thực sự mang lại giá trị cho cộng đồng giáo dục.

## B. Tác động xã hội

Ứng dụng tuyển sinh trực tuyến không chỉ là công cụ hỗ trợ cá nhân trong việc đăng ký xét tuyển, mà còn mang lại nhiều lợi ích tích cực cho cộng đồng và hệ thống giáo dục.

Thứ nhất, ứng dụng giúp tối ưu hóa quy trình tuyển sinh, giảm thiểu các thủ tục hành chính phức tạp. Thí sinh có thể đăng ký, theo dõi hồ sơ, và nhận kết quả trực tuyến thay vì phải đến trực tiếp trường, từ đó tiết kiệm thời gian, chi phí và công sức cho cả thí sinh lẫn cán bộ tuyển sinh.

Thứ hai, hệ thống góp phần giảm sự phụ thuộc vào giấy tờ và hồ sơ truyền thống, hướng đến môi trường tuyển sinh “xanh”. Việc chuyển đổi sang hình thức đăng ký và quản lý hồ sơ điện tử giúp tiết kiệm tài nguyên, giảm rác thải và bảo vệ môi trường.

Thứ ba, ứng dụng mang lại tính minh bạch và công bằng trong quá trình tuyển sinh. Thí sinh có thể tự kiểm tra tình trạng hồ sơ, điểm xét tuyển, và kết quả duyệt theo thời gian thực, giúp giảm sai sót, tiêu cực và tăng độ tin cậy trong công tác tuyển sinh.

Thứ tư, hệ thống tạo ra nhiều cơ hội nghề nghiệp mới trong lĩnh vực công nghệ thông tin, thiết kế giao diện, và quản lý dữ liệu giáo dục. Quá trình xây dựng, vận hành và bảo trì hệ thống đòi hỏi đội ngũ kỹ sư phần mềm, chuyên viên dữ liệu và cán bộ kỹ thuật có chuyên môn cao.

Thứ năm, ứng dụng giúp nâng cao chất lượng quản lý giáo dục, khi dữ liệu tuyển sinh được lưu trữ và phân tích tập trung. Nhà trường có thể dễ dàng thống kê, đánh giá và điều chỉnh chỉ tiêu tuyển sinh cho phù hợp với thực tế, góp phần nâng cao hiệu quả quản lý và hoạch định chiến lược đào tạo.

Cuối cùng, ứng dụng thúc đẩy chuyển đổi số trong lĩnh vực giáo dục, giúp các cơ sở đào tạo thích ứng tốt hơn với xu thế công nghệ hiện nay. Việc số hóa quá trình tuyển sinh không chỉ hiện đại hóa công tác quản lý, mà còn mở rộng khả năng tiếp cận giáo dục cho mọi đối tượng trên phạm vi rộng.

Tóm lại, hệ thống tuyển sinh trực tuyến mang lại ảnh hưởng xã hội sâu rộng: tăng hiệu quả, giảm chi phí, nâng cao tính minh bạch, đồng thời góp phần vào công cuộc chuyển đổi số giáo dục quốc gia và thúc đẩy phát triển bền vững trong kỷ nguyên số.

## **C. Kế hoạch cho kiến thức mới và chiến lược học tập**

### **1. Tìm Hiểu Web và công nghệ Firebase:**

#### **Mục tiêu:**

Hiểu rõ các công nghệ nền tảng được sử dụng trong hệ thống tuyển sinh, bao gồm **HTML, CSS, JavaScript** và **Firebase**.

Làm chủ các kỹ thuật lập trình front-end để phát triển giao diện và back-end để xử lý dữ liệu và xác thực người dùng.

#### **Kế hoạch:**

- HTML & CSS:** Nâng cao kỹ năng xây dựng cấu trúc trang web và thiết kế giao diện người dùng trực quan, hiện đại, tối ưu cho cả máy tính và thiết bị di động.
- JavaScript:** Học sâu hơn về DOM, sự kiện, và lập trình bất đồng bộ (asynchronous) để xử lý dữ liệu linh hoạt.

- **Firebase:** Tìm hiểu cách sử dụng Firebase Authentication, Firestore Database và Hosting để triển khai hệ thống đăng ký, đăng nhập và lưu trữ dữ liệu người dùng an toàn.
- **Thực hành:** Tạo các dự án nhỏ mô phỏng chức năng con (ví dụ: form đăng ký, đăng nhập, xem hồ sơ), từ đó tích hợp dần vào ứng dụng chính.
- **Tài nguyên học tập:** Tham khảo tài liệu chính thức từ Firebase Docs, MDN Web Docs, và các khóa học trực tuyến (Coursera, W3Schools, FreeCodeCamp).

## **2. Xây Dựng Giao Diện Người Dùng (UI):**

### **Mục tiêu:**

- Thiết kế giao diện người dùng thân thiện, dễ thao tác và có tính thẩm mỹ cao.  
Đảm bảo bố cục hợp lý giữa các thành phần: thanh menu, form đăng ký/xét tuyển, khu vực hiển thị thông báo, và khu vực quản lý hồ sơ.
- Đảm bảo giao diện **tương thích đa nền tảng** (responsive) trên các loại thiết bị và trình duyệt khác nhau.

### **Kế hoạch:**

- **Học và áp dụng các nguyên tắc thiết kế UI/UX cơ bản** như tính dễ đọc, nhất quán màu sắc, và tối giản nội dung.
- **Sử dụng CSS và thư viện hỗ trợ** (Bootstrap, Tailwind CSS) để tăng tốc độ thiết kế và đảm bảo tính responsive.
- **Xây dựng thanh menu chính** chứa các mục: *Hướng dẫn đăng ký, Tin tức, Đăng ký xét tuyển, Theo dõi hồ sơ, Đăng xuất*.
- **Thiết kế form và bảng hiển thị dữ liệu rõ ràng**, dễ thao tác, có xác thực dữ liệu đầu vào.
- **Thử nghiệm giao diện thực tế** với người dùng (thí sinh hoặc admin giả lập) để thu thập phản hồi và cải tiến trải nghiệm.

### **3. Quản lý dữ liệu và xử lý bất đồng bộ.**

#### **Mục tiêu:**

Hiểu cách **quản lý dữ liệu (state)** trong ứng dụng web và xử lý các **tác vụ bất đồng bộ (asynchronous)** như đọc, ghi dữ liệu từ Firebase hoặc gọi API.

Đảm bảo dữ liệu luôn được đồng bộ giữa giao diện và cơ sở dữ liệu, giúp ứng dụng hoạt động ổn định và mượt mà.

#### **Kế hoạch:**

- Quản lý dữ liệu (State):**

Học cách lưu trữ và cập nhật dữ liệu trong các biến và cấu trúc dữ liệu của JavaScript.

Thực hành xử lý các tình huống thay đổi dữ liệu trong form đăng ký, danh sách hồ sơ hoặc thông báo trạng thái mà không cần tải lại trang.

- Xử lý bất đồng bộ:**

Tìm hiểu và sử dụng **Promise, async/await, và callback** trong JavaScript để xử lý các tác vụ như đọc/ghi dữ liệu từ **Firebase Firestore** hoặc xác thực người dùng qua **Firebase Authentication**.

- Thực hành:**

- Viết các hàm tương tác với Firebase (ví dụ: thêm hồ sơ, cập nhật thông tin, lấy dữ liệu người dùng).
- Thực hành hiển thị thông tin động trên giao diện (ví dụ: trạng thái hồ sơ cập nhật tự động sau khi admin duyệt).
- Kiểm tra và xử lý lỗi khi thao tác với dữ liệu (ví dụ: mất kết nối mạng, nhập sai định dạng).

- Mục tiêu cuối cùng:**

Đảm bảo ứng dụng hoạt động ổn định, phản hồi nhanh và dữ liệu giữa người dùng và Firebase luôn được đồng bộ chính xác.

### **4. Tối Ưu Hóa Hiệu Suất:**

## **Mục tiêu:**

Hiểu và áp dụng các kỹ thuật tối ưu hóa hiệu suất cho ứng dụng web sử dụng HTML, CSS, JavaScript và Firebase, nhằm giúp hệ thống tải nhanh, hoạt động ổn định và mang lại trải nghiệm tốt nhất cho người dùng.

## **Kế hoạch:**

- **Tối ưu mã nguồn:**

Rà soát và loại bỏ các đoạn mã thừa, gộp hoặc nén (minify) các tệp CSS và JavaScript để giảm dung lượng tải.

Sử dụng kỹ thuật caching của trình duyệt và Firebase Hosting để tăng tốc độ truy cập.

- **Tối ưu hình ảnh và tài nguyên:**

Giảm kích thước ảnh, dùng định dạng hiện đại (như WebP), chỉ tải hình ảnh khi cần (lazy loading hình ảnh).

- **Tối ưu thao tác với Firebase:**

Hạn chế số lượng truy vấn và chỉ tải dữ liệu cần thiết từ Firestore.

Sử dụng Realtime Update hiệu quả để đồng bộ thông tin mà không làm chậm hệ thống.

- **Cấu trúc và giao diện:**

Chia nhỏ các phần giao diện (modular structure) để dễ quản lý và giảm độ phức tạp khi tải trang.

Sử dụng CSS hiệu quả (tránh lặp lại, tái sử dụng class, dùng external stylesheet).

- **Kiểm thử hiệu suất:**

Kiểm tra tốc độ tải trang bằng các công cụ như Google Lighthouse hoặc GTmetrix.

Đo lường thời gian phản hồi của Firebase và tối ưu lại các truy vấn chậm.

## **5. Kiểm Thử và Debugging:**

## **Mục Tiêu:**

- Biết cách sử dụng các công cụ debugging.
- Thực hành việc viết các loại kiểm thử như unit tests và integration tests.

## **Kế Hoạch:**

- Sử dụng DevTools của trình duyệt để debug.
- Tìm hiểu và triển khai kiểm thử bằng Jest hoặc các công cụ tương tự.

## **6. Triển Khai và Quản Lý Dự Án:**

### **Mục tiêu:**

Hiểu và áp dụng các phương pháp kiểm thử và gỡ lỗi (debugging) trong quá trình phát triển ứng dụng web.

Biết cách phát hiện, phân tích và khắc phục lỗi trong mã JavaScript, xử lý lỗi giao tiếp với Firebase, cũng như đảm bảo tính ổn định của hệ thống.

### **Kế hoạch:**

- **Sử dụng công cụ DevTools của trình duyệt:**

Dùng Chrome DevTools để kiểm tra lỗi JavaScript, theo dõi network request đến Firebase, phân tích hiệu suất tải trang và kiểm tra giao diện trên các kích thước màn hình khác nhau.

- **Kiểm thử thủ công (Manual Testing):**

Thực hiện kiểm thử trực tiếp các chức năng của trang web như đăng ký, đăng nhập, truy cập dữ liệu từ Firebase, và các thao tác CRUD để đảm bảo hoạt động đúng yêu cầu.

- **Kiểm thử tự động (Automated Testing):**

Tìm hiểu cách sử dụng các công cụ như Jest hoặc Mocha để viết unit test cho các hàm JavaScript quan trọng.

Thực hiện integration test cho các phần giao tiếp với Firebase nhằm đảm bảo tính toàn vẹn của dữ liệu.

- **Quản lý và ghi log lỗi:**

Sử dụng console.log, try...catch và cơ chế ghi log của Firebase để phát hiện, theo dõi và khắc phục lỗi trong quá trình chạy.

- **Đánh giá và cải tiến:**

Sau khi kiểm thử, tổng hợp kết quả, xác định nguyên nhân lỗi và cập nhật lại mã nguồn để đảm bảo ứng dụng hoạt động ổn định, an toàn và hiệu quả hơn.

## 7. Học Tiếp và Cập Nhật:

### Mục tiêu:

Liên tục cập nhật và mở rộng kiến thức về các công nghệ web hiện đại như HTML5, CSS3, JavaScript, và Firebase.

Theo dõi các xu hướng phát triển mới trong lĩnh vực ứng dụng web và cơ sở dữ liệu đám mây, đồng thời học hỏi từ cộng đồng lập trình viên để nâng cao kỹ năng và cải thiện chất lượng sản phẩm.

### Kế hoạch:

- **Theo dõi tài liệu và cập nhật kỹ thuật:** Thường xuyên đọc tài liệu chính thức của Firebase, ECMAScript (JS) và các nguồn học uy tín như MDN Web Docs để nắm bắt các tính năng mới, tiêu chuẩn web, và phương pháp tối ưu hóa hiệu suất.
- **Tham gia cộng đồng lập trình:** Tham gia các diễn đàn, nhóm Facebook, hoặc cộng đồng như Stack Overflow, Google Firebase Community, nơi các lập trình viên chia sẻ kinh nghiệm và giải pháp thực tế.
- **Thực hành và thử nghiệm:** Áp dụng các kỹ thuật mới vào dự án hiện tại hoặc xây dựng các ứng dụng nhỏ thử nghiệm để rèn luyện kỹ năng.
- **Cập nhật định kỳ:** Đặt mục tiêu kiểm tra và cải tiến mã nguồn, tối ưu bảo mật và giao diện mỗi quý để đảm bảo hệ thống luôn tương thích và hoạt động hiệu quả theo xu hướng công nghệ mới.

## D. Kết quả đạt được và hạn chế

### 1. Kết quả đạt được

Đồ án đã xây dựng thành công "Hệ thống tuyển sinh trực tuyến" ở mức độ cơ bản , đáp ứng được các mục tiêu cốt lõi đã đề ra:

- **Xây dựng thành công các chức năng cốt lõi:** Hệ thống đã hoàn thiện hai luồng nghiệp vụ chính:
  - **Luồng dành cho Thí sinh:** Cho phép người dùng đăng ký tài khoản (sử dụng Firebase Authentication ), đăng nhập, điền và nộp hồ sơ trực tuyến (bao gồm thông tin cá nhân, học bạ, nguyện vọng), và theo dõi trạng thái hồ sơ (Mới nộp, Đã duyệt, Yêu cầu bổ sung).
  - **Luồng dành cho Quản trị viên (Admin):** Cung cấp giao diện quản lý cho phép Admin xem danh sách hồ sơ, tìm kiếm, xem chi tiết và thực hiện nghiệp vụ duyệt hoặc yêu cầu chỉnh sửa hồ sơ.
- **Ứng dụng công nghệ Firebase hiệu quả:** Đã áp dụng thành công các dịch vụ của Firebase, bao gồm Firebase Authentication để xác thực người dùng và Firebase Realtime Database (hoặc Firestore) để lưu trữ và đồng bộ dữ liệu hồ sơ theo thời gian thực .
- **Tạo ra sản phẩm có tính thực tiễn:** Hệ thống giúp số hóa quy trình nộp hồ sơ thủ công, giảm thiểu sai sót giấy tờ, tiết kiệm thời gian cho cả thí sinh và nhà trường. Giao diện người dùng được xây dựng dựa trên HTML, CSS và JavaScript, đảm bảo tính tương tác cơ bản.
- **Nâng cao kiến thức và kỹ năng:** Nhóm đã học hỏi và vận dụng được quy trình phát triển một ứng dụng web hoàn chỉnh, từ khâu phân tích thiết kế (UML), lựa chọn công nghệ, đến triển khai và kiểm thử.

### 2. Khó khăn vướng mắc

Trong quá trình thực hiện đồ án , nhóm đã gặp phải một số khó khăn và vướng mắc:

- **Khó khăn về công nghệ:**

- **Firebase Security Rules:** Việc thiết kế và triển khai các quy tắc bảo mật (Security Rules) cho Firebase Database để vừa đảm bảo an toàn dữ liệu, vừa cho phép người dùng (Thí sinh và Admin) truy cập đúng quyền hạn là một thách thức. Ví dụ: làm sao để Thí sinh chỉ thấy hồ sơ của mình, nhưng Admin thì thấy được tất cả.
- **Xử lý bất đồng bộ:** Do JavaScript và Firebase hoạt động dựa trên cơ chế bất đồng bộ (asynchronous), việc xử lý các luồng dữ liệu (ví dụ: lấy thông tin tài khoản sau khi đăng nhập rồi mới lấy tiếp thông tin hồ sơ) đòi hỏi sự hiểu biết sâu về Promises, async/await để tránh lỗi logic.

- **Khó khăn về nghiệp vụ:**

- **Phân tích nghiệp vụ tuyển sinh:** Quy trình tuyển sinh thực tế có nhiều trường hợp phức tạp (điểm ưu tiên, xét tuyển kết hợp, nhiều đợt tuyển sinh) mà trong khuôn khổ đề án cơ sở chưa thể phân tích và đáp ứng hết.
- **Quản lý trạng thái hồ sơ:** Việc thiết kế luồng chuyển đổi trạng thái hồ sơ (ví dụ: khi nào Thí sinh được phép sửa, khi nào bị khóa) cần phải chặt chẽ để tránh xung đột dữ liệu.

- **Khó khăn về quản lý dự án:**

- **Quản lý thời gian:** Khối lượng công việc cho cả front-end (giao diện) và back-end (logic Firebase) là khá lớn so với thời gian thực hiện, dẫn đến một số chức năng phải rút gọn.
- **Phối hợp nhóm:** Vì dự án có nhiều thành viên (theo bìa báo cáo), việc đồng bộ mã nguồn (code) và đảm bảo các thành phần tích hợp trọn vẹn với nhau cũng là một thách thức.

### 3. Hướng phát triển trong tương lai

Để hệ thống trở nên hoàn thiện và có khả năng ứng dụng rộng rãi hơn, nhóm đề xuất các hướng phát triển sau:

- **Hoàn thiện chức năng cho Thí sinh:**

- Phát triển chức năng "Quên mật khẩu".
- Cho phép Thí sinh tải lên các tệp đính kèm (minh chứng học bạ, ảnh CCCD, giấy chứng nhận ưu tiên) thay vì chỉ nhập liệu văn bản.
- Tích hợp hệ thống thông báo (Email hoặc Notification) tự động khi trạng thái hồ sơ bị thay đổi (ví dụ: khi được duyệt hoặc bị yêu cầu bổ sung).

- **Nâng cao nghiệp vụ cho Admin:**

- Xây dựng trang Dashboard (Bảng điều khiển) với các biểu đồ thống kê (số lượng hồ sơ theo ngày, theo ngành, theo trạng thái) để Admin có cái nhìn tổng quan.
- Thêm chức năng xuất dữ liệu (Export) danh sách thí sinh ra file Excel/CSV để phục vụ công tác hậu kiểm và nhập liệu vào các hệ thống khác của nhà trường.

- **Cải tiến về kỹ thuật và bảo mật:**

- Tối ưu hóa Security Rules của Firebase một cách chi tiết và chặt chẽ hơn ở cấp độ từng tài liệu (document-level).
- Cải thiện giao diện người dùng (UI/UX), đảm bảo thiết kế đáp ứng (Responsive Design) để hệ thống hoạt động tốt trên cả thiết bị di động.
- Tích hợp các cổng thanh toán (Payment Gateway) để xử lý lẻ phí xét tuyển trực tuyến (nếu có).

## CHƯƠNG V. TÀI LIỆU THAM KHẢO

Trường Đại học Phenikaa, *Thông tin tuyển sinh đại học chính quy*, truy cập tại địa chỉ: <https://tuyensinh.phenikaa-uni.edu.vn/>.

Google Developers, *Firebase Documentation (Firestore, Authentication, Security Rules)*. Available at: <https://firebase.google.com/docs>.

Mozilla Developer Network, *MDN Web Docs (HTML, CSS, JavaScript references)*. Available at: <https://developer.mozilla.org/>.